ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

UDC 004.6.032.24:004.85 **doi: 10.32620/reks.2026.1.10**

Chaimaa BENYAMANI[2], Hassan BADI[1], Imad BADI[2], Aziz KHAMJANE[2]

[1] ***Sidi Mohamed Ben Abdellah University, Fes, Morocco***
[2] ***Abdelmalek Esaadi University, Tétouan, Morocco***

SENTIMENT ANALYSIS OF TRENDING TWEETS USING SPARK NLP AND DEEP LEARNING: A BENCHMARK STUDY OF CNN VS TRANSFORMER MODELS

*This study **investigates** the implementation of different sentiment analysis models, exploring their theoretical foundations, robust evaluation criteria, and significant findings, and integrating natural language processing methods to preprocess data collected from social media, the main source of information. The **goal** of this study is to present notable advancements in sentiment classification by implementing innovative models, including Convolutional Neural Networks and Transformer-based architectures such as BERT, DistilBERT, and RoBERTa, with Spark NLP for preprocessing. The **tasks** to be performed include: collecting the dataset from Twitter/X focusing particularly on trending topics that appear daily; performing preprocessing steps using Spark NLP, a prominent and scalable NLP library built on Apache Spark to handle scalable and distributed processing of textual data; assigning the polarity for each tweet by applying VADER lexicon-based tool; pretraining both TextCNN and Transformer-based models on lexicon-based VADER labels under identical parameters; fine-tuning the models on manually annotated tweets; comparing their effectiveness, evaluating their strengths, weaknesses, and overall performance in sentiment classification that can guide model selection in resource-constrained settings. The **methods** used include Spark NLP pipelines, lexicon-based weak labeling, two-phase supervised learning on Convolutional Neural Networks (TextCNN), and Transformer-based models (BERT, DistilBERT, RoBERTa) with early stopping and learning-rate warm-up strategies, and finally the comparative evaluation metrics. The dataset used in this work (12,422 tweets, including 3,475 manually labeled tweets) is not large. The use of Apache Spark enables distributed data processing and supports scalability for larger datasets. The **results** show that the Transformer-based models outperformed TextCNN model in terms of classification accuracy and robustness. RoBERTa achieved the highest accuracy (**85%**), followed by BERT and DistilBERT (**84%**) and TextCNN (**72%**). DistilBERT balances the predictive performance and computational efficiency well. **Conclusions**. The **scientific novelty** of this study focuses particularly on the integration of Apache Spark NLP preprocessing, followed by a benchmarking of CNN and Transformer-based models trained on trending tweets in a hybrid two-phase learning strategy that combined lexicon-based weak supervision with human annotation under the same experimental conditions to provide methodological insights and select appropriate model in our sentiment classification experiments.*

Keywords: *Sentiment Analysis; Apache Spark NLP; social media; TextCNN; Bidirectional Encoder Representations from Transformers (BERT); DistilBERT; RoBERTa; Two-phase learning; Weak supervision.*

1. Introduction

1.1. Motivation

The global popularity of the Internet has grown because it enables more flexible, convenient communication. Social media and its platforms allow users to have conversations, share information, broaden their knowledge on a particular field, and express their sentiments toward topics or trends.

Applications like X (formerly Twitter) have proved itself as a prominent microblogging platform [1] in evaluating individual behavior, allowing researchers to analyze sentiment patterns and detect their polarity, which can be positive, negative, or neutral. Sentiment analysis, popularly identified as opinion mining, has become a prominent area as a Natural Language Processing understanding benchmark to classify text into positive, negative, and neutral sentiment categories [2]. It automates the analysis of public insights and attitudes from a high volume of data gathered from the X platform to guide decision-making across various domains. The collected tweet polarity was classified in four key steps: data selection, preprocessing, feature extraction, and model training.

To avoid designing task-specific architectures and training from scratch, researchers have focused on pre-training different language models and fine-tuning them for downstream tasks [3]. This paradigm has notably improved the accuracy of sentiment analysis by adapting foundational models to specific applications and contexts. Google built a new language representation

called Bidirectional Encoder Representations from Transformers (BERT) built to pre-train deep bidirectional representations from unlabeled text and fine-tuned on labeled data for diverse natural language processing (NLP) tasks [3].

The progress in machine learning models has been rapidly accelerating, and new and more powerful algorithms continue to emerge, addressing the shortcomings of their predecessors. Within the scope of downstream tasks, CNNs and Transformers stand out as two leading architectures, which are commonly the subject of extensive comparative studies that highlight their strengths and weaknesses [4]. Transformer-based models, such as BERT, RoBERTa, and DistilBERT, set new benchmarks for NLP tasks, but they require substantial computational resources. While CNN models like TextCNN offer faster training, their application in scalable text processing remains underexplored. However, despite their success, the following question remains: **Which approach is more effective for specific tasks when both approaches are applied to the same large-volume tweet corpus? How does their performance compare to real-world applications?**

In this paper, we provide a comparative study of these models to offer insights into the optimal choice and architecture that best balances accuracy and efficiency in large-volume settings.

1.2. State of the art

There have been studies on text sentiment classification using BERT (Pre-training of Deep Bidirectional Transformers for Language Understanding), which is relevant to our research.

The authors of [5] developed a CNN-based model for Twitter sentiment classification using the enhanced Gorilla Troops Optimization algorithm. Using the SemEval-2016 Twitter datasets, the model achieved around 98% performance and demonstrated robustness in handling noisy and short-text data of social media. This study provides a valuable baseline for benchmarking Transformer-based models when integrated with Spark NLP framework.

In their paper [6], the authors proposed a sentiment analysis and polarity prediction system using tweets about COVID-19 vaccines. They leveraged Apache Spark for large-scale text preprocessing and applied deep learning techniques, particularly LSTM, to forecast sentiment trends. Although their work focuses on vaccine-related discourse, it demonstrates a scalable approach to collecting, preprocessing, and classifying tweets using NLP, which closely aligns with the methodology adopted in our study. We extend this line of research by comparing multiple models (TextCNN, DistilBERT, and RoBERTa) on a broader spectrum of trending topics for real-time sentiment classification.

Singla and Ramachandra [7] performed a comparative analysis of several transformer-based models (BERT, RoBERTa, and ALBERT) for multiclass sentiment analysis using a COVID-19 tweet dataset. They evaluated models using metrics such as F1-score and AUC, highlighting BERT as the most accurate, though computationally heavier, and RoBERTa as a faster alternative with acceptable performance. This work is closely related to our study as we also fine-tune transformer models, including DistilBERT and RoBERTa, for sentiment classification.

The study in [8] conducted a comprehensive comparison of different NLP models, including BERT base, Bio+Clinical BERT, and CNN, to predict drug review satisfaction using the UCI ML Drug Review dataset. The domain-specific Bio+Clinical BERT model outperformed the general-purpose BERT base model. While BERT-based models were good at understanding long-range context and specific medical terms, the simpler CNN was better at identifying key sentiment phrases, especially in reviews with mixed language or non-medical terms. This shows the importance of using both domain-specific, pertinent models and simpler models, depending on the type of text.

A hybrid sentiment analysis model that employs both BERT and CNN architecture to evaluate public opinion on Covid-19 tweets was proposed in [9]. The study revealed that BERT alone achieved 88.4% classification accuracy due to its contextual understanding. The CNN excelled at extracting local text patterns. By combining both models into one approach, they achieved an even higher accuracy of 90%. This study highlights the benefits of combining deep contextual embedding with convolutional pattern recognition in social media sentiment analysis.

The authors of this paper [10] built a live sentiment analysis pipeline on Twitter data using Apache Spark with Scala and the Naïve Bayes classifier on preprocessed tweet text. Although this work achieves good performance, its limitation is that it relies on simple and classical machine learning methods rather than transformer-based models. In our research, we advanced this restriction by combining Apache Spark NLP with advanced deep learning architectures, such as architectures like CNN and Transformer models.

The authors in [11] evaluate different machine learning models such as: Logistic Regression, SVM, Random Forest, and Naïve Bayes to extract which one performs well under the Spark distributed environment. The findings show that Spark increases the speed of data processing and ensures scalability. This study does not cover models that can capture the deeper semantic

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

meaning in tweets; instead, it relies only on classical machine learning models.

In their paper [12], the authors focused on deep learning and Transformer-based models such as BERT, RoBERTa, and DistilBERT. With their self-attention mechanisms, Transformers outperform earlier deep learning models such as CNNs and LSTMs when applied on various sentiment datasets. This study reinforces the idea of comparing CNN and Transformer models and shows that the latter have become the dominant and most preferred choice for modern sentiment analysis tasks, especially when used with Big Data frameworks such as Apache Spark NLP.

The studies presented multiple research problems from different perspectives using Transformer and convolutional neural network (CNN) models in conjunction with Big Data techniques. Only limited attention has been given to trending tweets, especially following recent restrictions on access to Twitter/X data. Apache Spark is now widely used in domains and areas with large amounts of data; however, its application to social media sentiment analysis remains very limited. Consequently, a specific research gap remains in integrating Spark NLP for scalable tweet preprocessing and in systematically comparing CNN and Transformer models on trending-topic tweets. This study addresses this gap by combining Spark NLP preprocessing with TextCNN, BERT, DistilBERT, and RoBERTa to assess their contextual accuracy and computational efficiency on dynamically collected Twitter data.

1.3. Objectives and Tasks

This study examines extensive textual datasets from different contexts and trending topics [3]. The process begins with collecting daily trending topics from X platform, followed by randomly gathering tweets related to these topics over a specific period.

Given the moderate dataset size, a robust text-processing framework is used to efficiently manage and process the dataset. After data collection, preprocessing techniques using Spark NLP were applied to clean and normalize the tweets. For sentiment classification, we train and fine-tune TextCNN, BERT, DistilBERT, and RoBERTa on the processed dataset. We evaluate their performance using metrics such as accuracy, F1-scores, classification reports, and confusion matrices [9] to analyze training duration and computational efficiency for both CNN and Transformer-based architectures.

The primary goal of this research is to identify an appropriate model for categorizing the sentiment of trending Twitter/X topics. The selected model should achieve the highest overall accuracy and a strong F1-score, which provides a balanced evaluation of precision and recall in an imbalanced dataset. The model that achieves the best performance on these metrics is considered the most suitable choice for this task.

The remainder of this paper is structured as follows: Section 2 presents the methodology and implementation, including the data collection process, preprocessing pipeline, and baseline sentiment classifier for the model. In addition, it presents and justify the choice of the models by outlining the theoretical background of TextCNN and Transformer-based architectures. Section 3 presents the comparative results for TextCNN, BERT, DistilBERT, and RoBERTa. Section 4 discusses the findings of this comparison and identifies the top-performing model.

2. Materials and methods of research

In recent years, NLP (Natural Language Processing) has captured the attention of many researchers and engineers. NLP is a subfield of computer science and artificial intelligence that relies on machine learning to simplify understanding and communication between humans and computers. NLP has been applied in various domains to help resolve problems or analyze certain situations, thereby forecasting the future evolution of products or services and identifying potential drawbacks. In this context, we shed light on the sentiment analysis of tweets collected from X.

The increased focus in sentiment analysis in recent years by both industry and academic community is driven by the goal of determining public sentiment from this data for businesses, policymakers, and researchers to understand public opinion, track emerging trends, and make informed decisions [13]. The pipeline shown in Fig. 1 will encompass several steps, each of which will be explained in detail below.

2.1. Data collection

The data collection process (Fig. 1) begins by creating an account on X (formerly Twitter) and selecting the appropriate subscription plan for tweet collection. We opted for the Basic Plan for this project. Then, we connect to the X API using the Tweepy Python library and the API key provided by the X Developer Portal.

In this phase, we employed a unique approach by collecting data related to a variety of trending topics. To identify the most relevant and up-to-date topics, we extracted data from Trends24, a platform that provides real-time trending topics on X, using BeautifulSoup for web scraping. We retrieve and rank these trending topics based on their popularity, filtering out non-English topics and retweets for consistency in the sentiment analysis.

This approach enabled us to gather data about the most current global events, emerging crises, and phenomena across multiple domains, providing insight

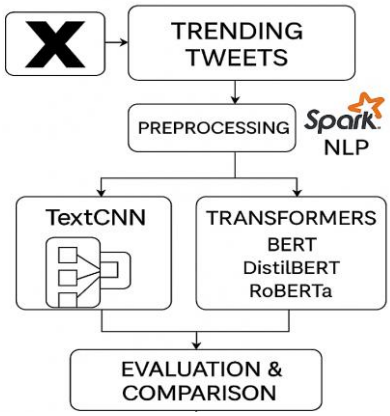

Fig. 1. Tweet classification pipeline

into topics that are gaining momentum. Once the trending topics for the day were identified, tweets related to these topics were randomly collected:

- We construct search queries using each trending topic, ensuring that tweets are in English and excluding retweets to avoid duplication.

