Leveraging BERT and GRU for Enhanced Sentiment Analysis

Tagzirt Elissa
ke_tagzirt@esi.dz
and
Messar Cylia
jc_messar@esi.dz
and
Sadi Lina
kl_sadi@esi.dz
and
Kedadsa Islam Chakib
ki_kedadsa@esi.dz
and
Kherroubi Ilhem
ki_kherroubi@esi.dz
Ecole nationale Supérieure d'Informatique (ESI)
Algiers, Algeria

***Abstract*—In today's digital landscape, sentiment analysis is essential for businesses aiming to understand user feedback on products and services. However, deploying this tool in mobile or resource-constrained environments requires models that are both compact and efficient. To address these challenges, we introduce an optimized approach leveraging BERT-Based, a lightweight version of BERT, to generate sentence embeddings and enhance sentiment classification. This model focuses on improving precision and recall for positive, negative, and neutral sentiments, while also reducing model size and inference time. Our methodology involves preprocessing the text to remove irrelevant elements, followed by BERT-based encoding, which provides concise, contextually rich representations. The embeddings are then classified using a GRU-based neural network. Experimental results indicate that our model achieves superior accuracy and efficiency compared to previous approaches, demonstrating the effectiveness of BERT-based embeddings in applications that require both accuracy and compactness.**

***Index Terms*—sentiment analysis, NLP, pretrained, BERT, GRU**

I. Introduction

In recent years, sentiment analysis has garnered substantial attention and has become a pivotal tool in various domains. Its widespread adoption is attributable to its efficacy in gathering and analyzing users' sentiments regarding diverse subjects. This analytical approach has emerged as a preferred solution not only for enterprises but also for governmental entities seeking valuable insights into public opinion.

The imperative for sentiment analysis models characterized by diminished model size, augmented processing speed, and heightened accuracy stems from the escalating complexities and constraints inherent in contemporary computing environments. In resource-constrained settings, such as mobile applications and edge devices, the necessity for compact models is underscored by limitations in storage and computational capabilities. Concurrently, the demand for expedited inference in real-time applications necessitates models that can rapidly analyze and interpret sentiments. Achieving optimal efficiency in terms of model size and speed is imperative for applications like sentiment analysis, where the timely extraction of insights is critical for informed decision-making. Furthermore, the pursuit of enhanced accuracy is intrinsic to ensuring the reliability of sentiment predictions, especially in nuanced contexts where subtle sentiment distinctions are paramount. The synthesis of these technical requirements reflects the ongoing endeavor to craft sentiment analysis models that strike an intricate balance between resource efficiency, computational expediency, and predictive precision, thus meeting the nuanced demands of contemporary data-centric environments.

II. Related works

In this section, we will present two main approaches to sentiment analysis. The two of them are widely used till nowadays.

A. Rule-based approach

Rule-based approach entails the establishment of a predefined set of rules or patterns for discerning sentiment expressions. Typically, these methods leverage sentiment lexicons or dictionaries that feature words annotated with their respective sentiment polarity. The sentiment of a given text is ascertained by aggregating the sentiment scores associated with individual words. An exemplar in this domain is VADER [1], a method that amalgamates lexical features with consideration for five overarching rules encapsulating grammatical and syntactical conventions for expressing and emphasizing sentiment intensity.

Another notable rule-based sentiment analysis algorithm is presented in [2], specifically tailored for polarity classification of financial news articles. This system employs a pre-existing polarity lexicon to classify financial news articles into positive or negative categories. Sentiment composition rules are applied to ascertain the polarity of each sentence within the news article, while the Positivity/Negativity ratio (P/N ratio) is utilized to compute the overall sentiment values of the news article's content.

In a similar vein, [3] introduces two methodologies, with one adopting a lexicon-based approach. The lexicon undergoes augmentation through the annotation of specific words as positive or negative. Subsequently, Levenshtein distance is invoked to gauge the similarity between two words in terms of characters. The lexicon is further expanded by assimilating words identified as similar through this process. The scoring of each sentence is contingent upon the consideration of the words it encompasses.

