International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 15(2023) pp. 307-320

Submitted: 13 Jan, 2013; Accepted: 26 May, 2018; Publish: 13 June, 2023

A new strategy for incorporating BERT embeddings to enhance static word embeddings: The case of COVID-19 SA

Nouhaila Bensalah[1], Habib Ayad[1], Abdellah Adib[1] and Abdelhamid Ibn El Farouk[2]

[1]Team: Data Science & Artificial Intelligence Hassan II University of Casablanca Casablanca 20000, Morocco
nouhaila.bensalah@etu.fstm.ac.ma, ayad.habib@gmail.com, abdellah.adib@fstm.ac.ma

[2]Teaching, Languages and Cultures Laboratory Mohammedia, Hassan II University of Casablanca, Morocco
farouklettres@gmail.com

Abstract: **COVID-19 has claimed many lives to date, not only due to the virus' physical infection but also due to mental illness, which is related to people's emotions and psychology. People have been panicked, nervous, and sad as the number of positive cases has increased quickly worldwide. This deadly epidemic has been shown to have a direct influence on the population's physical and mental health. Throughout this period, social media platforms have played a crucial role in the global spread of news about the outbreak, as individuals shared their emotions over them. Based on this overwhelming evidence, we aim to build a powerful system to analyze people's feedback on Twitter, targeting specific keywords associated with the outbreak, either directly or indirectly. Therefore, we assume that the effectiveness of contextual word vectors generated with Language Models (LMs) can be further enhanced with the inclusion of static word embeddings that are specially trained on social media (tweets about COVID-19). Moreover, we evaluate different approaches to combine static word embeddings in order to take advantage of their complementarity. Furthermore, we proposed a new technique for dealing with the imbalanced dataset problem. As compared to previous studies, the experimental findings proved that our proposed technique improves the efficiency of the COVID-19 Sentiment Analysis system. Furthermore, a fair comparison of both contextual and static embeddings through Sentiment Analysis reveals that our technique beats the static embeddings trained only from scratch or the ones generated from LMs.**

Keywords: Sentiment Classification, Contextual Embeddings, Traditional Embeddings, COVID-19, Twitter, BERT

I. Introduction

The Covid-19 disease was first announced on December 31, 2019, in Wuhan, Hu-bei province, China, and it quickly spread worldwide. Dr. Tedros Adhanom Ghebreyesus, Director-General of the World Health Organization (WHO), subsequently declared the outbreak a pandemic on March 11, 2020. [36]. The term "pandemic" refers to a disease that spreads rapidly and encompasses a geographic region, such as a country or the entire world [34]. Pandemics have marked human history for decades, spreading fear and killing millions of people, whether the outbreak was a plague, smallpox, or influenza [18]. Likewise, COVID-19 has shifted the world's attention back to catastrophic epidemics and challenged our ability to address the threat of highly infectious viruses, including coronaviruses, which are a well-recognized health threat [15]. Starting from China, the COVID-19 virus has infected and taken the lives of many people from Italy, Spain, the United States, the United Kingdom, Brazil, Russia, and numerous other countries. Despite this, more than 80% of infected people suffer mild to moderate disease and survive without being hospitalized [3, 40]. The main symptoms include fever, dry cough, and exhaustion, which are mild in the majority of patients. Although severe symptoms may occur such as chest discomfort or pressure, loss of voice or movement, and breathlessness for a minority of people [13, 39]. Those who get critically ill are more likely to be men and older, the risk increasing progressively with each decade over the age of 50 [3]. As the epidemic destroyed the lives of millions of people, many countries imposed harsh lockdown measures for different periods to break the pandemic chain [36]. During the lockdown phase, many users preferred social media to communicate their feelings about the virus. Thus, we were motivated to assess human feelings about the outbreak by analyzing this massive social media data [12]. Social media is a key element in people's daily life as it links them to the rest of the globe. It is impossible to work without having access to social media to keep up with all the latest news, such as coronavirus outbreaks and stock market changes [11, 35]. Nowadays, people rely heavily on posts and tweets published on social media sites like Twitter, Facebook, and Instagram. Hence, examining people's attitudes according to the classification of tweets collected through the social media platform has a

significant influence on the global audience. In this study, we will examine what people think about COVID-19 on Twitter, as it is a popular social media platform and microblogging medium wherein people publish and exchange messages named "tweets". Approximately 600 million tweets per day and 200 billion tweets per year have been posted on Twitter as it has emerged as a leading data outlet for online media conversations identified with both public and global situations.

Previous approaches to COVID-19 tweet analysis highlight the relevance of traditional Machine Learning (ML) and Deep Learning (DL) based algorithms for Sentiment Analysis (SA) related tasks [1, 4, 5, 16]. In most of these works, they choose DL-based techniques for SA. As DL continues to gain more attention, Neural Network architectures like Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) have proven a decent performance improvement in tackling various Natural Language Processing (NLP) tasks including Text Classification, Language Modeling, Machine Translation, etc [6, 7, 22, 33]. One of the drawbacks of the DL models is that they require massive computational resources. These challenges have prompted researchers to question the feasibility of knowledge transfer through large trained Language Models (LMs). Hence, the need for Transfer Learning is increasing as a result of the development of numerous large-scale models. The core idea of Transfer Learning is to transfer parameters or information from one language model to another. LMs have changed recently, and they have yielded far significant improvements when compared to traditional ones. Such trained LMs have been deployed for transferring information to tackle several NLP problems, and their performance was promising over ML and DL approaches. Meanwhile, static embedding models have substantial benefits over LMs. They can be trained on social media datasets with low computational resources [38]. Since the vocabulary in social media posts, especially about the COVID-19 virus is mostly written in regional dialects, is different from the vocabulary that LMs have seen during the training phase. Hence, these two models (static and contextual embedding models) are complementary, and their combination can improve the performance of SA on social media reviews. Therefore, we tried a Language Model based on Bidirectional Transformer Encoded Representations (BERT) along with static embedding models trained on COVID-19 tweets.