- Using the X API, a random number of tweets per topic (between 10 and 20) is retrieved to maintain a balanced dataset.

- We store text content for each retrieved tweet.

- Due to the API rate limits introduced after Elon Musk's acquisition of Twitter and given that we subscribed to the Basic Plan, we are limited to collecting a maximum of 15,000 tweets per month and 15 requests per 15 minutes to stay within the allowed limits.

- The collected tweets are stored in a structured Pandas DataFrame.

Figure 2 illustrates the distribution of trending hashtags related to the main topics in our dataset.

The Top 20 trending hashtags presented in Fig. 2 are a mix of entertainment and celebrity topics (#beckysangels), television shows (#bbb25, #GrandeFratello), sports and events (#wweraw, #superbowl), and global discussions (#elonmusk, #trump) that went viral on the X platform from January 12, 2025, to February 17, 2025. These hashtags are directly linked to real public trends during the data collection window, including global social, political, and cultural events.

Each day within this period, we collected the top 10 trending hashtags and randomly retrieved tweets associated with them. The total number of tweets obtained in this one-month was 15 000 tweets in the dataset because we did a subscription for API basic plan.

Fig. 3 provides an overview of the distribution of the top 10 most frequent hashtags collected in January 2025, along with the dates on which each reached its highest activity on Twitter. Some hashtags were repeated over several days, but their frequency gradually decreased. The results shown in Fig.3 confirm the diversity of topics in our dataset, reflecting real-world Twitter trends. For instance, hashtags like #tiktok and #trump coincided with major social and political events this year, which matches the official decision of the U.S. government to implement restrictions on TikTok in mid-January 2025.

Examples of tweets from our dataset: For instance, a **positive** tweet in the corpus reads, "*Have a fantastic day! #FridayFeeling #FridayVides."* A **neutral** example is, "*Yesterday I announced a new public awareness campaign to combat the spread of meth across Wisconsin*." In contrast, a **negative** tweet example expressed personal frustration: *"I hate that finding funny TikTok creators as the app is dying".* These examples clearly illustrate the range of content and sentiment captured in the dataset.

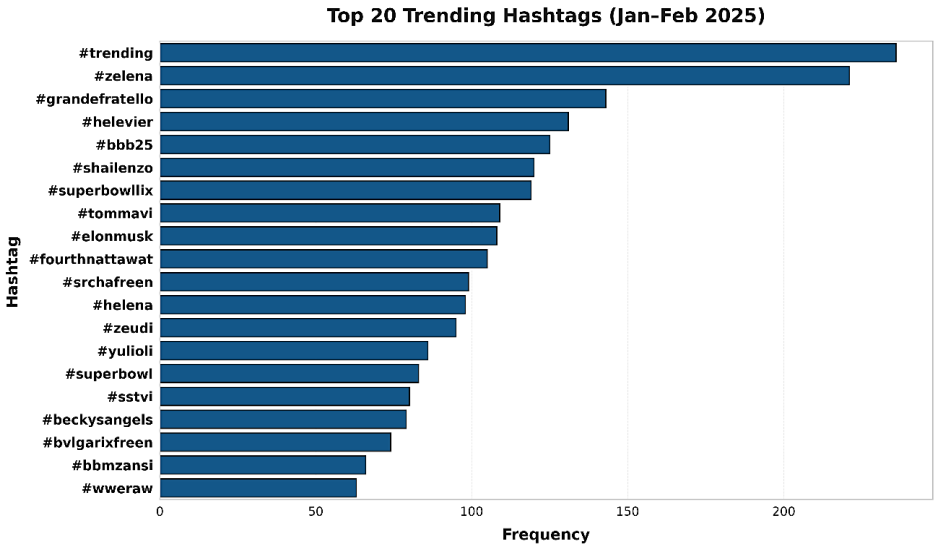

Fig. 2. Top 20 trending hashtags from January to February 2025

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

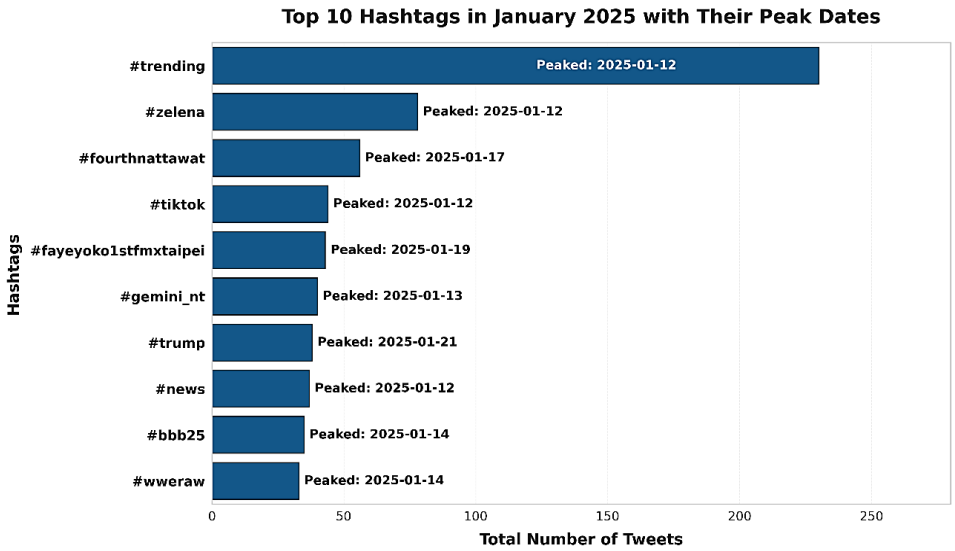

Fig. 3. Top 10 distributed hashtags in January 2025

The dataset contains approximately 15,000 tweets, which is moderate in size by "Big Data" standards. The use of the Big Data framework Spark NLP is justified by its ability to preprocess and analyze high-velocity social media data, thereby ensuring a modular, scalable processing pipeline. In this study, we focus on the use of the Big Data framework to combine velocity and variety, even though the volume aspect was not clearly represented. This setup prepares the pipeline for future expansion to larger and real-time tweet streams with minimal modification.

2.2. Data preprocessing

2.2.1. Introduction and description of NLP

In recent years, the scale of data has been massively growing, being a key factor of the Big Data scenario, which presents high volume, velocity, and variety of data that is too large for traditional applications to handle and requires new high-performance processing [14]. In this context, the integration of a scalable data-processing pipeline is essential to take advantage of the patterns, connections, and trends contained in these vast repositories across diverse data sources.

The raw tweets collected on X via its API often produce highly unstructured and noisy datasets, due to people's spontaneous use of social media [15], which can be emojis, emoticons, hashtags, user mentions, and common web elements, such as email addresses and URLs. In addition, they may include phone numbers, percentages, timestamps, dates, and generic numerical values. Therefore, raw Twitter data must be preprocessed to create a dataset that can be easily learned by various classifiers [15].

Data preprocessing is a vital step in Twitter-based sentiment analysis, as it transforms raw, unstructured tweets into a clean, analyzable format, reducing complexity and improving model performance [16]. It encompasses various techniques, including data preparation and reduction methods.

Once the data preprocessing stage is successfully completed, the resulting dataset provides a reliable, well-structured foundation for applying deep learning algorithms. In the following, we introduce Apache Spark NLP for data preprocessing and outline the overall steps for cleaning the dataset.

2.2.2. Apache Spark NLP

Over the last few years, Apache Spark has established itself as the standard tool for scalable text processing. It provides a high-level API for processing vast amounts of data in parallel, including tasks such as removing unwanted characters and stop words, and converting text to lowercase. A practical application of Spark NLP would involve presenting a holistic view of each tool's suitability for processing social media data in real-time and for data cleaning and preprocessing workflows using Apache Spark and Python [17].

Spark NLP is an open-source library that provides a unified framework for all NLP processing tasks. It stands out as the only library capable of scaling up training and inference in any Spark cluster, leveraging transfer learning, applying the latest and greatest NLP algorithms and models, and delivering high-quality enterprise-grade solutions [18]. Spark NLP aims to outgrow older libraries for production use in software systems. The library features straightforward, efficient, and accurate NLP tools for ML pipelines, designed to scale seamlessly in a

distributed environment. The upgraded Spark NLP pipeline incorporates the following NLP steps:

Document Creation:

To get through the process in Spark NLP, we first need to transform raw data into Document type.

Tokenizer:

When working with text data for analysis or building machine learning models, the first step is to transform the text into meaningful features that the models will use later. This step is called tokenization. Its main goal is to split text into fragments, called tokens, that can be words, characters, emojis, sentences, or other chunks of text, generating a list of tokens that enables further NLP tasks.

Normalizer:

The Normalizer annotator in Spark NLP performs text normalization on the data. It is used to remove all dirty characters from text following a regex pattern, convert text to lowercase, remove special characters, and punctuation, and transform words based on a provided dictionary. The Normalizer improves the accuracy and quality of the analysis and models. We applied the following cleanup patterns to the collected tweets to ensure a clean and normalized version of our data:

- **"@\\w+":** Users frequently refer to others in their tweets by placing an @ symbol before the intended username [19]. The normalizer identifies and removes X mentions from the input text by incorporating this expression into the code, reducing noise from usernames that do not contribute to the tweet's meaning.

- **"https?:**\\/\\/\\S+**":** Uniform Resource Locators (URLs), do not aid in text classification and may even mislead the model if they include conflicting words [19]. So, it is better to normalize all tweets by adding the above configuration.

- **"#\\w+":** The regular expression is used to match hashtags in tweets and remove them because hashtags often contain metadata rather than contributing to the sentiment or the main meaning of the text.

- **"[^a-zA-Z\s]":** The regular expression is used to remove special characters, numbers, emojis and punctuation while keeping only letters (both uppercase and lowercase) and spaces.

- **"\\b[a-zA-Z]\\b":** This expression is designed to match single-character words consisting solely of an alphabet letter that cannot reflect the sentiment or the meaning of the text.

- **"(.)\\1{2,}":** This configuration is used to detect and match repeated characters that appear three or more times in a row because when dealing with social media text, people often elongate words for emphasis such as "soooo happy".

Stop Words Cleaner:

The process of cleaning words in Spark NLP involves removing commonly used words from text data. These words such as “like”, “the”, “a”, “and”, “in” are meaningless and can impede the performance of many natural language processing (NLP) tasks. Eliminating these stopwords can greatly enhance the accuracy of these tasks and optimize resources usage for subsequent text processing.

Lemmatization:

Lemmatization is one of the most common pre-processing tools used in Natural Language Processing and artificial intelligence. It aims to convert a particular word to its root word while considering its meaning. It helps refine textual analysis, information retrieval, and language understanding tasks.

Finisher:

The spark NLP Embeddings Finisher allows users to complete the text data-generated embeddings. This step is crucial for preparing the embeddings for downstream tasks to ensure compatibility within the spark NLP ecosystem.

The dataset used in this study is modest in size due to API limitations. The choice of Spark NLP for data preprocessing is justified not solely by data volume but also by the nature of tweets, which are typically high-velocity, high-variety, and unstructured. Adopting this framework ensures scalability and readiness for larger datasets.

After completing the pre-processing steps, a pipeline that includes the necessary stages is built. When the pipeline is constructed, it is fitted to the collected data and transformed to obtain clean and normalized results. Following this transformation, the dataset was reduced to 12 422 unique and valid tweets. We convert the Spark DataFrame into a Pandas DataFrame, which facilitates exploratory data analysis, visualization, and integration with machine learning models.

The distribution of keywords within our cleaned dataset is visualized in Fig. 4 as WorldCloud. The presence of political and conversational terms in the visualization suggests that the dataset captures diverse trending topics, emotional expressions, and user interaction characteristics that capture the dynamic nature of trending Twitter discussions.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Fig. 4. Word cloud of collected tweet

Since the dataset is small and does not require distributed computation, scalability was not experimentally confirmed in this study. The main goal was to ensure the framework's choice is valid and robust, regardless of the tweet volume that may be involved in future extensions of this work, and to provide a preprocessing pipeline that remains applicable and scalable to larger scenarios. To justify the choice of Spark NLP, we executed three preprocessing methods (Spark NLP, NLTK, SpaCy) under the same GPU A100 (Google Colab Pro) to demonstrate the strengths of Spark NLP over other python libraries. Spark NLP completed its execution process in **2.89 s**, followed by NLTK in **3.49 s**, and SpaCy in **48.75 s.** The Apache Spark NLP framework achieves faster preprocessing and scales efficiently, ensuring reproducibility by preprocessing millions of tweets across extended workloads using the same pipeline with only a slight increase in runtime. Apache Spark DataFrame enables a fully reproducible pipeline that can be easily adapted and applied to Big Data environments. In contrast, the performance of NLTK and SpaCy degrades more rapidly as the database size increases.