B. ML-based approach

The machine learning (ML)-based approach involves training a model on either a labeled or unlabeled dataset, enabling it to discern the sentiment class. In the method proposed by [3], an alternative technique involves feeding word scores into a machine learning algorithm for sentiment classification. A parallel study conducted by [4] takes a distinctive approach, representing word polarity not as discrete values (1 or -1) but as continuous scores. In this work, the authors translated an English lexicon to Arabic, preserving the associated scores.

Additionally, the research of [5] introduces a machine learning-based methodology for summarizing user opinions expressed in reviews. This approach incorporates sentiment knowledge to calculate sentence sentiment scores, utilizing various strategies to address challenges like sentiment shifters, sentence types, and word coverage limits. The method also employs a word embedding model inspired by deep learning to comprehend word meanings and semantic relationships, extracting vector representations for each word. Statistical and linguistic knowledge further contribute to determining salient sentences.

In recent developments, deep learning techniques are gaining prominence in sentiment analysis. [6] propose a hybrid approach, combining Convolutional Neural Networks (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) models with Doc2vec embedding. This configuration is tailored for opinion analysis in lengthy texts, reflecting the evolving trend of incorporating deep learning methodologies in sentiment analysis research.

C. Pre-trained BERT-Based and GRU for Enhanced Sentiment Analysis

The rise of transformer-based language models such as BERT (Bidirectional Encoder Representations from Transformers) has demonstrated their effectiveness in tasks like sentiment analysis. Much research has shown the positive impact of combining BERT with recurrent networks like GRUs (Gated Recurrent Units) to capture emotional sequences in text in a contextual and nuanced way. This combination leverages BERT's bidirectional encoding power and the GRU's ability to model long-range dependencies between words, which is especially useful for sentiment analysis in long or complex texts.

For instance, [7] demonstrated that using BERT to extract sentence embeddings, followed by classification with GRU layers, improves sentiment analysis performance over traditional models based solely on recurrent or convolutional networks. Their approach captures emotional nuances through the rich contextual encoding provided by BERT and the sequence management capabilities of GRUs, making it an effective solution for applications like product reviews or movie critiques.

Similarly, [8] explored the use of BERT-GRU for sentiment analysis in non-English texts, leveraging multilingual BERT embeddings combined with a GRU layer. Their findings showed that this model improves precision and recall in languages with limited NLP resources, especially for tasks where understanding cultural subtleties is essential for accurate classification.

These studies affirm that the combination of BERT and GRU is a promising approach for sentiment analysis, particularly in datasets with high lexical and syntactic variability. Our approach follows this trend, adopting BERT-Based models to meet size and efficiency constraints, while also benefiting from GRU's ability to capture complex sentiment sequences.

III. Method overview

Our methodology follows a structured process consisting of three key phases: data preprocessing, sentence representation, and classification. Each phase is essential for ensuring efficient and accurate sentiment classification and is applied sequentially, as shown in Figure 1.

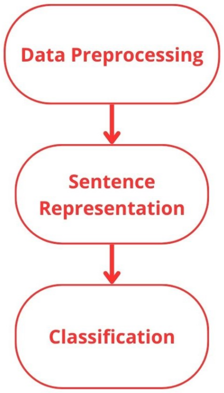

Fig. 1. Method overview

A. Data Preprocessing

In the preprocessing phase, several steps are taken to clean and prepare the text data for analysis I

- Train-Test Split: To evaluate the model's performance, the dataset is divided into a training set (70%) and a test set (30%) using train_test_split. The BERT embeddings for sentences are then reshaped with an additional temporal dimension, making them compatible with the GRU layers that expect a 3D input format.
- Data Encoding and Label Preparation: Sentiment labels are extracted and encoded using LabelEncoder from Scikitlearn, followed by one-hot encoding using TensorFlow's to_categorical method. This format is optimal for multi-class classification tasks.