In this paper, we proposed a new technique for dealing with the imbalanced dataset problem. Next, we suggest that the effectiveness of contextual word vectors trained with Language Models for social media can be further enhanced by incorporating static word embeddings that are specially trained on social media (tweets about Coronavirus or COVID-19). We evaluate different approaches to combine static word embeddings in order to take advantage of their complementarity. The combined static embeddings and the contextual ones are fed into four DL architectures in order to select the best scheme for combining BERT-based vectors with static ones. This paper includes the following contributions

- Embeddings at both word and character levels were generated.
- Concatenation, Principal Component Analysis (PCA), and ordinary autoencoder are used to combine various static word embeddings in order to find effective embeddings.
- Four deep learning architectures were evaluated to determine the optimal way to combine BERT-based vectors with static ones.
- A new technique for combining character- and word-level embeddings is proposed.
- In order to address the unbalanced Covid-19 dataset, a new data augmentation technique was used.

The rest of this paper is structured as follows: Section II investigates classification-related works using tweets about COVID-19. Similarly, Section III describes the materials, suggested approach, and implementation. Section IV presents the experimental settings. Furthermore, Section V examines the findings and compares them to current state-of-the-art approaches, and Section VI concludes the paper with a list of upcoming works.

II. Related works

SA or opinion mining refers to the process of gathering people's ideas, feelings, attitudes, and emotions regarding a topic or situation based on massive amounts of unstructured data. Recently, a tremendous amount of research has been conducted to create methods to evaluate and explain the process of SA in many languages. Nemes et al. [29] proposed an RNN-based SA model. They created a COVID-19-based Twitter dataset and sorted it into four fine-grained classes: weak positive, weak negative, strongly positive, and strongly negative. Their method outperformed TextBlob's [24]. Balahur [2] analyzes Twitter datasets by means of supervised Machine Learning (ML) methods, including Support Vector Machine (SVM), using a basic, linear kernel (to minimize the over-fitting of the data) and the training set's unigrams and bigrams as features. Their obtained findings using those approaches on the Twitter data showed clearly that the unigram and bigram-based techniques are more efficient than the SVM. The results incorporate unique labels, modifiers, and emotional keywords which are used to improve the quality rating of emotions. Ortis et al. proposed a multimodal embedding space approach for analyzing and extracting text from several sets of images [30]. The model assessed the emotions throughout the images using an SVM on the text characteristics. Using multiple-output support vector regression and the multiple-input multiple-output method, a novel multi-stage based prediction model was proposed by Han et al. [17]. Their research also included a comparison of three key prediction models. The model was tested using both simulated and real-world data. Both quantitative and thorough evaluations were performed in terms of expected accuracy and computational costs. In [20], the authors introduce a sentiment classification model called "LeBERT," which utilizes a combination of sentiment lexicon, N-grams, BERT, and CNN. To evaluate the effectiveness of the model, three publicly available datasets are used: Amazon product reviews, IMDb movie reviews, and Yelp restaurant reviews. Accuracy, precision, and F-measure

are employed as performance metrics. The experimental findings demonstrate that LeBERT model surpasses the current state-of-the-art models, achieving an F-measure score of 88.73% for binary sentiment classification. Jun Wu et al. [41] introduce an optimized BERT model called "hierarchical multi-head self-attention and gate channel BERT." The model consists of three modules: the hierarchical multi-head self-attention module, which extracts features hierarchically; the gate channel module, which filters information instead of using BERT's original Feed Forward layer; and the tensor fusion model, which utilizes a self-attention mechanism to fuse different modal features. Experimental results indicate that their proposed method achieves promising performance on the CMU-MOSI dataset, improving accuracy by 5-6% compared to traditional models. Naseem et al. [28] examined classic ML methods like Random Forest (RF), SVM, Decision Tree (DT), Naive Bayes (NB) and DL approaches such as CNN and Bidirectional Long Short-Term Memory (BiLSTM) using different embeddings generated using Glove [10], Word2Vec [32], and FastText [25]. The performance of the classifiers was evaluated using their own COVID-19-related tweets dataset, which was categorized into three (positive, neutral, and negative). The results revealed that DL-based techniques outperformed traditional ML algorithms. They also use transformer-based learning methods, such as BERT [14], XLNET [43] and ALBERT [23], which achieved the maximum accuracy of 92.90%.
The purpose of this paper is to introduce an improved word embedding model that combines contextual and static embeddings. Our proposed model is built upon contextual models, which have demonstrated their efficacy in several NLP applications, and static embedding models. Nevertheless, contextual models have excessively high computing costs in many use cases and are sometimes difficult to interpret. As a result, employing contextual models like BERT to generate word embeddings associated with recent topics discussed on social media (the COVID-19 pandemic) is challenging. Static embedding models, on the other hand, can be efficiently trained on social media with fewer computational resources. As a consequence, they outperform both existing static embedding approaches and contextual embedding methods alone.

III. Methodology

A. Satic Word embedding models

This section describes the static word embedding models that will be used:

1) Word2vec [25]

Mikolov et al. presented Word2vec, a prediction-based approach for generating dense embeddings of unique words from a huge unlabeled dataset. It is built on a simple neural network that learns the mapping of words to vectors. Word2vec includes two models, Continuous Bag-Of-Words (CBOW) and Skip-Gram (SG), along with advanced optimization approaches such as hierarchical softmax and negative sampling. CBOW focuses on predicting a center word given a window of contextual words, whereas SG predicts the contextual words based on a center word. The two main parameters for predicting CBOW or SG embeddings are the dimension of the vectors representing the size of the word embeddings, and the length of the context window, which represents the set of words that must be utilized as a context either before or after the core word to generate the word vectors.