2.3. Lexicon-Based Sentiment Analysis with VADER

To determine the polarity of tweets, we use VADER (Valence Aware Dictionary and Sentiment Reasoner), a lexicon and rule-based sentiment analysis tool specifically tuned to sentiments expressed in social media. VADER is more advanced and generally better for tweets because it considers context, negation, and intensity modifiers. We start by applying the VADER Sentiment Analyzer to the tweets in the dataset to obtain each tweet's sentiment. VADER assigns sentiment scores to each word in a text, sums up the sentiment scores of all words, and scales the score between -1 and +15. If the compound score, which represents a single score for the overall sentiment of a text, is less than -0.05, the sentiment is classified as negative; if the compound is between -0.05 and 0.05, the sentiment is neutral; if the compound score is greater than 0.05, the sentiment is positive. Fig. 5 shows the sentiment distribution of the tweets, and Fig. 6 shows the proportion of each polarity after applying the VADER sentiment analyzer. Out of **12 422 tweets**, **45.4%** were classified as positive, **28.6%** as neutral, and **26%** as negative. All sentiment categories are sufficiently represented, supporting fair and unbiased model training and evaluation. This distribution demonstrates that the data are suitable for reliable sentiment classification tasks.

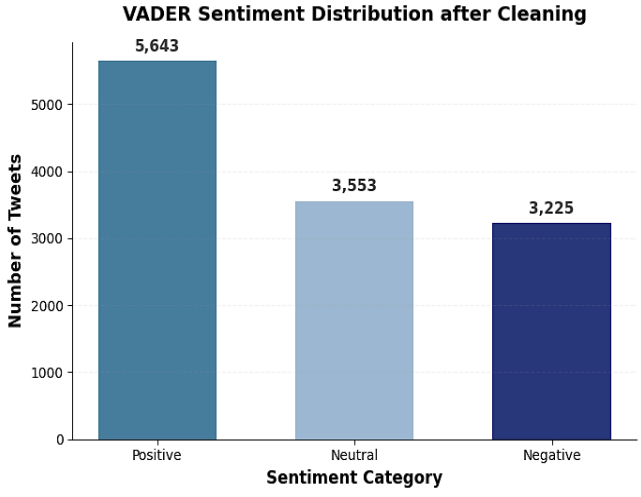

Fig. 5. VADER Sentiment distribution of tweets

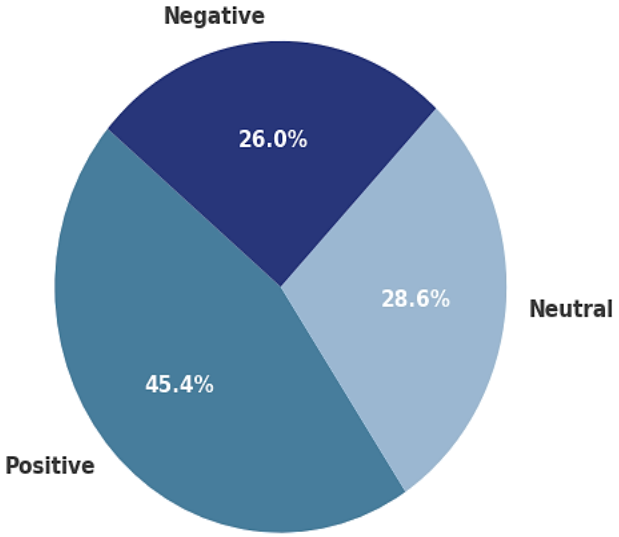

Fig. 6. Proportion of VADER Sentiment

A total of 3,475 tweets were manually annotated as positive, negative, or neutral by a single annotator to allow the models to learn natural human sentiment, rather than VADER logic that misclassifies sarcastic expressions, for example. A set of verifications was implemented to ensure reliability, starting by double-checking ambiguous tweets that were confusing and difficult to judge, such as sarcasm, mixed sentiment, or that lacked context, so we reviewed them for a second time in a separate interval. For unclear cases, we asked another author for an additional opinion to determine the exact polarity of the tweets. To avoid contradictions in classifying sentiment, we ensured that similar sentences with similar linguistic structures received the same labels. By adopting this approach, we reinforce the internal coherence of our manually annotated dataset, even though we rely on a single primary annotator.

We combined automated VADER scores with manual human evaluation to overcome the limitations of purely lexicon-based annotation. Initially, VADER produced a polarity score for all 12 422 cleaned tweets. Subsequently, a subset of 3 475 tweets, representing approximately 28% of the corpus, was manually relabeled, considering contextual cues, sarcasm, and sentiment intensity that VADER might misinterpret. The models were first pretrained on VADER-labeled tweets and subsequently fine-tuned and evaluated on human-annotated tweets. This approach forces the models to align with human judgment rather than lexicon-based polarity. Fig. 7 shows the new distribution of tweets after the manual human annotation of the subset.

Fig. 8 compares VADER-predicted sentiment labels with human annotations for the subset of 3,475 tweets. The diagonal cells (977, 1 017, 952) represent the agreements between the two evaluators. We can see that 125 negative tweets were incorrectly labeled as positive and 100 positive tweets were classified as negative. These confusions arise in tweets containing sarcasm or mixed emotions, where lexical polarity cannot capture the exact sentiment. Overall, the agreement rate of **84.78%** in the subset labels confirms that VADER shows strong consistency with human annotation, although human labeling remains essential for refining training models and ensuring contextual accuracy in sentiment analysis. Table 1 presents examples of misclassified tweets by the VADER tool. These cases reveal key weaknesses of lexicon methods, including the failure to detect implicit negativity or urgency and misinterpretation of contextual expressions. This further justifies the addition of the human annotation phase, as manual review can better detect sarcasm, idiomatic language, and nuanced emotional tones that automated tools may overlook.

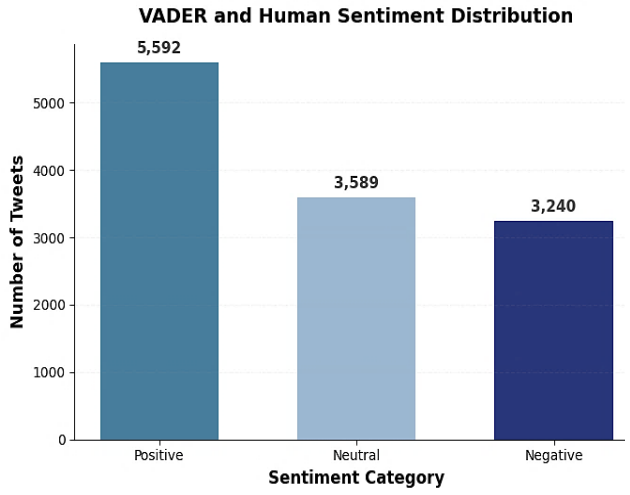

Fig. 7. VADER and Human Sentiment distribution

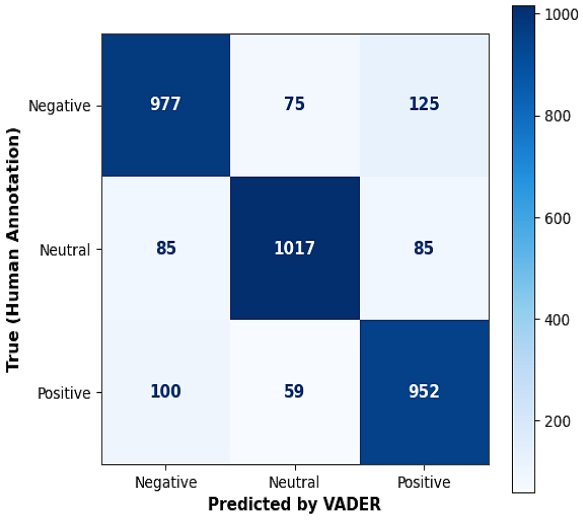

Fig. 8. VADER vs Human Sentiment Labels

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Table 1

Examples of VADER-Predicted and Human-Annotated tweets

TWEET (ANONYMIZED FOR PRIVACY)	VADER LABEL	HUMAN LABEL	INTERPRETATION
"The price paid now becomes the sustainable investment later."	Negative	Positive	VADER misses the constructive financial context expressing optimism about sustainability
"Save Wisconsin!"	Positive	Negative	The human annotators correctly recognized the urgency and concern implied by crisis-related plea.
"FAA grounds SpaceX's Starship after mid-flight explosion, reports property damage on Turks and Caicos"	Positive	Negative	VADER over-weighted terms like "reward" and "program", classifying the tweet as positive while ignoring the negative event.

2.4. Model presentation and theoretical background

2.4.1. Models' justification choice

We selected TextCNN over recurrent-based alternatives, such as LSTM and GRU, because of its one-layer convolutional design with multiple filter widths and global max pooling, enabling much faster training on short, noisy texts for sentiment classification with fewer parameters. In contrast, Transformer models (BERT and RoBERTa) leverage large-scale pre-training to capture deep contextual representations. DistilBERT was included to examine the size-speed trade-offs by compressing these representations into a smaller and faster model. This selection benchmarks a lightweight baseline vs heavyweight state-of-the-art and guides the choice of the real-world model.

2.4.2. Convolutional Neural Networks CNNs for NLP

Convolutional Neural Networks (CNNs) are a type of deep learning model that are traditionally used in image processing [20] and are designed to process complex data formats, which are often represented as multiple arrays. The multi-layered network identifies complex features within data, such as characteristics extracted from images and text. Recently, the application of CNNs has been extended to text-related problems alongside image classification, object detection, and other fields. TextCNN is a powerful text classification model that categorizes text data into various classes. The TextCNN model applies multiple convolutional filters to word embeddings to capture different n-grams and patterns in the text. Each filter produces a feature map that is then passed through a pooling layer to retain the most important features. The resulting feature maps were flattened and passed to a fully connected layer for classification. Using the softmax activation function, we obtain the final output. TextCNN is suitable for diverse text classification problems because it effectively captures local features and patterns in the text.

2.4.3. Transformer models

The birth of Transformers was a turning point in the evolution of deep learning models. In the pre-Transformer era, numerous models dominated NLP. These models had several limitations and weaknesses that prevented them from reaching high-performance results, including inefficiency in processing large sequences, reliance on sequential computation, and difficulty capturing long-range dependencies, which made the training slow and resource-intensive. Transformers were introduced to overcome these limitations by enabling parallel processing and capturing long-range dependencies more easily. The key innovation was the self-attention mechanism, which allows the model to learn the relationships between distant parts of a sequence and enhances its ability to process and understand intricate data. BERT, DistilBERT, and RoBERTa are considered the most well-known Transformers models.

- BERT (Bidirectional Encoder Representations from Transformers): BERT is a model that has considerably advanced language processing tasks. BERT is designed to pretrain deep bidirectional representations from unlabeled text by simultaneously considering left and right context across all layers, enabling it to capture more profound semantic relationships between words, as shown in Fig. 9.

E_1, E_2 …. En is an input sequence passed into the Transformer architecture, where each token undergoes embedding, self-attention, and feedforward transformations within multiple stacked Transformer blocks. These layers refine each word's contextual representation, and the output embeddings

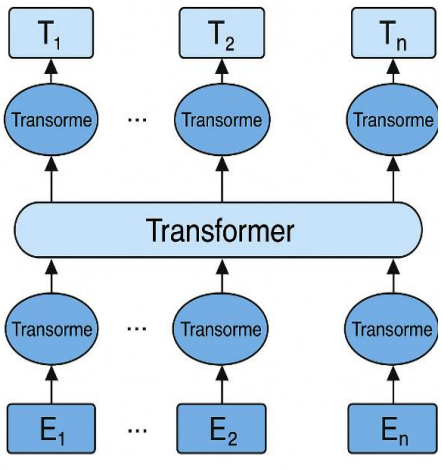

Fig. 9. BERT architecture

T_1, T_2, …...Tn encode the words' contextual meaning within the sequence. The model incorporates the following two training strategies:

1. Masked Language Model (MLM): 15% of the words in a sequence are masked, prompting the model to predict the initial value of these masked words based on the context detected from the unmasked words in the sequence.

2. Next Sentence Prediction (NSP) is a training objective that helps the model to capture inter-word dependencies and predicts whether the second sentence logically follows the first one [21].

The input representation of bidirectional encoder representations from transformers (BERT) is the sum of three key components, as shown in Fig. 10:

1. Token embeddings: This contains token IDs for each word in a sequence, including special tokens [CLS] at the beginning and [SEP] at the end of the sequence to separate sentences [7].

***2.* Segment embeddings:** Segment or sentence embeddings. Each segment has its own embedding separated by the [SEP] token [7], which helps BERT differentiate between sentence A and sentence B in paired-input tasks.

3. Position embeddings: This indicates each token's position within the sequence [7].

- DistilBERT (Distilled BERT): DistilBERT is a distilled version of BERT and has the same general architecture as BERT but with fewer layers and optimizations for efficiency, as shown in Fig. 11. DistilBERT is a lighter and more efficient version of BERT. It was developed using a technique called knowledge distillation, in which a smaller model is trained to mimic a larger model's behavior, retaining 97% of BERT's performance while being 60% faster and 40% smaller. In contrast to BERT, DistilBERT removes the Next Sentence Prediction objective and focuses solely on Masked Language Modeling. Additionally, it uses distillation loss, which helps it mimic BERT's predictions while remaining computationally efficient.