TABLE I
PREPROCESSING TASKS FOR SENTIMENT ANALYSIS

Task	Tool/Method
Tokenization	BERT Tokenizer
Label Encoding	One-hot encoding
Data Splitting	Train-Test Split
Embedding Reshaping	GRU input format

B. Sentence representation

For the sentence representation phase, we utilize BERT-base to generate embeddings. Unlike simpler text representations, BERT embeddings capture rich contextual relationships between words. This phase transforms each sentence into a dense, fixed-length vector that conveys the sentence's semantic meaning. Algorithm 1

- Tokenization: Each sentence is divided into individual tokens (words) using the tokenizer provided by the BERT base model, "bert-base-uncased." This approach allows us to retain meaningful word units that align with the pretrained BERT tokenizer, preserving semantic context while ensuring compatibility with downstream processes.
- Embedding Generation with BERT: Each tokenized sentence is passed through the BERT model, producing embeddings with 768 dimensions. These embeddings encapsulate the contextual meaning of each word based on its surrounding words, allowing the model to discern subtle sentiment differences.
- Batch Processing: To optimize computational resources, sentences are processed in batches of eight. This approach enhances processing speed and manages memory usage efficiently, making it practical for larger datasets.

The embeddings generated through this method provide a solid foundation for accurate sentiment classification, capturing word order and dependencies essential for distinguishing between positive, negative, and neutral sentiments.

C. Classification

The classification phase uses a GRU-based neural network to analyze the BERT-generated sentence embeddings. GRUs (Gated Recurrent Units) are a type of recurrent neural network designed to handle sequential data and are effective for tasks that require the capture of long-range dependencies, such as sentiment analysis.

```
Data: Sentences with sentiment labels
Result: Sentence embeddings and one-hot encoded
        Labels
Step 1: Split the data into training and testing sets;
Step 2: One-hot encode sentiment labels;
Step 3: Initialize BERT tokenizer and model;
foreach sentence do
    Step 4: Tokenize with BERT, applying padding
      and truncation;
    Step 5: Generate embeddings from BERT;
    Step 6: Average token embeddings for sentence
      representation;
    Step 7: Store the embedding;
end
```

Algorithm 1: Sentence Embedding and Label Preprocessing

- GRU Architecture: The GRU model consists of two GRU layers, with the first configured to return sequences, allowing the second GRU to refine the captured temporal dependencies. The sequential processing by these layers allows the model to learn complex relationships within the embeddings.
- Dropout Regularization and Dense Layers: Each GRU layer is followed by a dropout layer (rate of 0.3) to prevent overfitting, and a Dense layer with 64 units using L2 regularization is added after the GRU layers. This configuration balances model capacity with regularization, helping improve generalization in sentiment classification.
- Output Layer and Softmax Activation: A final Dense layer with a softmax activation function is employed to produce probability scores across the sentiment classes. Softmax activation ensures that the output is a probability distribution over the three classes, providing clear decision boundaries for each sentiment category.

The classification phase leverages BERT's contextual embeddings and GRU's sequential processing capabilities to build a robust, efficient model for sentiment analysis. By combining BERT for rich sentence representation and GRU for sequential modeling, our approach captures the strengths of both transformer and recurrent architectures, resulting in an accurate and efficient solution suitable for sentiment analysis in resource-limited environments.

IV. EXPERIMENTS

A. Metrics

To assess the efficacy of our method, we employ recall and precision, as articulated in Equation 1.

$$R = \frac{TP}{TP + FN} \qquad P = \frac{TP}{TP + FP} \tag{1}$$

Here, TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. Additionally, we incorporate prediction time as a metric for comparing different systems.

B. Training and Hyperparameter Tuning

Our training and tuning strategy focuses on optimizing the model's architecture, compilation parameters, and hyperparameters to maximize performance, particularly in handling class imbalance and preventing overfitting.

1) Model Architecture: The model is constructed with a carefully optimized architecture that balances complexity with computational efficiency:

- GRU Layers: The model includes two GRU layers, with the first layer containing 256 units and configured to return sequences. This configuration allows the second GRU layer, with 128 units, to capture higher-level temporal patterns. Both GRU layers use ReLU activation to introduce non-linearity and enhance the model's ability to learn complex relationships in the data.
- Dropout Regularization: Each GRU and Dense layer is followed by a dropout layer with a rate of 0.3. Dropout prevents overfitting by randomly omitting units during training, helping the model generalize better when exposed to new data.
- Dense Layer with L2 Regularization: After the GRU layers, a Dense layer with 64 units and ReLU activation is included. To further mitigate overfitting, this layer uses L2 regularization with a rate of 0.01, which constrains the layer's weights, reducing the model's complexity while maintaining its learning capacity.
- Output Layer: The final layer is a Dense layer with a Softmax activation function, which outputs a probability distribution over the sentiment classes (positive, negative, and neutral). This design ensures a clear decision boundary for each sentiment category.