2) GloVe [32]

The Global Vectors for Word Representation, or GloVe, is a variation on the word2vec strategy for effectively generating word vectors. Traditional vector space models to represent words have been generated through matrix factorization-based techniques, e.g. Latent Semantic Analysis (LSA), which both achieve better results when utilizing global text statistics, but are not in the same class as other techniques like word2vec at catching importance and displaying it on NLP tasks. Briefly, GloVe is a method for integrating global measures of matrix factorization techniques such as LSA alongside regional context-based learning in word2vec. Rather than using a window to describe neighboring settings, GloVe uses statistics from the whole text corpus to construct an explicit term context or term co-occurrence matrix. The final output is a model of learning which might lead to enhanced word embeddings in general.

3) FastText [10]

CBOW and SG embedding methods provide compact word vectors to words, which are atomic units. They are not able, however, to manage the Out Of Vocabulary (OOV) issue because they disregard inner sub-word knowledge (i.e., sequences of neighboring letters), which is important in languages with a wide and diverse vocabulary. Bojanowski et al. [10] developed FastText, which is basically an optimization on word2vec. Each word in the dictionary is handled as a bag of character n-grams, and the embedding of the characters generated are merged to form the token's vector. As a consequence, the FastText model can handle OOV words and represent words' morphology and lexical similarity.

B. Combining static word embeddings

In this paper, we are interested in the combination of static word embeddings through concatenation, Principal Component Analysis (PCA) and ordinary autoencoder in order to obtain effective embeddings that can achieve good performance. The combination approaches are briefly detailed as follows:

- **Concat**: We take a straightforward approach of concatenating the three-word embedding types from each combination set. This results in a 900-dimensional vector representation for each word.
- **PCA**: The PCA technique is applied to concat embeddings. First, the matrix comprising all words is mean-centered using Z-scores based on these embeddings. Next, we compute PCA using the correlation method to obtain a new coordinate system. Finally, we project the data onto the new basis by only considering the first 300 components.

- **AutoE** We investigate the use of ordinary autoencoder [37]. The employed auto-encoder consists of a single hidden layer containing 300 hidden units. It accepts the Concat embeddings as input and produces a 900-node vector as output. To generate the combined word embedding for each word, we use the numerical values vector generated by the hidden layer.

C. Contextual Word embedding models

In this paper, we use BERT as contextual word embedding model

1) BERT [14]

BERT represents a contextual language model proposed by Devlin et al. [14] in 2019, which is built on a multilayer bidirectional transformer encoder that takes into account respectively left and right contexts. BERT was developed based on two unsupervised applications: the masked language model and sentence prediction. In the masked language model, a portion (15%) of the tokens in the input sequence is randomly masked, and the model predicts the masked token in a multilayered context. The hidden vectors generated for the masked tokens are then passed through an output Softmax over the vocabulary. In sentence prediction, the goal is to comprehend the relationship between two phrases. The first token in each input sequence is a special classification token, and the sequence representation is the final hidden state corresponding to this token. The final embedding of BERT is the result of combining token embedding, segment embedding, and position embedding. Token Embedding pertains to the embedding of the current word, Segment Embedding refers to the index embedding of the phrase in which the current word is positioned, and Position Embedding is the index embedding of the current word position. Additionally, we employ multilingual BERT, which is a pre-trained model that uses a Masked Language Modeling (MLM) target with 102 languages from Wikipedia.

D. Classification

In this work, we explore the potential applications of BERT as a linguistic model for representing COVID-19 reviews. The primary benefit of BERT over static embedding models is its capability to build contextualized word embeddings. By employing a static embedding model, the token "sentence" will have an identical representation across different sentences, such as "The judge gave a sentence" and "A sentence is a linguistic structure." On the other hand, when adopting BERT, the token "sentence" will have a distinct representation for those phrases. Indeed, BERT creates contextualized word vectors that consider the context into account to provide effective embedding. Fine-tuning BERT for a specific task, on the other hand, is inefficient in terms of parameters. Unlike BERT, which is a dynamic embedding model trained on a large corpus of data, static embedding models can handle domain-specific datasets with unique features. Additionally, static embedding models have demonstrated adequate performance while requiring less training time. By combining static and contextualized embeddings, our approach can optimize sentiment analysis efficiency for COVID-19 comments.

Given a text sentence $S = \{w_1, w_2, ..., w_t, ..., w_n\}$, where w_i refers to the i-th word in the sentence, we aim to combine BERT-derived contextual embeddings with static word vectors using an attention mechanism to obtain a sentence representation. In our first approach, we adopt a CNN (CNN-1) architecture to obtain the sentence vector from the contextual embeddings. The CNN-1 architecture employs 64 filters, each with a window size of 2, 3, or 5 words. To avoid the problems associated with gradient vanishing [19], we used a linear Rectification Unit (ReLU) [26] as the activation function. As illustrated in Figure. 1, to capture the most salient features, we use a max-pooling layer after obtaining the sentence vector from the CNN-1 architecture. The extracted features are then merged, and a global max-pooling layer is applied to the combined features to obtain the final sentence representation. In addition, we adopt the Bidirectional Gated Recurrent Unit (BiGRU) to capture the local features of the combined static embeddings. The BiGRU layer is employed with a hidden state size of 200 for the backward and forward layers. This layer outputs the entire sequence of features, including information from each time step. For the convolutional and BiGRU layers, we used respectively a ReLU and a hyperbolic tangent (tanh) as activation functions. Next, the attention technique is added on the top of the output generated by BiGRU., i.e., $h_1, h_2, .., h_t, ..., h_n$. For each time step t, we first feed h_t into a fully-connected network to obtain u_t which represents the hidden representation of h_t, and then estimate the importance of the token w_t as the similarity of u_t with the context vector generated via CNN-1 p, a softmax function is then used to generate a normalized significance weight α_t. Next, we compute the text vector d as a weighed arithmetic mean of h using the weights $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_n\}$.