-RoBERTa (Robustly optimized BERT approach): RoBERTa is an optimized version of BERT model. It employs dynamic masking, generating a different mask pattern for each input sequence during model training [7], which is one of the key improvements in RoBERTa over BERT. In addition, RoBERTa is trained with larger mini-batches, more data, and longer training times, and it removes the Next Sentence Prediction task, which was originally part of BERT's pretraining. As illustrated in Fig. 12, these changes help RoBERTa achieve high performance on many NLP tasks compared to the original BERT model.

3. Results and Discussion

3.1. Model Implementation and Training

A two-stage weakly supervised fine-tuning approach was adopted to ensure methodological consistency and fair comparison. All models (TextCNN, BERT, DistilBERT, and RoBERTa) were trained using the same two-stage strategy. In the first phase, each model was pretrained on the full dataset (12 422 tweets) labeled with VADER tool to allow the model to learn general sentiment features. In the second phase, the pretrained models were fine-tuned and evaluated on the manually annotated subset (3 475 tweets) to align their predictions with human-annotated sentiment.

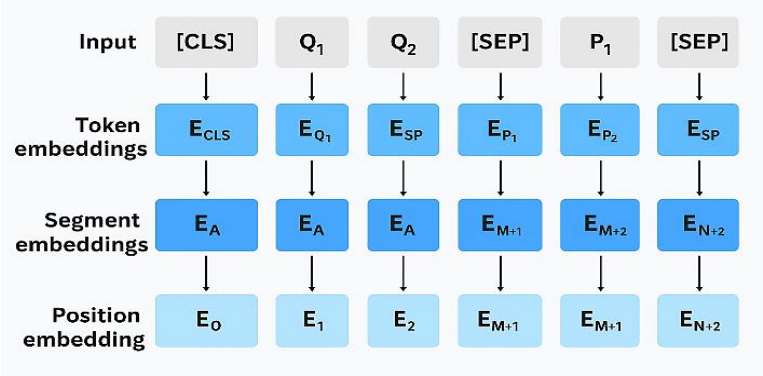

Fig. 10. BERT input representation

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

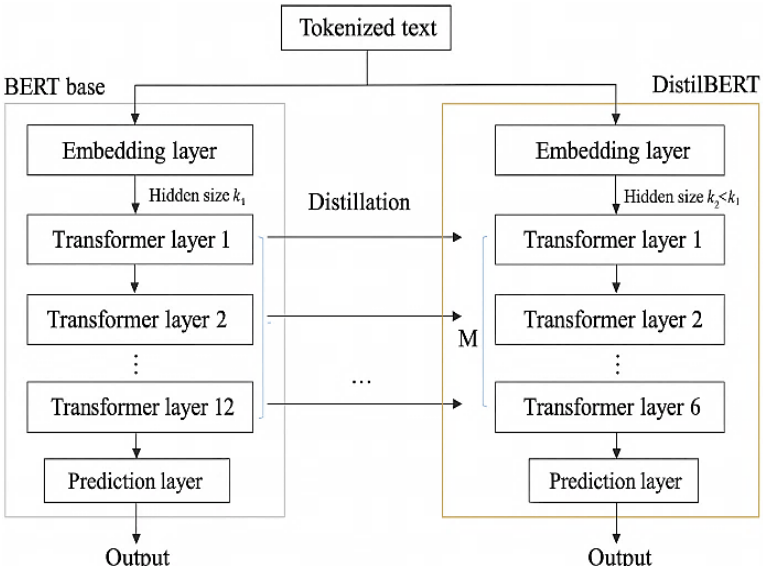

Fig. 11. Comparison between BERT vs DistilBERT architectures

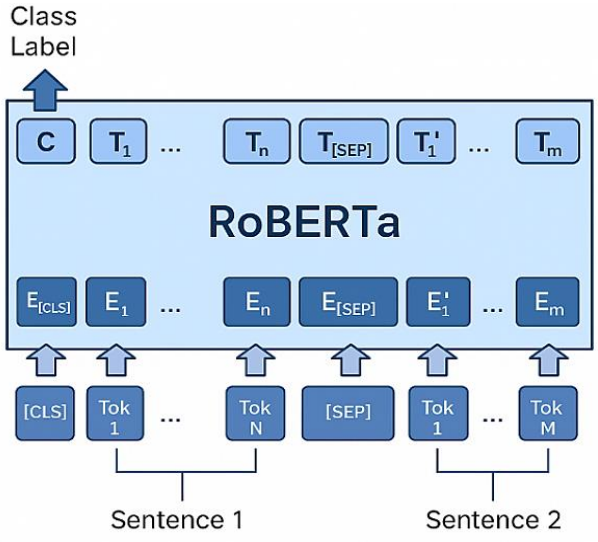

Fig. 12. RoBERTa architecture

We start the training pipeline by fixing the random_seeds of Python, NumPy, and TensorFlow environments to guarantee a reproducible workflow. In the pretraining phase, VADER-labeled dataset (12 422 tweets) was split using a stratify argument to ensure a balanced class distribution, preventing biased training and evaluation: 80% in the training sets (9 937 tweets), 10% for validation (1 242 tweets), and 10% for testing (1 243 tweets). In the fine-tuning phase, we used the human-annotated subset that contains 3 475 tweets. A new stratified split was applied with 80% for training and 20% for testing (695 tweets) to evaluate the model only on this subset. Although the human-labeled tweets constitute a part of the VADER dataset, no overlap was captured between the two phases. However, the final evaluation uses a portion of the human test subset that was never used in any training step to provide unbiased results.

Two stabilization techniques were applied during fine-tuning in the Transformers case to prevent catastrophic forgetting on the limited human-annotated dataset. First, we froze the lower encoder layers for each Transformer model to preserve the linguistic and sentimental features learned during pretraining on VADER-labeled tweets. Second, a learning-rate warm-up over the first 10% of steps was implemented to gradually ramp up the optimizer's learning rate and enable the model to adapt to human-labeled data.

Several experiments were conducted to determine the optimal freezing ratios for each model. Each freezing configuration was evaluated in three random seeds over three runs to judge whether a freezing ratio **0%, 33%, 50%, 67%, 80% is** truly better and overcome the stochastic behavior of Transformer models during the training process, preventing conclusions based on the performance of a single run that can be potentially noisy run. For each freezing ratio, we report the mean accuracy, the standard deviation to measure the stability of the results with respect to randomness, and the average best epoch across three seeds, which provide insights into each model's convergence. For BERT model, Table 2 shows that a freezing ratio of **67%** provides the best fine-tuning configuration for our dataset, with the highest accuracy of 81.10% and the lowest variance across three different seeds, indicating good performance and exceptional consistency. For DistilBERT, freezing **50%** of the layers is the best choice that balances the adaptation of the model and training stability, as shown in Table 3. Table 4 demonstrates that freezing **67%** of the encoder layers of RoBERTa is the optimal fine-tuning configuration for the model, which maximizes its generalization ability, achieving the highest mean accuracy (82.63%) with the lowest variance in three consecutive runs.

Table 2

BERT Freezing ratios comparison

Freeze ratio	Mean accuracy	Std accuracy	Avg best epoch
0%	0.36	0.031	1
33%	0.60	0.153	3
50%	0.68	0.122	3
67%	0.81	0.012	5
80%	0.74	0.064	3

Table 3

DistilBERT Freezing ratios comparison

Freeze ratio	Mean accuracy	Std accuracy	Avg best epoch
0%	0.52	0.098	2
33%	0.50	0.073	5
50%	0.63	0.069	4
67%	0.50	0.133	6
80%	0.60	0.100	5

Table 4

RoBERTa Freezing ratios comparison

Freeze ratio	Mean accuracy	Std accuracy	Avg best epoch
0%	0.49	0.152	2
33%	0.53	0.153	2
50%	0.54	0.112	6
67%	0.82	0.019	4
80%	0.74	0.121	5

• TextCNN Implementation:

Sentiment classification was developed using a TextCNN model applied to the collected dataset. The process begins by mapping sentiments to numerical labels, tokenizing the cleaned tweets, and transforming them into sequences. The TextCNN model is constructed, comprising an embedding layer followed by multiple convolutional layers with ReLU activation and global max pooling to extract n-gram-level features. Dense layers and dropout regularization are used for classification before the final softmax output layer. The model was compiled using the categorical cross-entropy loss function and optimized using Adam, with early stopping based on validation loss (patience=3) to avoid overfitting. The model was trained for 10 epochs with a batch size of 32. To evaluate the model, we used classification metrics and a confusion matrix to visualize the distribution of the correctly predicted sentiment classes. These metrics provide insights into how the model can properly classify sentiments in the class imbalance.

The classification report heatmap in Fig. 13 shows that the TextCNN model pretrained on VADER-labeled tweets achieved 75% accuracy and a macro-F1 score of 0.72, demonstrating strong performance on positive and neutral tweets. However, negative sentiment was the most challenging with a score of 0.50. F1-score, indicating that lexicon-based signals were insufficient to capture context negativity, such as sarcasm. Once the TextCNN model was fine-tuned on 3 475 human-annotated tweets, the model accuracy decreased to 72% due to the higher complexity of human judgments and macro-F1 = 0.72 on the human test set (695 tweets), as shown in Fig. 14. Negative sentiment detection improved (F1= 0.61), proving that fine-tuning helped the model align better with human interpretation. Neutral (F1 = 0.83) and positive (F1 = 0.72) classes remained stable, confirming that the model retained its previous strengths after fine-tuning. Despite the small dataset size, the fine-tuned TextCNN model generalized well by combining lexicon-based and human-based sentiment understanding.

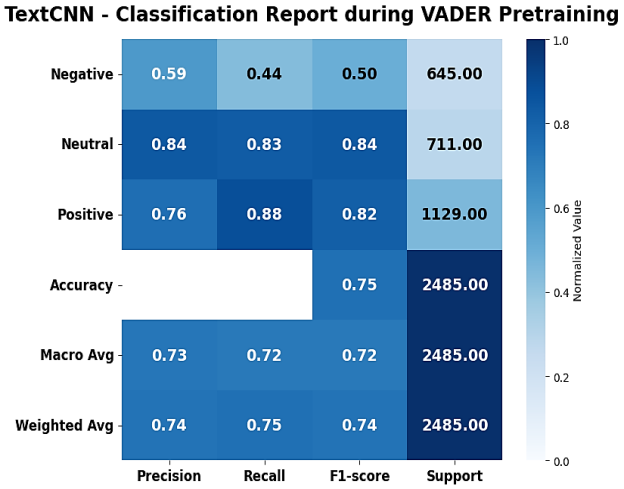

Fig. 13. Classification Report of TextCNN during VADER Pretraining

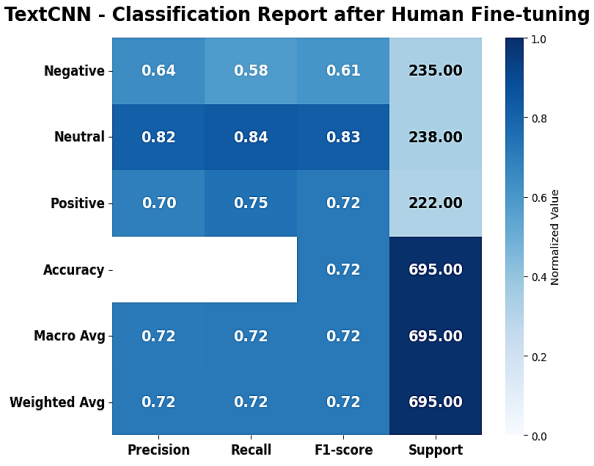

Fig. 14. Classification Report of TextCNN after Human Fine-Tuning

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

The TextCNN performance after the fine-tuning phase on the test set (695 tweets) is summarized in the confusion matrix in Fig. 15. The diagonal values (137, 199, 166) indicate correctly classified tweets. Negative tweets were misclassified as neutral (36) and positive (62) when they contained positive words but expressed frustration. Neutral tweets were mislabeled as negative (29) and positive (10) when they included polite or informative promotional phrasing without emotional intensity. A few positive tweets were misclassified as negative (47); most errors were attributed to contextual ambiguity rather than model weakness. Fig. 16 shows the training and validation accuracy and loss over epochs for the TextCNN model. Early stopping was not triggered during this fine-tuning phase, as the validation loss continued to decrease slightly through the final epoch. The graphs show that the model learns well on the training data while maintaining close validation performance, suggesting it generalizes well and does not overfit. The improvement across epochs reflects a balanced learning rate and consistent optimization process.