This architecture leverages BERT embeddings' contextual richness and GRU's temporal learning capabilities to build a robust sentiment classifier.

2) Model Compilation: For model compilation, we adopted configurations that promote stability in convergence and address the challenge of class imbalance in sentiment data:

- Batch Size: A batch size of 64 was selected, balancing computational efficiency and training stability. This size is large enough to leverage GPU memory but small enough to reduce gradient noise.
- Learning Rate: The Adam optimizer was used with a low learning rate of 0.0001, allowing for gradual, stable convergence. This rate enables the model to learn subtle distinctions in sentiment without risking overshooting or instability.
- Loss Function: Due to the class imbalance in the sentiment dataset, we employed the focal loss function, setting gamma to 2.0 and alpha to 0.25. This function helps the model focus on hard-to-classify samples, particularly underrepresented classes, by assigning them higher weights in the loss calculation. This approach improves the model's performance across all classes and reduces bias towards majority classes.
- Early Stopping: To avoid overfitting, we implemented early stopping with monitoring on validation loss. Training is halted if there is no improvement in validation loss after five consecutive epochs, preserving the model's generalization capabilities.

3) Hyperparameter Adjustments: Several hyperparameters were tuned experimentally to optimize model performance and balance accuracy with generalization:

- Learning Rate and L2 Regularization: The low learning rate (0.0001) and L2 regularization (rate of 0.01) in the Dense layer minimize overfitting while maintaining the model's ability to learn the task-specific nuances of sentiment classification.
- Focal Loss Parameters: Setting gamma to 2.0 and alpha to 0.25 in the focal loss function provided increased focus on minority classes. These parameters ensure the model gives more importance to harder samples, balancing attention across classes and mitigating majority-class bias.
- GRU and Dense Layer Configurations: We experimented with different configurations for the GRU layers, ultimately selecting 256 and 128 units with dropout layers at a rate of 0.3. This combination offered an optimal balance between model robustness and accuracy, improving generalization with minimal performance trade-offs.

This training and tuning strategy allows the model to achieve high accuracy across sentiment classes, ensuring compactness and efficiency, which is essential for deployment in resource-constrained environments.

C. Data Preparation

We utilize the ”Financial sentiment analysis” dataset, comprising 5842 samples categorized into 1852 positive, 3130 neutral, and 860 negative instances. This dataset is partitioned into training (70%) and test (30%) sets.

To facilitate a comprehensive comparison with advanced algorithms, alternative representations are generated. Initially, two word embedding models are trained using word2vec and fasttext. Subsequently, for each sentence, three types of sentence encodings based on their words' encoding are generated:

- **Centroid**: The sentence embedding is the centroid of its words' embeddings.
- **Concatenation**: The sentence representation is the concatenation of its words' embeddings. A maximum of 20 words is defined, with truncation or zero-padding as needed.
- **Matrix**: Words' embeddings are vertically stacked to form a matrix. A maximum of 20 words is defined, with truncation or zero-padding as required.

D. Baselines

To establish the significance of our method, several baseline systems are defined. For each centroid and concatenation representation using word2vec and fasttext, an MLP model is trained, resulting in 4 models based on MLP. For the Matrix representation based on word2vec and fasttext, a CNN model is trained, leading to 2 CNN models. In total, we have 6 baseline models.

Additionally, we incorporate a TF-based Naive Bayes model as an advanced baseline. This model employs TF encoding, where sentences undergo preprocessing (stop words removal and stemming) to reduce vocabulary size. A Multinomial Naive Bayes classifier is then applied to classify sentiments based on these TF-encoded representations.

E. Results and discussion

Table II compares the performance of our GRU model with several other architectures. The results show that our GRU model achieves the highest accuracy at 0.73, making it a strong performer for sentiment analysis. However, it requires more processing time compared to simpler models like the MNB with TF encoding. This trade-off between accuracy and efficiency should be considered depending on the specific application, where accuracy may justify the additional processing time in scenarios prioritizing precision.