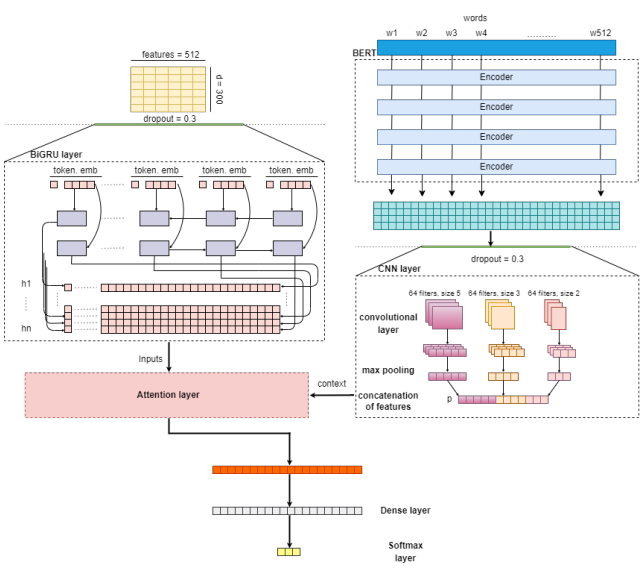
$$u_t = tanh(\mathbf{W}_w h_t + b_w) \tag{1}$$

$$\alpha_t = \frac{exp(u_t^T p)}{\sum_t exp(u_t^T p)} \tag{2}$$

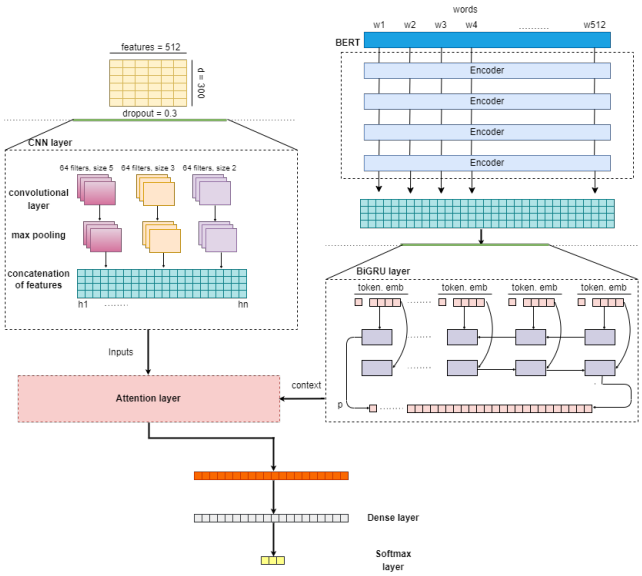
$$d = \sum_t \alpha_t h_t \tag{3}$$

where $\mathbf{W}_w$ and b_w are the weight matrix and bias, respectively.

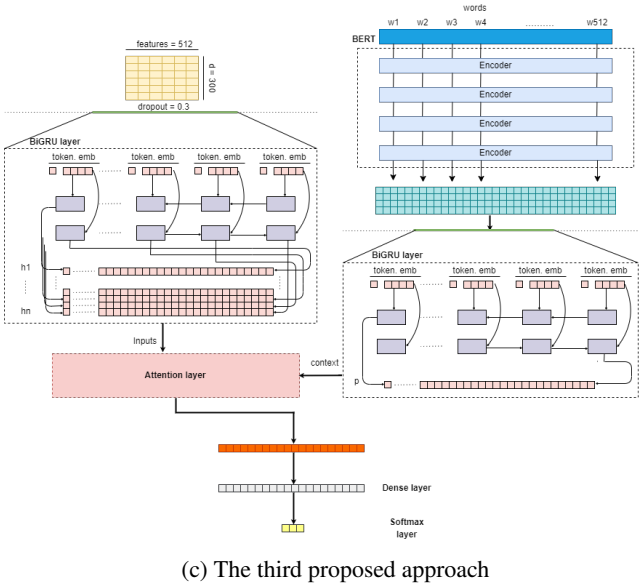
The proposed unified model can select the meaningful features of the static embedding model along with the context generated by the BERT model. This is due to the fact that we merge the sentence vector of the CNN-1 model applied to the context vectors and the outputs of BiGRU applied to the static word vectors through the attention mechanism. After merging the two types of vectors via the attention mechanism, we apply a Dense layer, followed by a softmax classification layer, see Figure. 1(a). In our second approach, the BiGRU layer is used to generate the sentence vector from the contextual embeddings. Furthermore, to extract regional characteristics from the combined static embeddings, we employ a CNN architecture (CNN-2). We used 64 filters with window sizes of 2, 3, and 5-word embeddings with a max-pooling layer for the convolutional layer. A BiGRU layer with a hidden state size of 200 for the forward and backward layers is employed. The BiGRU layer simply returns