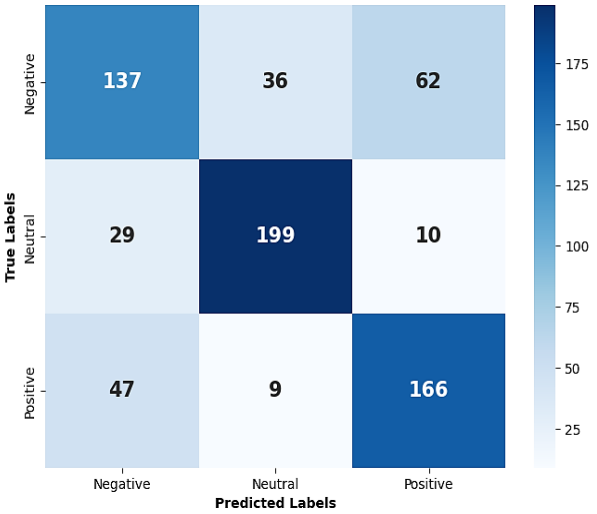

Fig. 15. Confusion Matrix of TextCNN after Fine-Tuning

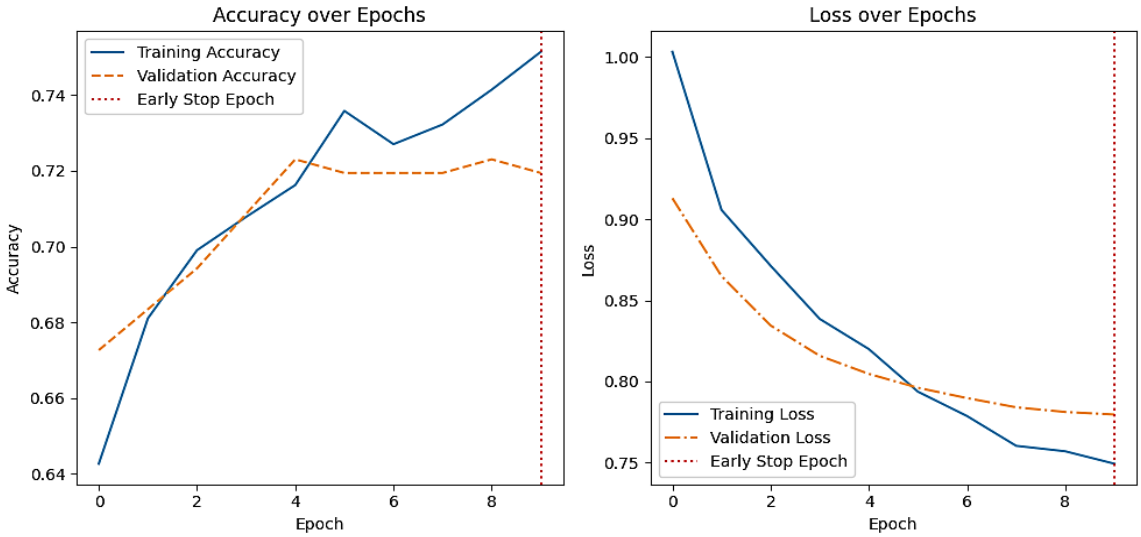

Fig. 16. Training and Validation Performance of the TextCNN Model after Fine-Tuning

• BERT Implementation:

The sentiment analysis pipeline begins by splitting data into training and test sets using stratified sampling to preserve class distribution. Tokenization was performed using a BERT tokenizer to convert them into ready input text to the model. These inputs are then encoded into input_ids, attention_mask, and token_type_ids. A pre-trained BERT model is configured for sequence classification with sentiment classes using Adam optimizer and categorical cross-entropy loss. The model was pretrained on 12 422 tweets for 10 epochs with a batch size of 32, a learning rate of 2e-5, and early stopping with a patience of 3 applied to the validation loss to avoid overfitting. The pretrained model was fine-tuned on 3 475 human-annotated tweets to understand the sentiment interpretation, and the parameters used were as follows: batch size = 16, epoch = 10, learning rate=2e-6, and early stopping (patience = 3) to ensure stable convergence since the dataset is much smaller than the previous one used in the pretraining phase. After training, the model's performance was evaluated using various metrics. Finally, predictions are visualized to assess how well the model distinguishes between different sentiment classes. This BERT model configuration leverages deep contextual understanding of language.

Fig. 17 and Fig. 18 present a comparison between BERT model performance during the VADER pretraining phase and after Fine-tuning on the human-annotated subset. During VADER pretraining, BERT achieved 90% accuracy with a balanced class distribution, confirming that it effectively learned sentiment features from VADER-labeled data. After fine-tuning the human subset, the accuracy reached 84%, indicating robust performance despite the dataset's smaller size (3 475 tweets), with strong recall for Neutral (0.85) and Positive (0.89) tweets. These values reflect desirable refinement rather than performance loss between the two phases. The small decrease in the metrics demonstrates that the model became more selective and context-aware. Overall, the two-stage training successfully preserved BERT sentiment knowledge from VADER and effectively adapted it for human sentiment analysis, confirming the robustness of the weak-to-strong supervision approach in this study.

The confusion matrix in Fig. 19 indicates that the BERT model classified most tweets well within each sentiment category, as presented in the diagonal values (182, 202, 197), indicating a balanced across sentiments. Misclassifications mostly occur between semantically close categories: 33 Negative tweets were predicted as Positive and 20 as Neutral, suggesting that statements may contain words that confuse the classifier. 15 Neutral tweets were misclassified as Positive and 21 as Negative, reflecting the ambiguity of neutral expressions. Only a small overlap is observed between the Positive and other classes, indicating a strong ability to detect positive tone. The matrix confirms that fine-tuning enabled BERT to achieve a well-balanced class discrimination, with a clear dominance of correct predictions. The model effectively integrated human judgment while maintaining a robust sentiment structure learned during the VADER pretraining phase.

The loss vs epochs chart in Fig. 20 shows that the fine-tuned BERT model rapidly converged within the first few epochs. Although the validation loss fluctuated between epochs 1 and 4, reflecting small adjustments in the unfrozen layers as they adapt to human-labeled data, it stabilized thereafter, indicating consistent generalization. The early stopping mechanism was activated at epoch 5, indicating that the model reached its optimal state. The stability of these curves demonstrates the effectiveness of the freezing strategy and the learning-rate warm-up that allowed a controlled adaptation of the model to the new data without sudden changes in its pretrained weights.

• DistilBERT Implementation:

The DistilBERT model is a lighter, faster alternative to BERT. It retains 97% of BERT's language understanding abilities while being more performant. DistilBERT has fewer transformer layers (6 instead of 12), no token-type embeddings, and a smaller hidden size, which significantly reduces training time and memory consumption. In this implementation, a pre-trained DistilBERT base is fine-tuned for sentiment classification. The model accepts input IDs and attention masks and processes them through the DistilBERT encoder to compute the loss and optimize predictions.

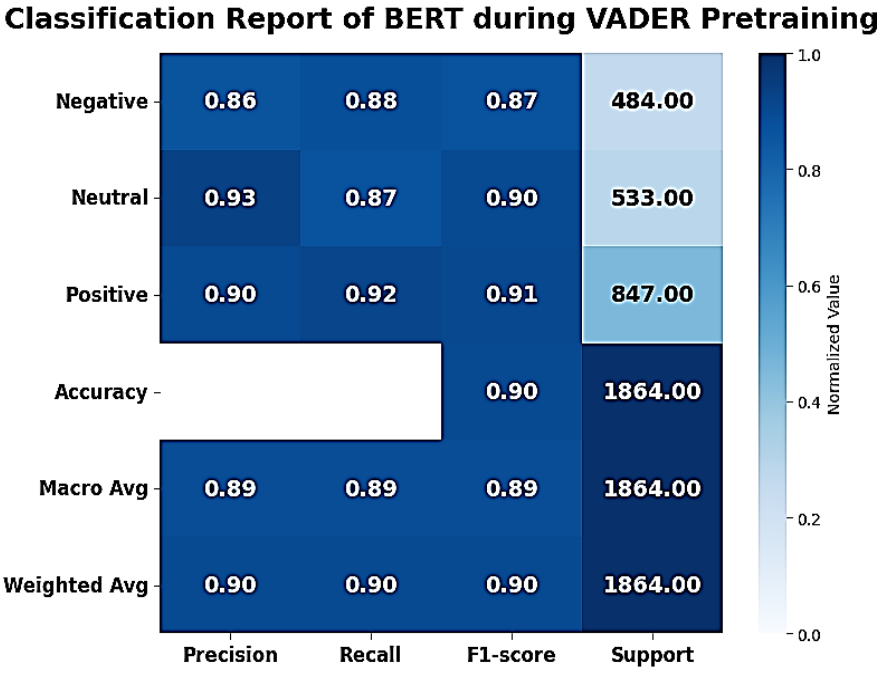

Fig. 17. Classification Report of BERT during VADER Pretraining

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

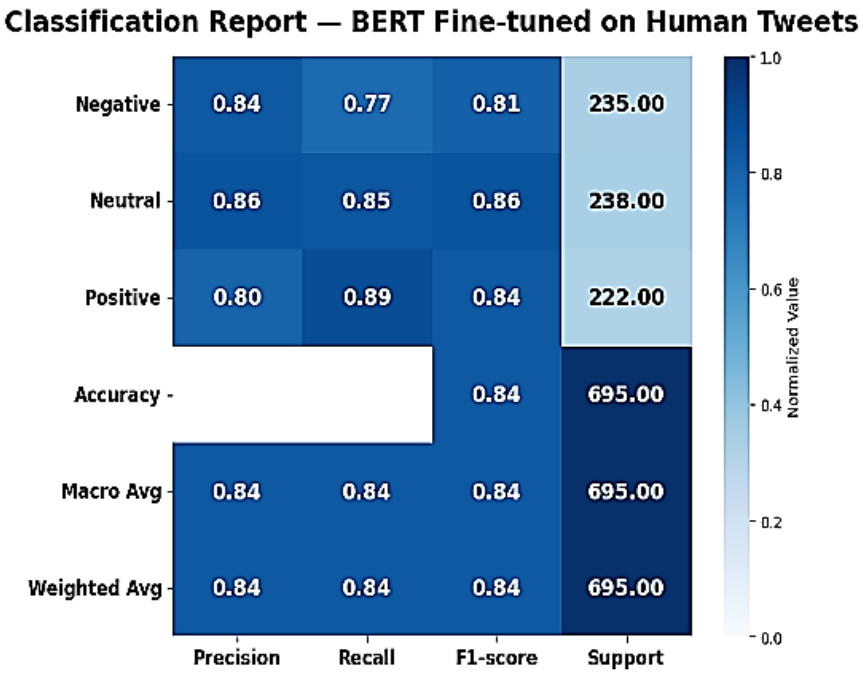

Fig. 18. Classification Report of BERT after Human Fine-Tuning

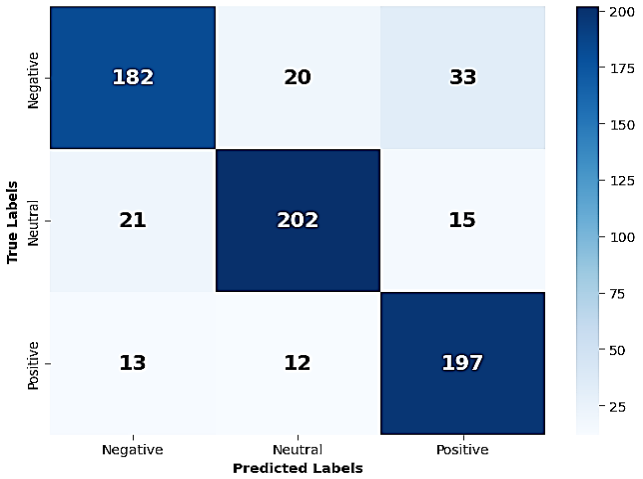

Fig. 19. Confusion Matrix of BERT after Fine-Tuning

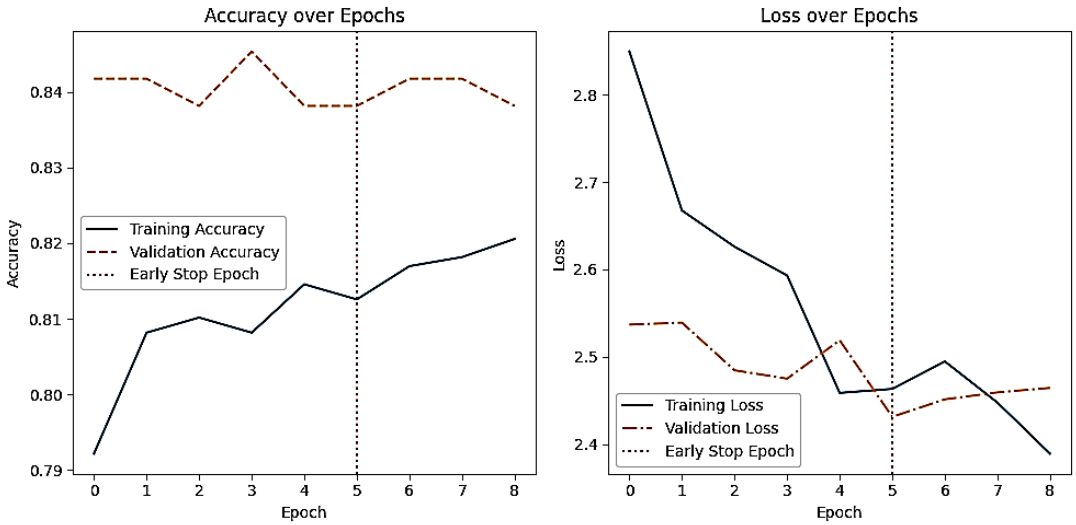

Fig. 20. Training and Validation Performance of the BERT after Fine-Tuning

The model was trained in the pretraining phase with an Adam optimizer (lr≈2e-5) and a batch size of 16. In the fine-tuning phase, the model used a learning rate of 2e-6 and a batch size of 16 to slowly refine its representations and prevent overfitting. DistilBERT is approximately half the size of BERT and uses approximately half the GPU memory. We used early stopping with patience 3 over 10 epochs on the validation loss, and dropout to manage overfitting. This streamlined architecture makes it suitable for quick experimentation and deployment in environments with limited resources while still achieving notable performance.