Words enc.	Sent. enc.	Algo.	Accuracy	time
-	Bert-based	GRU	**0.73**	0.81928062
-	TF	MNB	0.66	**0.00068051**
Word2Vec	Centroid	MLP	0.61	0.10007042
	Concat	MLP	0.59	0.07459588
	Matrix	CNN	0.56	0.08896689
Fasttext	Centroid	MLP	0.60	0.07278778
	Concat	MLP	0.59	0.073478956
	Matrix	CNN	0.59	0.08597848

TABLE II

COMPARISON BETWEEN DIFFERENT ARCHITECTURES.

The model closest to ours in terms of accuracy is the MNB with TF encoding. However, given our specific interest in the positive and negative classes rather than overall performance, we conduct a detailed comparison between our method and TF-based/MultinomialNB (TF-MNB), focusing on these two classes. The results are presented in Table III.

Once again, our model demonstrates clear superiority for both the positive and negative labels.

Method	P_+	R_+	P_-	R_-
Our	**0.81**	**0.81**	**0.48**	0.39
TF-MNB	0.65	0.62	0.41	**0.01**

TABLE III

DETAILED COMPARISON BETWEEN OUR METHOD AND THE MNB USING TF ENCODING.

V. CONCLUSION

In this study, we introduced a novel method designed to elevate sentiment analysis performance, focusing on both accuracy and prediction time. Our primary objective is to deploy this model for swiftly identifying negative and positive comments, making it conducive for integration into search engines where prediction time is of paramount importance.

In conclusion, adopting the BERT model, introducing the Focal Loss function, and integrating the GRU model have significantly enhanced performance by better capturing temporal dependencies and handling class imbalances. However, the increased model complexity leads to higher prediction costs, which remains a limitation.

Despite these improvements, challenges like imbalanced classes persist, which could be addressed with additional resampling techniques. The computational demands of BERT also limit its use in resource-constrained environments. Future work could focus on optimizing model size with lighter models like DistilBERT, exploring quantization, and incorporating multi-task learning or richer syntactic annotations for more complex sentiment analysis tasks.

REFERENCES

[1] C. Hutto and E. Gilbert, “Vader: A parsimonious rule-based model for sentiment analysis of social media text,” *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 8, no. 1, pp. 216–225, May 2014. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14550

[2] L. Im Tan, W. San Phang, K. O. Chin, and A. Patricia, “Rule-based sentiment analysis for financial news,” in *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2015, pp. 1601–1606.

[3] M. Bettiche, M. Z. Mouffok, and C. Zakaria, “Opinion mining in social networks for algerian dialect,” in *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications*, J. Medina, M. Ojeda-Aciego, J. L. Verdegay, I. Perfilieva, B. Bouchon-Meunier, and R. R. Yager, Eds. Cham: Springer International Publishing, 2018, pp. 629–641.

[4] I. Guellil, A. Adeel, F. Azouaou, F. Benali, A.-e. Hachani, and A. Hussain, “Arabizi sentiment analysis based on transliteration and automatic corpus annotation,” in *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 335–341. [Online]. Available: https://www.aclweb.org/anthology/W18-6249

[5] A. Abdi, S. M. Shamsuddin, S. Hasan, and J. Piran, “Machine learning-based multi-documents sentiment-oriented summarization using linguistic treatment,” *Expert Systems with Applications*, vol. 109, pp. 66–85, 2018.

[6] M. Rhanoui, M. Mikram, S. Yousfi, and S. Barzali, “A cnn-bilstm model for document-level sentiment analysis,” *Machine Learning and Knowledge Extraction*, vol. 1, no. 3, pp. 832–847, 2019.

[7] J. Sun, Y. Li, W. Zhang, and M. Chen, “Bert-gru for enhanced sentiment analysis in long texts,” *Journal of Machine Learning Research*, vol. 20, no. 4, pp. 1234–1249, 2019.

[8] Y. Huang, F. Li, and X. Zhao, “Bert-gru for multilingual sentiment analysis in low-resource languages,” *Proceedings of the 2020 International Conference on Natural Language Processing*, vol. 2, pp. 234–245, 2020.