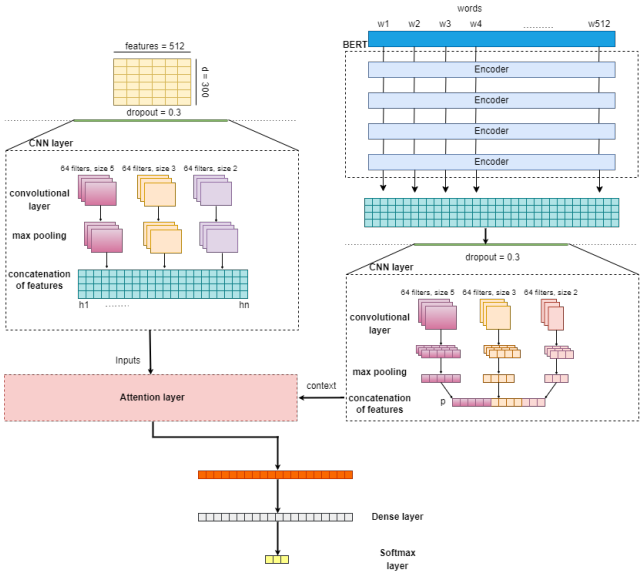
(a) The first proposed approach



(b) The second proposed approach



(c) The third proposed approach



(d) The fourth proposed approach

Figure. 1: The proposed approaches

the sequence acquired in the previous time step. The attention mechanism is then used to combine the sentence vector from the BiGRU model applied on the contextual vectors and the outputs of CNN-2 applied on the static word vectors, see Figure. 1(b). For our third approach, we employed an architecture that extracts local characteristics from contextual embeddings provided by the BERT model and the combined non-contextual embeddings extracted using static embedding models (Word2vec, FastText, and GloVe). We utilized the BiGRU architecture, but we only return the sequence acquired in the last time step. The latter is applied on top of the contextual embeddings. The BiGRU layer is utilized for the combined static embeddings, with a hidden state size of 200 for the forward and backward layers. This layer outputs the whole sequence of features, including the information from each time step, rather than just the sequence generated in the last time step. Subsequently, the attention mechanism is used to merge the sentence vector of the BiGRU model applied to the contextual word vectors with the outputs of the BiGRU model applied to the static word vectors, see Figure. 1(c). For our last approach, we employed the same CNN model (CNN-2) adopted in the second approach to extract the regional characteristics from the combined static embeddings. Furthermore, the sentence vector was extracted from the contextual embeddings using the CNN model (CNN-1) deployed in the first method. The attention mechanism is then used to combine the sentence vector from the CNN-1 model applied to the contextual vectors with the outputs of the CNN-2 model applied to the static word vectors. After concatenating the two types of sentence vectors with the attention mechanism, we apply a Dense layer, followed by a softmax classification layer, see Figure. 1(d).

IV. Experimental setting

A. Dataset

We use the dataset proposed in [28], which had 90,000 distinct tweets annotated with positive, neutral, and negative labels. The COVIDSENTI dataset was partitioned into three equal-sized subsets: COVIDSENTI-A, COVIDSENTI-B, and COVIDSENTI-C. COVIDSENTI-A contains a substantial proportion of tweets about the government's action concerning COVID-19. COVIDSENTI-B is composed of tweets involving COVID-19 problems, social distancing, confinement, and working from home. As a result, it largely addresses the temporal shift in a person's activity as a function of the number of victims, anxiety information, and so on. COVIDSENTI-C is built using tweets on COVID-19 cases, the epidemic, and resting at home. Thus, it mainly presents people's behavior patterns as a result of an increase in the number of cases. Figure. 2 illustrates the distribution of these tweets based on their class (degree of satisfaction).

B. Preprocessing

To represent covid-19 reviews utilizing the mentioned word embedding models, the reviews need to be preprocessed. Preparing the data is a crucial step in ML and DL, as it involves removing irrelevant information and optimizing the data to improve the learning process of classification models and enhance their accuracy. Superfluous information can be defined as any data that contributes either very little or no contribution at all to the target class prediction; yet, it increases the feature vector size, introducing additional computational complexity. Therefore, if no or insufficient preprocessing is performed, the performance of classification models decreases. Therefore, data cleaning or preparation activities are performed before encoding [27]. For this purpose, a series of techniques are applied in order to enhance the text.

1. Hashtags are used to represent subjects on almost all social networking sites, i.e., #StayHome, #StaySafe, #COVID-19 and #Coronavirus. In most scenarios, hashtags are unnecessary and can have an impact on the effectiveness of the classification algorithm. As a result, we began with basic text cleaning, deleting links, @mentions, and punctuation. Special letters, punctuation, and numerals are removed from the dataset since they are ineffective for sentiment recognition.

2. The text is then lowercase in the second step. We fold all capital letters into lowercase to avoid confusing one word with another because of the capitalization.

3. Several words are spliced with other terms, including hashtag keywords like "stayathomestaysafe" and "coronavirus," which must be "stay at home stay safe" instead. As a result, we accomplish this process using word segmentation.

C. Exploratory Data Analysis

We perform an exploratory study to gain a better understanding of our dataset. Our goal in this section is to extract hidden and unknown knowledge from data in such a way that we may acquire an instantaneous, direct, and simple representation of it. When using visual graphs, the human brain receives a more immediate and accurate assessment of similarities, trends, and correlations through a picture rather than a sequence of numbers.

1. **Keyword Trend Analysis:** To begin, we conducted a search term trend analysis using the preprocessed corpora to identify the most frequently used phrases. Users have complained about coronavirus infections, coronavirus epidemics, separateness, coronavirus pandemics, coronavirus crises, and remaining at home, according to our findings. Statistics on the top ten most regularly used keywords are collected and reported in the table. Figure. 3 depicts a word cloud of the most prevalent terms in the positive, negative, and neutral tweet classifications.

2. **Topic Modeling:** We use Latent Dirichlet Allocation (LDA) to examine the distributions of topics in our dataset to quantitatively analyze them [9]. LDAs are mixture models, which means that documents can belong to many topics and the membership is fractional [20]. In addition, each topic is a combination of words in which words can be shared across multiple topics. This allows for a type of "fuzzy" unsupervised clustering in which a single document can belong to many topics, each having an associated probability. LDA is a bag-of-words model in which each vector represents a number of terms. In LDA, a number of topics should be provided. We have limited the number of topics to six in this paper. The topics representing a word distribution and the document topic distributions are learned after the training of LDA. Figure. 4 shows the word count and the importance of keywords within the top six topics.