In the classification report of VADER pretraining phase shown in Fig. 21, DistilBERT achieved an accuracy of 91% with high precision and recall across all classes. This indicates that the model successfully captured the sentiment structure from the VADER-labeled corpus. During the human fine-tuning phase, the classification report in Fig. 22 shows that the model maintained strong performance, achieving an accuracy of 84%. As human annotations introduce greater linguistic diversity and reduce label bias compared to the automated VADER labels, the gap in accuracy between the two phases was expected. Notably, recall for positive (0.90) and neutral (0.84) tweets reflects strong adaptation to contextual and emotional signals absent in lexicon-based labels. The fine-tuning stage enabled DistilBERT to better align with human-labeled sentiment nuances.

The confusion matrix in Fig. 23 indicates that after fine-tuning, DistilBERT achieved well-balanced alignment among sentiment classes. The correctly classified samples fall along the diagonal, with 199 tweets in the neutral and positive classes and 188 tweets in the negative class. The limited error dispersion suggests that DistilBERT learned to manage context dependency in expressions. Misclassified examples are primarily concentrated between adjacent polarities, where sentiment is weakly expressed (mildly positive or negative opinions), as expected in short, informal tweet language.

Fig. 21. Classification Report of DistilBERT during VADER Pretraining

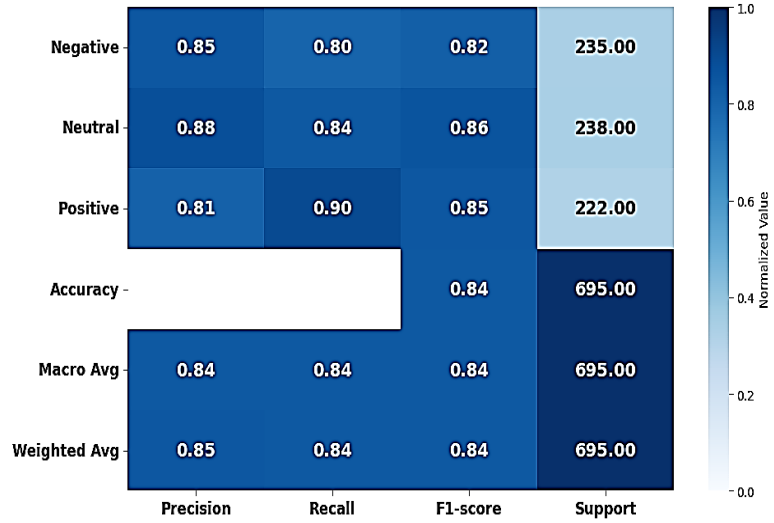

Fig. 22. Classification Report of DistilBERT after Human Fine-Tuning

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Fig. 23. Confusion Matrix of DistilBERT after Fine-Tuning

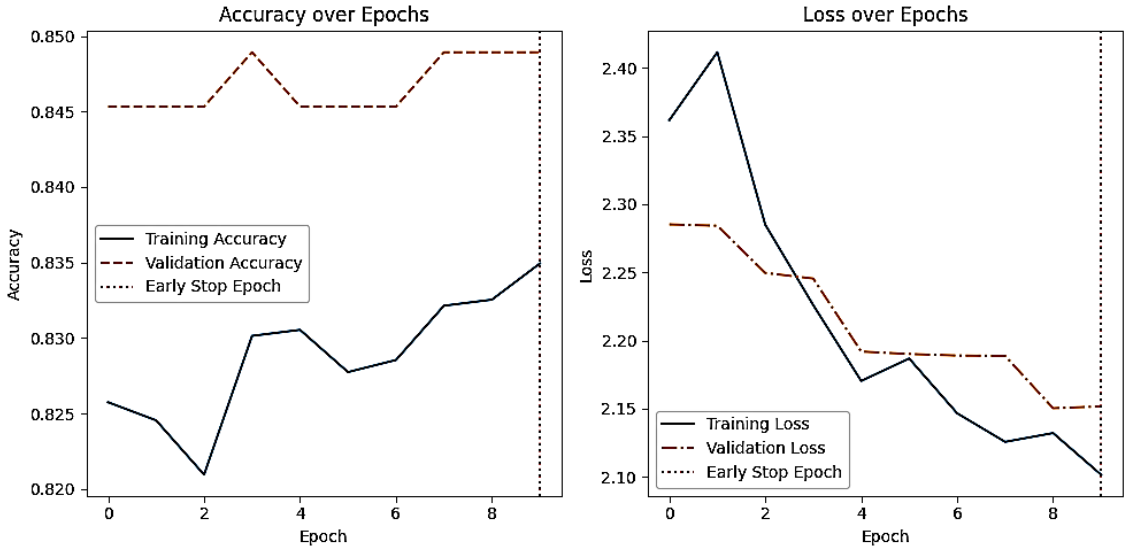

Fig. 24. Training and Validation Performance of the DistilBERT Model after Fine-Tuning

The training and validation losses in Fig. 24 decrease steadily before stabilizing at epoch 9, when early stopping is activated. The validation accuracy was higher than the training accuracy (0.84–0.85), confirming that the model generalized consistently and avoided overfitting. The smooth convergence of both curves highlights the effectiveness of the fine-tuning setup (4 frozen layers, small learning rate, and batch size of 16), allowing DistilBERT to accurately match human sentiment while maintaining its pretrained knowledge.

• RoBERTa Implementation:

To train the RoBERTa model for the sentiment classification task, the first step is to tokenize the input text that requires input IDs and attention masks. Therefore, the data are well encoded, and the model is initialized with the appropriate number of output labels. The model is compiled in the pretraining phase with the Adam optimizer at 2e-5 learning_rate and CategoricalCrossentropy and trained with a batch_size of 32 over 10 epochs using Early Stopping in the training to avoid overfitting with a patience equal to 3. In the Fine-tuning stage, the batch size was updated to 16 and the learning rate to 2e-6 with the same number of epochs and patience, as the dataset in this step has a small size. Performance is monitored using validation loss, and the RoBERTa model is evaluated on test data to assess its predictive quality.

The RoBERTa classification report in Fig. 25 achieved an overall accuracy of 92% during the VADER pretraining phase with balanced precision, recall, and F1-score of 0.92 each. All sentiment classes were correctly recognized, with slightly higher recall for

positive (0.94) and negative (0.92) tweets, confirming RoBERTa's robust ability to capture polarity from lexicon-based VADER labels. The model effectively learned general sentiment patterns before human fine-tuning.

After fine-tuning the 3 475 human-annotated tweets, the accuracy decreased to 85%, with the macro-averaged precision, recall, and F1-score all at 0.85 (Fig. 26). This moderate reduction is expected because human labels capture a more nuanced and sentimental meaning than those generated by VADER.

The confusion matrix in Fig. 27 confirms this impressive performance, indicating that most instances were correctly classified. Positive (201/222) and Neutral (195/238) tweets were recognized with high accuracy, demonstrating the model's ability to capture both emotional polarity and contextual neutrality. Overall, RoBERTa exhibits reliable classification behavior, with errors arising from the ambiguity between the Neutral and Negative expressions.

The model charts in Fig. 28 illustrate the stable convergence of RoBERTa during fine-tuning. Training accuracy progressively rises, while validation accuracy improves quickly in the first epoch and then stabilizes around 82-83%, and the model adapts rapidly to the human-labeled data. After several epochs, the training loss keeps decreasing, while the validation loss stops improving, showing that the model begins to reach its optimal capacity. The model signals the start of overfitting on epoch 7, so the model is stopped to provide the best balance. The figure emphasizes the advantage provided by the freezing strategy (first 8/12 layers) and learning-rate warm-up, which allowed RoBERTa model to learn quickly and manage new examples and unseen tweets well.

Fig. 25. Classification Report of RoBERTa during VADER Pretraining

Fig. 26. Classification Report of RoBERTa after Human Fine-Tuning

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

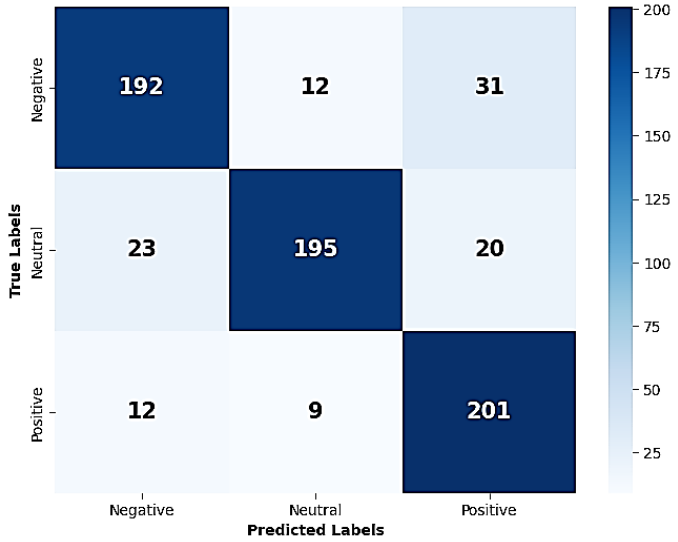

Fig. 27. Confusion Matrix of RoBERTa after Fine-Tuning

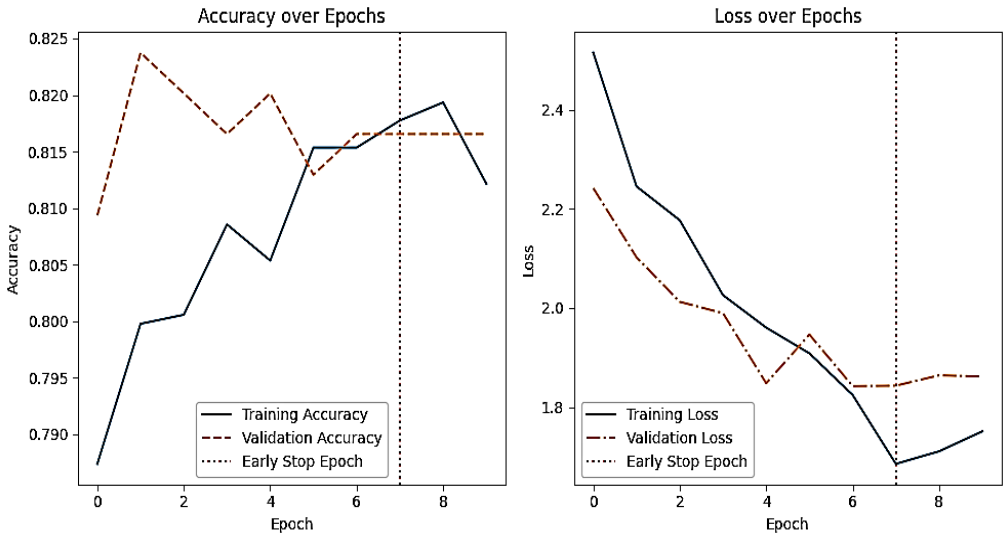

Fig. 28. Training and Validation Performance of the RoBERTa Model after Fine-Tuning

3.2. Baseline: Trained models only on Human Labeled Tweets

In this part of the study, we trained the TextCNN, BERT, DistilBERT, and RoBERTa models on 3 475 tweets that were manually annotated without relying on VADER tool. This methodology was used to evaluate the yields of the proposed two-phase training strategy. The baseline shows substantially lower results due to the small size of the dataset used in this step, as shown in Table 5: TextCNN and BERT models achieved an accuracy of 69% for each, while DistilBERT and RoBERTa performed better in accuracy with 78% and 75%, respectively. The results obtained fully confirm the necessity of the two-phase approach and the improvements it provides to the performance metrics of TextCNN and Transformer-based models, as illustrated in Fig. 29.