D. Data augmentation

The three-class dataset exhibits an imbalanced distribution of classes, as previously mentioned. Therefore, we applied a data augmentation technique to balance the unbalanced dataset. In the NLP field, it is hard to augment text due to the high complexity of language. Not every word we can replace with others such as a, an, the. Also, not every word has a synonym. Even changing a word, the context will be totally different. On the other hand, generating an augmented image in the computer vision area is relatively easier. Even by introducing noise or cropping out a portion of the image, the model can still classify the image. In this paper, we used a novel open-source data augmentation library known as AugLy [31]. AugLy is an open-source data augmentation library that offers over 100 augmentations across four modalities: audio, image, text, and video. The augmentations provided in AugLy are inspired by the real-world perturbations that people make to data online every day. For example, AugLy provides augmentations such as overlaying text, emojis, and screenshot transforms for images and videos, and inserting punctuation or similar characters for text. Unlike other text augmentation libraries that mainly focus on word-level augmentations, AugLy offers a range of syntactic augmentations, including character-level augmentations, that

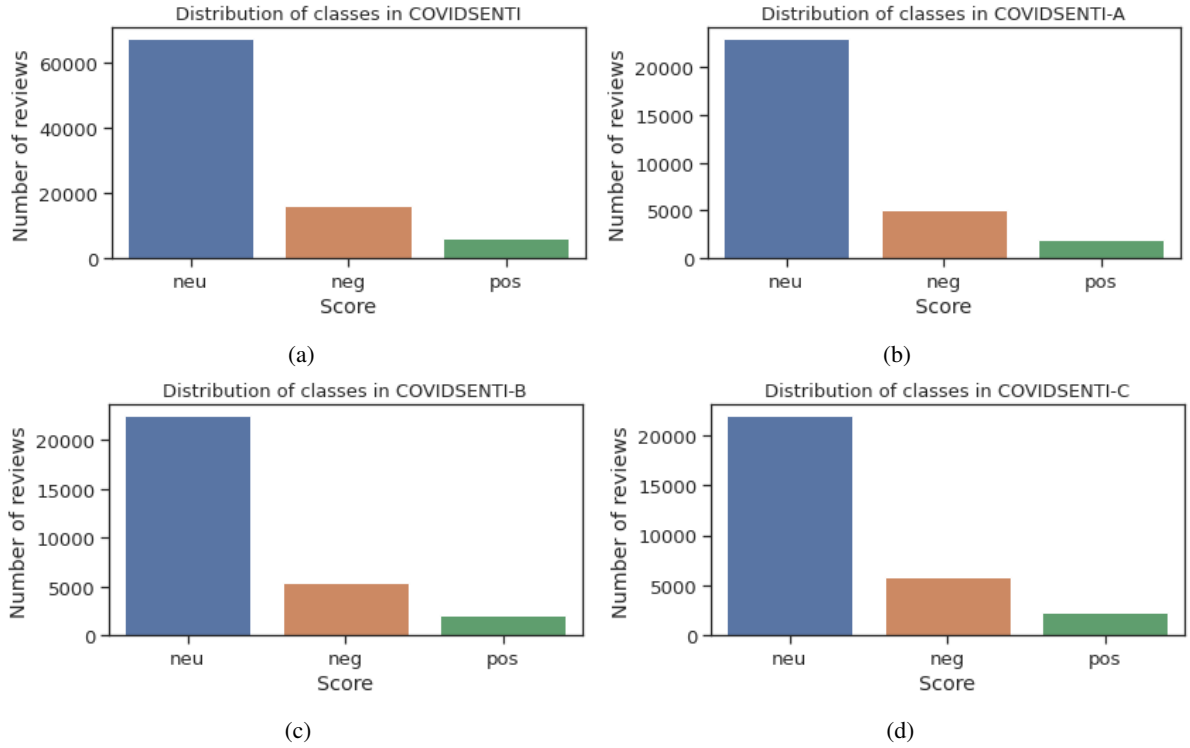


Figure. 2: Class distribution in COVIDSENTI (a), COVIDSENTI-A (b), COVIDSENTI-B (c), COVIDSENTI-C (d)

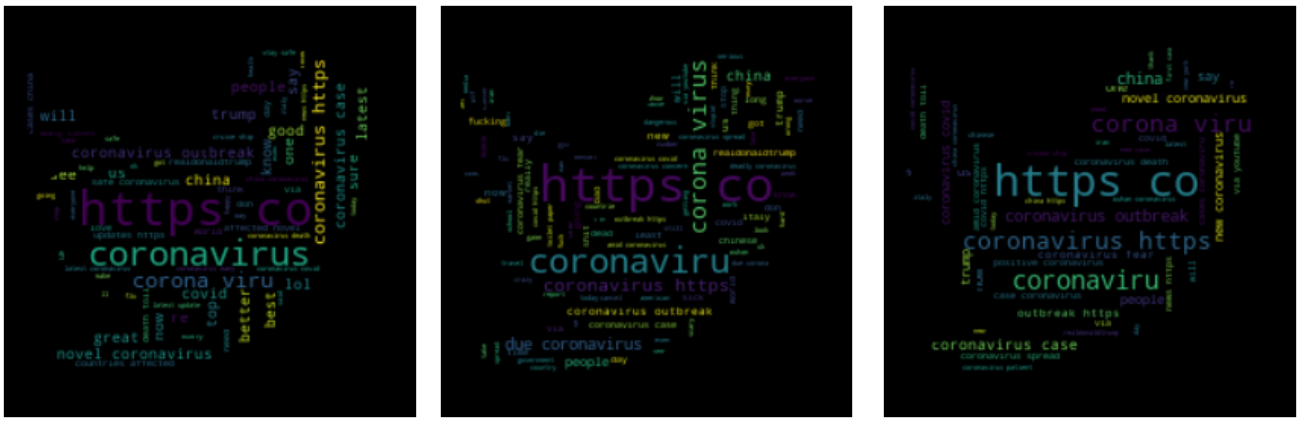


Figure. 3: Word cloud of (a) positive, (b) negative, and (c) neutral tweets

are commonly used online to avoid detection, such as inserting punctuation, zero-width or bidirectional characters, and changing fonts. The ratio of sentiments before and after applying AugLy is shown in Figure. 5. In this study, we adopt an 80:20 split ratio, which means that 80% of the data is used for model training and 20% is used for model testing. To ensure model generalizability and reduce variance, the data is shuffled before being split. Shuffling also helps to make the training data more representative of the overall distribution of the data and minimize the risk of model overfitting. The Tables 1, 2, 3 and 4 show the number of tweets in the training, validation and test sets of COVID-SENTI, COVID-SENTI-A, COVID-SENTI-B and COVID-SENTI-C, respectively, with and without the AugLy approach.

Since the length of COVID-19 reviews varies, padding and truncation of texts were necessary to equalize their lengths.

Table 1: Train and test count after data splitting for COVID-SENTI

Technique	Dataset	Positive	Negative	Neutral	Total
Original	Total data	6280	16335	67385	90000
	Testing set	628	1633	6739	9000
	Training set	5087	13231	54582	72900
	Validation set	565	1471	6064	8100
AugLy	Total data	66280	66335	67385	200000
	Testing set	6628	6633	6739	20000
	Training set	53686	53732	54582	162000
	Validation set	5966	5970	6064	18000

Mostly all processed sentences have a length that is less than or close to 512 tokens, based on the cumulative distribution function across the length of reviews. As a result, the length of the processed sentences was restricted to 512 tokens. For

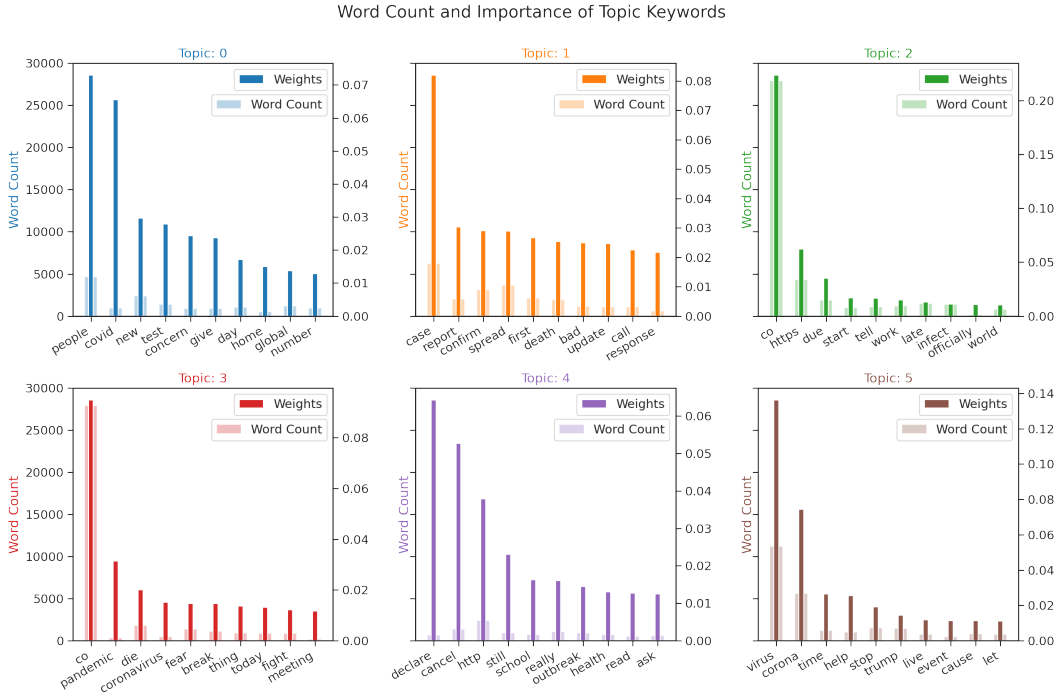


Figure. 4: Word count and the importance of keywords within the top six topics

Table 2: Train and test count after data splitting for COVIDSENTI-A

Technique	Dataset	Positive	Negative	Neutral	Total
Original	Total data	1968	5083	22949	30000
	Testing set	197	509	2294	3000
	Training set	1594	4117	18589	24300
	Validation set	177	457	2066	2700
AugLy	Total data	20968	20083	22949	64000
	Testing set	2097	2009	2294	6400
	Training set	16984	16267	18589	51840
	Validation set	1887	1807	2066	5760

Table 3: Train and test count after data splitting for COVIDSENTI-B

Technique	Dataset	Positive	Negative	Neutral	Total
Original	Total data	2033	5471	22496	30000
	Testing set	204	547	2249	3000
	Training set	1647	4431	18222	24302
	Validation set	183	493	2024	2700
AugLy	Total data	20033	20471	22496	63000
	Testing set	2004	2047	2249	6300
	Training set	16226	16582	18222	51030
	Validation set	1803	1844	2024	5670

Table 4: Train and test count after data splitting for COVIDSENTI-C

Technique	Dataset	Positive	Negative	Neutral	Total
Original	Total data	2279	5781	21940	30000
	Testing set	228	578	2194	3000
	Training set	1845	4683	17772	24300
	Validation set	206	520	1974	2700
AugLy	Total data	20279	20781	21940	63000
	Testing set	2028	2078	2194	6300
	Training set	16425	16833	17772	51030
	Validation set	1826	1870	1974	5670

static embeddings, the nltk.tokenizer module [8] is adapted to the task of tokenizing, or dividing a text into its constituent parts. To tackle the OOV problem, we use the WordPiece tokenizer by Wu et al. [42], which segments each word that is not in the lexicon into subword units.

E. Word embeddings

Within the word embedding layer of a neural network model, word embeddings are employed as feature representations. Below is further information on each word embedding alternative in our experiment.

1) Word2vec

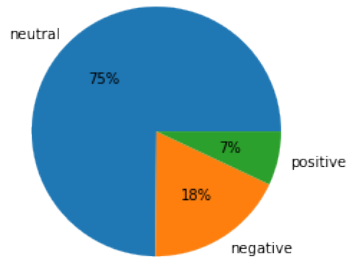
We train the word2vec model using Gensim library with our dataset. To build this model, we use a random search strategy to adjust the hyperparameters. The training hyperparameters that were utilized to obtain the word representations are as follows: (size=300, window=5, Number of negatives sampled= 1×10^{-2}, Negative= 100).