Table 5

Models' performance in the Baseline (Only Human-Annotated) training

MODEL	ACCURACY	F1-NEGATIVE	F1-NEUTRAL	F1-POSITIVE	MACRO-F1
TextCNN	0.69	0.66	0.73	0.69	0.69
BERT	0.69	0.72	0.67	0.67	0.69
DistilBERT	0.78	0.79	0.79	0.75	0.78
RoBERTa	0.75	0.77	0.77	0.70	0.75

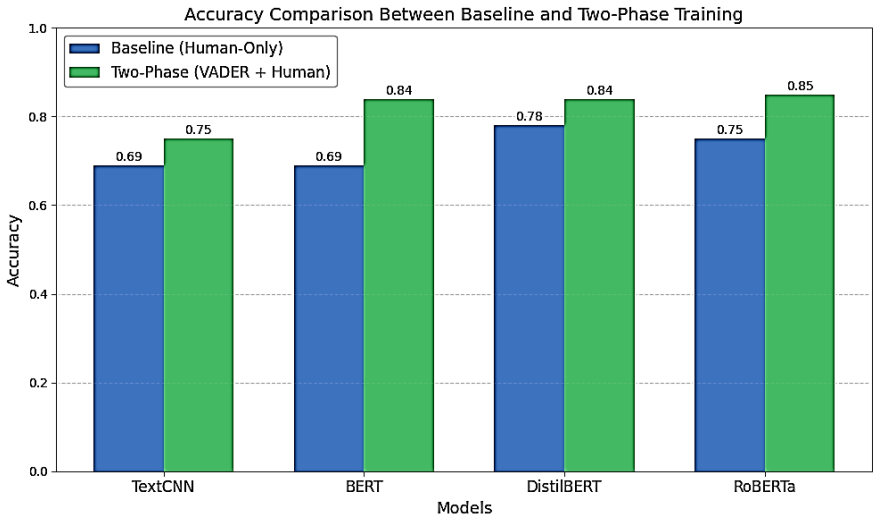

Fig. 29. Comparison of models Accuracy in baseline and two-phase training

3.3. Performance Evaluation

This study presents and compares the performance of four different natural language processing models, namely, TextCNN, BERT, DistilBERT, and RoBERTa, on a multi-class sentiment classification task with three labels: Positive, Negative, and Neutral. The comparison focuses on various evaluation metrics, including Precision, Recall, F1-score, and overall Accuracy, as well as additional indicators, such as confusion matrices and training time, to enhance semantic accuracy across both the VADER pretraining phase on 12 422 tweets and the fine-tuning stage on 3 475 human-annotated tweets. All models were trained under comparable experimental settings during the two learning phases to ensure transparent comparison, as illustrated in Table 3 and Table 4. Minor differences such as the number of frozen layers or batch size were adjusted to match the computational capacity of each model.

Let us start with TextCNN, a lightweight convolutional neural network for text. The model achieves an accuracy of 0.72, but it shows weak performance on the Negative sentiment with an F1-score of 0.61 when trained on a small human-labeled dataset. This behavior is expected given that TextCNN relies only on training data, without a large-scale pretraining, as in the case of the Transformers architecture. The BERT model performed best in the neutral class (F1-score: 0.86) with an overall accuracy of 0.84, indicating that the model is more stable and shows strong generalization across all sentiment classes. DistilBERT gives a competitive performance with an overall accuracy of 0.84 and F1-score=0.84, confirming that model compression preserved most of BERT's strengths while reducing computational costs. Finally, RoBERTa was the best overall performer with 0.85 accuracy and perfectly balanced F1-scores of 0.83 (Negative), 0.86 (Neutral), and 0.85 (Positive). These outcomes confirm that the fine-tuning phase notably improved the contextual alignment of all transformer-based models, while RoBERTa offered the most stable and precise sentiment discrimination.

The analysis demonstrates that RoBERTa is the top performer in terms of accuracy, delivering strong results across all sentiment classes due to its dynamic masking and large-scale pretraining. DistilBERT achieved equivalent performance while significantly reducing training time and computational costs compared with RoBERTa and BERT, making it a compelling choice. Although BERT remains highly competitive, it is more resource-intensive than DistilBERT. TextCNN performs the worst among the models.

Tables 6 and 7 and Fig. 31 provide a benchmark for training configurations and computational performance for TextCNN, BERT, DistilBERT, and RoBERTa models. TextCNN achieved the fastest training time, making it suitable for quick tasks with simple architecture. Although all models used the same maximum sequence length of 100 tokens (Fig. 30) to ensure that every tweet could be represented without truncation while avoiding unnecessary padding, BERT and RoBERTa require longer training times because of their deeper architecture and larger attention layers. DistilBERT offers a balanced alternative. The differences in the number of epochs reflect the early stopping mechanism and each model's complexity. Overall, Transformer models, such as BERT, DistilBERT, and RoBERTa, demand higher computational resources because of their attention-based understanding mechanisms, which are absent in simpler CNN models, such as the TextCNN model.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

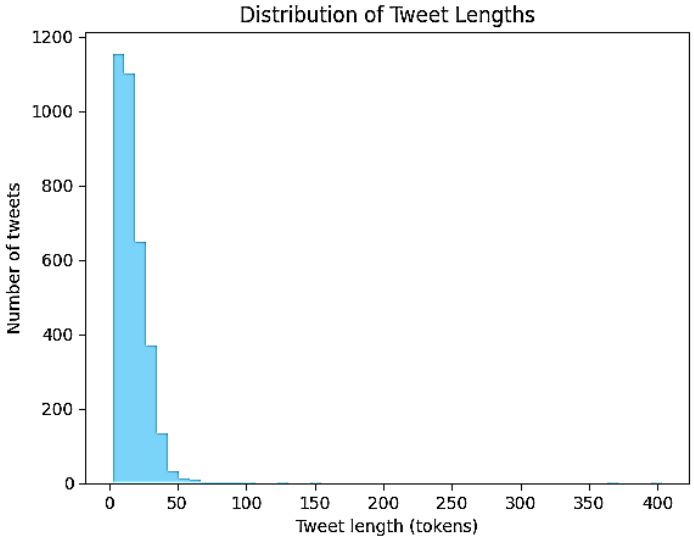

Fig. 30. Distribution of tweet lengths

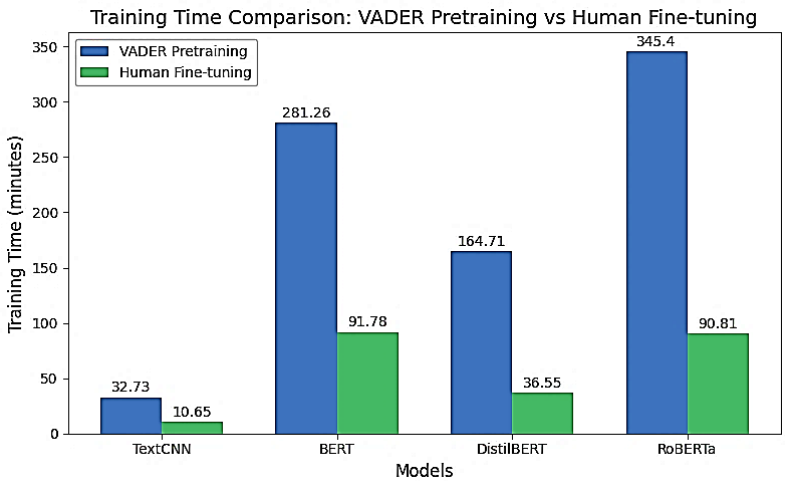

Fig. 31. Training Time Comparison

Table 6

Model training parameters during VADER Pretraining phase (12 422 tweets)

MODEL	BATCH SIZE	LEARNING RATE	EPOCHS	OPTIMIZER	LOSS FUNCTION
TextCNN	32	1e-04	10	Adam	CategoricalCrossEntropy
BERT	32	2e-05	10	AdamW	SparseCategoricalCrossEntropy
DistilBERT	16	2e-05	10	Adam	CategoricalCrossEntropy
RoBERTa	32	2e-05	10	Adam	CategoricalCrossEntropy

MAX SEQUENCE LENGTH	EARLY STOPPING	PRETRAINED MODEL	DROPOUT RATE	HARDWARE	TRAINING TIME
100	Yes (epoch=3)	N/A	0.5	Colab GPU (T4)	32.73 sec
100	Yes (epoch=5)	bert-base-uncased	0.2	Colab Pro (A100)	281.26 sec
100	Yes (epoch=2)	distilbert-base-uncased	0.2	Colab Pro (A100)	164.71 sec
100	Yes (epoch=9)	roberta-base-uncased	0.2	Colab Pro (A100)	345.40 sec

Table 7

Model Fine-Tuning parameters on 3 745 Human-Annotated Tweets

MODEL	BATCH SIZE	LEARNING RATE	TRAINING TIME	FROZEN LAYERS	WARM-UP RATIO	EARLY STOPPING
TextCNN	32	1e-04	10.65 sec	N/A	N/A	Yes (epoch=9)
BERT	16	2e-06	91.78 sec	8/12 (67%)	0.1	Yes (epoch=5)
DistilBERT	16	2e-06	36.55 sec	4/6 (67%)	0.1	Yes (epoch=9)
RoBERTa	16	2e-06	90.81 sec	8/12 (67%)	0.1	Yes (epoch=7)

4. Discussion

The comparative evaluation in this work focuses on TextCNN and Transformer-based models (BERT, DistilBERT, and RoBERTa), examining their architectures and how their results are generated, leveraging the strengths of each. Transformer models outperformed TextCNN because they can capture long-range dependencies and contextual meaning, which TextCNN cannot. A two-phase training strategy was performed that proved beneficial for Transformers. We started by pretraining the models on VADER-labeled tweets, then fine-tuned them on the human-annotated tweets. RoBERTa achieved the highest overall accuracy of 85%, followed by BERT and DistilBERT. The freezing strategy showed that preserving 67% of BERT and RoBERTa encoder layers and 50% of DistilBERT's encoder layer yielded the best stability and adaptation in performance metrics.

Although we used this approach, some misclassifications still occurred during the test phase, as shown in the confusion matrices. Error analysis arises from linguistically complex sentences, such as sarcastic praise, implicit sentiment, and mixed-polarity tweets. For example, "*Am holding an Election Day discussion about the crucial battle in Wisconsin today*" was predicted as Negative after training, but it was classified Neutral in our original dataset. In contrast, "*Let's go! I need an Ulta trip too" is predicted to be Neutral, but its true label is* Positive. This weakness in this approach highlights the limitations of both VADER lexicon-generated labels and the small human-annotated dataset.

These findings suggest that Transformer architectures, especially RoBERTa, are well-suited for sentiment analysis when applied to trending Twitter and X topics characterized by unstructured, informal phrasing. The two-phase training strategy increases both accuracy and macro F1-score for Transformer models, providing additional performance gains that confirm the effectiveness of our proposed approach. Future extensions of this work should include a larger dataset with domain-specific pretraining to move into the Big Data domain and apply the same pipeline with slight adjustments.

5. Conclusion

This study provides a comprehensive sentiment analysis on widely trending tweets collected from the X platform, focusing on both Convolutional Neural Network particularly TextCNN, and Transformer-based models (BERT, DistilBERT, and RoBERTa) to classify tweets into positive, negative, and neutral categories. The preprocessing pipeline was implemented in the Apache Spark NLP library to facilitate and accelerate the scalable, efficient processing of tweet data through distributed computation, followed by exploratory visualizations and charts to gain deeper insights into the dataset's structure and the distribution of sentiment-labeled data. Each model was evaluated based on standard classification metrics and confusion matrices.

The comparative analysis reveals that RoBERTa outperformed the other models, achieving the highest scores across most evaluation metrics and demonstrating strong ability to capture the contextual sentiment of short, informal texts. In addition, RoBERTa consistently demonstrated robust performance across epochs, albeit with higher computational costs. This trend was observed across all Transformer-based models due to the enormous number of parameters and self-attention mechanisms, which require more resources and training time than simpler architectures such as TextCNN.

The study successfully achieved its main objective of improving sentiment classification performance and identifying the model with the highest accuracy across the metrics we evaluated. The results demonstrate the effectiveness of the proposed strategy, which pretrains models on VADER-generated weak labels before fine-tuning on human-annotated tweets. This methodology produced significant performance gains, enhancing the overall accuracy and reliability of the final sentiment predictions, which more closely aligned with human interpretation.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Several promising directions could be pursued for future work. Domain-specific fine-tuning, in which transformer models are adapted to specific contexts and sectors to improve sentiment interpretation, is a key extension. The work can be extended to the management of multilingual tweets, considering the diversity of users on social media.

Contribution of authors: Conceptualization, formulation of research objectives and tasks, methodology description, data collection and preprocessing steps, implementation of models and methods, analysis of results and benchmarking, visualization, redaction of the paper – **Chaimaa Benyamani**; Supervision, guidance on research methodology, review and editing of the manuscript – **Hassan Badi, Imad Badi and Aziz Khamjane.**

Conflict of Interest

The authors declare that they have no conflicts that could have influenced the research, its analysis, or the conclusions presented in this paper.

Financing

This study was supported by the National Council for Scientific and Technological Research (CNRST) PhD scholarship awarded to Chaimaa Benyamani.

Data Availability

The dataset will be available upon reasonable request.

Use of Artificial Intelligence

The author has used artificial intelligence tools within acceptable limits to assist in translating the Abstract, Title, and Author Names into Ukrainian. The IA was not used in conducting research, analyzing data, or presenting results.

Acknowledgements

This work is supported by the National Council for Scientific and Technological Research (CNRST) under the Doctoral Monitoring Program. The authors would like also to thank Abdelmalek Essaâdi University, Tétouan for providing access to research facilities.

All authors have read and agreed to the publication of the final version of this manuscript.

References

1. Mann, S., Arora, J., Bhatia, M., Sharma, R., & Taragi, R. *Twitter Sentiment Analysis Using Enhanced BERT. Intelligent Systems and Applications, Lecture Notes in Electrical Engineering*, 2023, vol. 959, pp. 263-271. Springer Nature Singapore. DOI: 10.1007/978-981-19-6581-4_21.

2. Ogunleye, B., Sharma, H., & Shobayo, O. Sentiment Informed Sentence BERT-Ensemble Algorithm for Depression Detection. *Big Data and Cognitive Computing*, 2024, vol. 8, no. 9, article no. 112. DOI: 10.3390/bdcc8090112.

3. Silva Barbon, R., & Akabane, A.T. Investigating Towards Transfer Learning Techniques—BERT, DistilBERT, BERTimbau, and DistilBERTimbau for Automatic Text Classification from Different Languages: A Case Study. *Sensors*, 2022, vol. 22, no. 21, article 8184. DOI: 10.3390/s22218184.

4. Fu, J. *A Comparison of CNN and Transformer in Continual Learning*. Master's Thesis, KTH Royal Institute of Technology, School of Electrical Engineering and Computer Science (EECS), 2023. TRITA-EECS-EX; 2023:793. URN: urn:nbn:se:kth:diva-340947f. PDF Available at: https://kth.diva-portal.org/smash/get/diva2%3A1820229/FULLTEXT01.pdf (accessed 12 April 2025).

5. Li, F., Li, J., & Abza, F. Sentiment Analysis of Tweets Employing Convolutional Neural Network Optimized by Enhanced Gorilla Troops Optimization Algorithm. *Scientific Reports*, 2025, vol. 15, article no. 795, pp. 1-16. DOI: 10.1038/s41598-025-85392-6.

6. Badi, H., Badi, I., El Moutaouakil, K., Khamjane, A., & Bahri, A. Sentiment Analysis and Prediction of Polarity Vaccines Based on Twitter Data Using. Deep NLP Techniques. *Radioelectronic and Computer Systems*, 2022, no. 4, pp. 19-29. DOI: 10.32620/reks.2022.4.02.

7. Singla, S., & Ramachandra, N. Comparative Analysis of Transformer-Based Pre-Trained NLP Models. *International Journal of Computer Sciences and Engineering*, 2020, vol. 8, no. 11, pp. 4044-4050. DOI: 10.26438/ijcse/v8i11.4044.

8. Ling, Y. Bio+Clinical BERT, BERT Base, and CNN Performance Comparison for Predicting Drug-Review Satisfaction. In: *Proceedings of the Workshop on Applied Data Science for Healthcare: Applications and New Frontiers of Generative Models for Healthcare, (KDD DSHealth 2023)*. Available at: https://arxiv.org/abs/2308.03782 (accessed 12 April 2025).

9. Kumar, G., Agrawal, R., Sharma, K., Gundalwar, P. R., Kazi, A., Agrawal, P., Tomar, M., & Salagrama, S. Combining BERT and CNN for Sentiment Analysis: A Case Study on COVID-19. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 2024, vol. 15, no. 10, pp. 676-685. DOI: 10.14569/IJACSA.2024.0151069.

10. Mareeswari, V., Patil, S. S., & Ramanan, G. *Real Time Sentiment Analysis of Tweets using Apache*

Spark and Scala. ACS Journal for Science and Engineering, 2021, Vol. 1, No. 2. DOI:10.34293/acsjse.v1i2.9.

11. Kumar, S., Nandakumar, K., Rajesh, R. A Feature Extraction Based Improved Sentiment Analysis for Real-Time Twitter Data Through Apache Spark. *International Journal of Computer Applications*, 2023, vol. 184, no. 51. ISSN: 0975-8887. DOI: 10.5120/ijca2023922633.

12. Kale, T. V.; Mendhe, S. A Review on Advances in Sentiment Analysis: A Deep Learning Approach Using Transformer Based Models. *Proceedings of the Fourth International Conference on Sentiment Analysis and Deep Learning (ICSADL-2025)*. IEEE. DOI: 10.1109/ICSADL65848.2025.10933230.

13. Sánchez-Moreno, P., & García-Muñoz, R. Sentiment analysis: A comprehensive review of recent advances and applications. *Frontiers in Physics*, 2024, vol. 12, article no. 1477714. DOI: 10.3389/fphy.2024.1477714.

14. García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J.M., & Herrera, F. Big Data Preprocessing: Methods and Prospects. *Big Data Analytics*, 2016, vol. 1, no. 1, article no. 9. DOI: 10.1186/s41044-016-0014-0.

15. Joshi, S., & Deshpande, D. Twitter Sentiment Analysis System. *International Journal of Computer Applications*, 2018, vol. 180, no. 47, pp. 35-39. DOI: 10.5120/ijca2018917319.

16. Jonnala, N. S., Alotaibi, R., & Reddy, B. R. Leveraging hybrid model for accurate sentiment analysis of Twitter data. *Scientific Reports*, 2025, vol. 15, article no. 1319, pp. 1–12. DOI: 10.1038/s41598-025-09794-2.

17. Martins, P., Cardoso, F., Váz, P., Silva, J., & Abbasi, M. Performance and Scalability of Data Cleaning and Preprocessing Tools: A Benchmark on Large Real-World Datasets. *Data*, 2025, vol. 10, no. 5, article no. 68, pp. 1-25. DOI: 10.3390/data10050068.

18. Kocaman, V., & Talby, D. Spark NLP: Natural Language Understanding at Scale. *Software Impacts*, 2021, vol. 8, article no. 100058. DOI: 10.1016/j.simpa.2021.100058.

19. Marco Pota, M., Ventura, M., Catelli, R., & Esposito, M. An Effective BERT-Based Pipeline for Twitter Sentiment Analysis: A Case Study in Italian. *Sensors*, 2021, vol. 21, no. 133. DOI: 10.3390/s21010133.

20. Ashbaugh, L., & Zhang, Y. *A* Comparative Study of Sentiment Analysis on Customer Reviews Using Machine Learning and Deep Learning. *Computers*, 2024, vol. 13, no. 12, article no. 340. DOI: 10.3390/computers1312340.

21. Gunasekara, S. P. *Enhancing the Detection of Adversarial Attacks Using Deep Learning Neural Transformer Models.* Doctoral Dissertation, The George Washington University, ProQuest Dissertations & Theses, 2025. ProQuest Document ID: 3142138338. ISBN: 9798346805526. Available at: https://www.proquest.com/dissertations/docview/3142138338 (accessed 10 October 2025).

Received 21.09.2025, Received in revised form 12.12.2025
Accepted date 15.01.2026, Published date 22.01.2026

АНАЛІЗ НАСТРОЇВ ПОПУЛЯРНИХ ТВІТІВ ІЗ ВИКОРИСТАННЯМ SPARK NLP ТА ГЛИБИННОГО НАВЧАННЯ: ПОРІВНЯЛЬНЕ ДОСЛІДЖЕННЯ CNN ТА ТРАНСФОРМЕРНИХ МОДЕЛЕЙ

Шайма Бенямані, Хасан Баді, Імад Баді, Азіз Хамжане

Предметом цього дослідження є впровадження різних моделей аналізу сентименту, дослідження їхніх теоретичних основ, надійних критеріїв оцінювання та суттєвих результатів із інтеграцією методів обробки природної мови для попередньої обробки даних, зібраних із соціальних мереж — основного джерела інформації. Мета роботи полягає у представленні помітного прогресу в класифікації сентименту шляхом застосування інноваційних моделей, таких як згорткові нейронні мережі (CNN) та архітектури на основі Transformer – BERT, DistilBERT і RoBERTa, із використанням Spark NLP для попередньої обробки**. Основні завдання** включають: збір набору даних із Twitter/X із фокусом на щоденних трендових темах; виконання кроків попередньої обробки за допомогою Spark NLP – провідної та масштабованої бібліотеки NLP, створеної на базі Apache Spark для ефективної обробки великомасштабних текстових даних; застосування лексиконної моделі VADER для визначення полярності кожного твіта; попереднє навчання моделей TextCNN та Transformer на основі лейблів VADER при однакових параметрах; тонке налаштування моделей на вручну анотованих твітах; порівняння їхньої ефективності в задачах класифікації сентименту, оцінка сильних та слабких сторін, а також загальної продуктивності, що може допомогти у виборі моделі в умовах обмежених ресурсів. **Використані методи** включають Spark NLP-пайплайни, слабку лейблізацію на основі лексикону та двофазне супервізоване навчання згорткових нейронних мереж (TextCNN) і моделей на основі Transformer (BERT, DistilBERT, RoBERTa) із застосуванням стратегій ранньої зупинки та поступового нарощування

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

швидкості навчання, а також порівняльних метрик оцінювання. Хоча набір даних (12 422 твіти, з яких 3 475 вручну анотовано) не є великим за обсягом, використання Apache Spark у цьому дослідженні забезпечує розподілену обробку даних і демонструє масштабованість для майбутніх, більших масивів. **Результати** показали, що моделі на основі Transformer перевершили TextCNN за точністю та стійкістю класифікації: RoBERTa досягла найвищої точності (85%), далі йдуть BERT і DistilBERT (84%), тоді як TextCNN досягла 72%. DistilBERT забезпечує баланс між якістю прогнозування та обчислювальною ефективністю. **Висновки. Наукова новизна** цього дослідження полягає в інтеграції попередньої обробки Apache Spark NLP, а також у бенчмаркінгу моделей CNN і Transformer, навчених на трендових твітах у гібридній двофазній стратегії, що поєднує слабкий лексиконний супервізор із ручною анотацією в ідентичних експериментальних умовах, аби надати методичні висновки та практичні рекомендації щодо вибору відповідних моделей для задач класифікації сентименту.

Ключові слова: аналіз тональності; Apache Spark NLP; соціальні медіа; TextCNN; двобічні енкодерні подання на основі трансформерів (BERT); DistilBERT; RoBERTa; двоетапне навчання; слабкий нагляд.

Чайма Беньямані – PhD Student, Національна школа прикладних наук Аль-Хосейма, Лабораторія прикладних наук Аль-Хосейма, Університет Абдельмалек Есааді, Тетуан, Марокко.

Хасан Баді – PhD, Лабораторія інженерних наук, Мультидисциплінарний факультет Таза, Університет Сіді Мохамеда Бен Абделла, Фес, Марокко.

Імад Баді – проф., Департамент математики та інформатики Університету Абдельмалека Есааді, Лабораторія прикладних наук Марокко.

Азіз Хамджане – проф., Департамент математики та інформатики Університету Абдельмалека Есааді, Лабораторія прикладних наук Марокко.

Chaimaa Benyamani – PhD Student, National School of Applied Sciences Al-Hoceima, Laboratory of Applied Sciences Al-Hoceima, Abdelmalek Esaadi University, Tétouan, Morocco,
e-mail: chaimaa.benyamani@etu.uae.ac.ma, ORCID: 0009-0000-3316-1749.

Hassan Badi – PhD, Laboratory of Engineering Sciences, Multidisciplinary faculty of Taza, Sidi Mohamed Ben Abdellah University, Fes, Morocco,
e-mail: hassan.badi@usmba.ac.ma, ORCID: 0000-0002-1568-9790.

Imad Badi – Prof., National School of Applied Sciences Al-Hoceima, Laboratory of Applied Sciences Al-Hoceima, Abdelmalek Esaadi University, Tétouan, Morocco,
e-mail: ibadi@uae.ac.ma, ORCID: 0000-0002-2844-4629.

Aziz Khamjane – Prof., National School of Applied Sciences Al-Hoceima, Laboratory of Applied Sciences Al-Hoceima, Abdelmalek Esaadi University, Tétouan, Morocco,
e-mail: akhamjane@uae.ac.ma, ORCID: 0000-0002-3508-8968.