2) GloVe

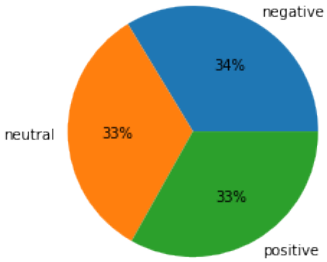
Glovepython library was used to build GloVe model. Specifically, a list of the COVID-19 reviews, each one is divided into a series of terms, which is then fed into the model. Finally, on the list, a sliding window is used to produce the representation of each unique token within the data. This study consists of a series of experiments to determine the appropriate training hyperparameters in order to develop the embedding model. The settings used are: (size=0, window=5, epochs=30, lr=0.05, alpha=0.75).

3) FastText

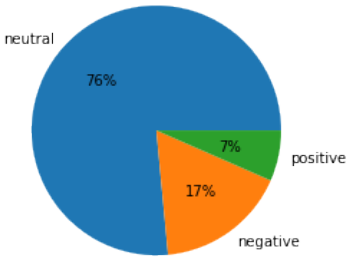
We use Gensim's native implementation of FastText to generate FastText embeddings. To enhance the values of the hyperparameters, a random search approach is applied. This



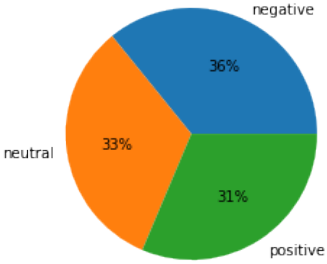
(a) COVIDSENTI without AugLy



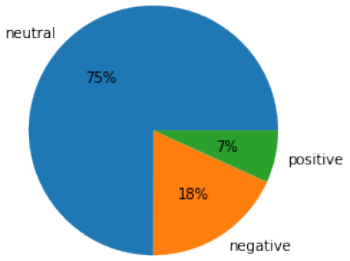
(b) COVIDSENTI with AugLy



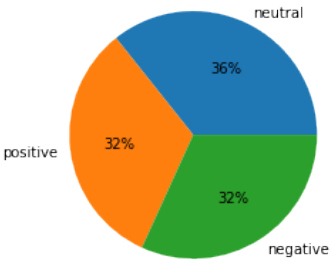
(c) COVIDSENTI-A without AugLy



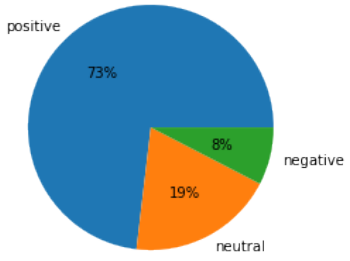
(d) COVIDSENTI-A with AugLy



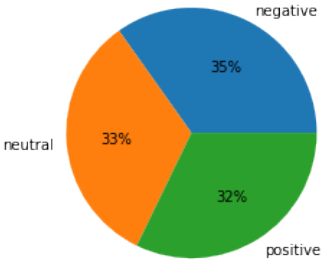
(e) COVIDSENTI-B without AugLy



(f) COVIDSENTI-B with AugLy



(g) COVIDSENTI-C without AugLy



(h) COVIDSENTI-C with AugLy

Figure. 5: Ratio of sentiment with and without AugLy using COVIDSENTI, COVIDSENTI-A, COVIDSENTI-B and COVIDSENTI-C datasets

yields vectors with 300 dimensions. The training hyperparameters used are: (size=300, window=5, Number of negatives sampled= 1×10^{-2}, Negative= 100).

F. BERT

Our experiments incorporate BERT to investigate its efficacy in SA. In this paper, we use the multilingual-base-uncased version with 12 layers, 12 attention heads, and a hidden size of 768 parameters, totaling 110M.

V. Result and discussion

We used the ADAM optimizer [21] to train the models, using a learning rate of 0.001, a batch size of 200 during 200 epochs, and categorical cross-entropy as the loss function. We employed the conventional metrics for text classification tasks to assess our models: accuracy, precision, recall,

and F1-score. Several experiments are carried out, including the usage of Simple Concat, PCA and AutoE for combining static embeddings and BERT for contextual embeddings, as well as imbalanced and balanced data generated using AugLy. Furthermore, various DL models for merging static and contextual embeddings have been investigated.

Table 5: Performance of combined word embeddings using AugLy technique for COVIDSENTI

Models	**Accuracy**		
Combinations	Simple Concat	PCA	AutoE
The first proposed approach	0.96	**0.98**	0.95
The second proposed approach	0.93	**0.95**	0.91
The third proposed approach	0.94	**0.96**	0.93
The fourth proposed approach	0.92	**0.96**	0.89

Table 6: Performance of combined word embeddings using AugLy technique for COVIDSENTI-A

Models	**Accuracy**		
Combinations	Simple Concat	PCA	AutoE
The first proposed approach	0.95	**0.97**	0.93
The second proposed approach	0.90	**0.92**	0.87
The third proposed approach	0.93	**0.95**	0.91
The fourth proposed approach	0.95	**0.96**	0.92

Table 7: Performance of combined word embeddings using AugLy technique for COVIDSENTI-B

Models	**Accuracy**		
Combinations	Simple Concat	PCA	AutoE
The first proposed approach	0.94	**0.96**	0.92
The second proposed approach	0.90	**0.93**	0.89
The third proposed approach	0.94	**0.95**	0.91
The fourth proposed approach	0.93	**0.95**	0.90

Table 8: Performance of combined word embeddings using AugLy technique for COVIDSENTI-C

Models	**Accuracy**		
Combinations	Simple Concat	PCA	AutoE
The first proposed approach	0.93	**0.96**	0.92
The second proposed approach	0.91	**0.94**	0.90
The third proposed approach	0.93	**0.95**	0.89
The fourth proposed approach	0.94	**0.96**	0.91

As shown in Tables 5, 6, 7 and 8, the significant improvements are achieved by the combination of the static embeddings using PCA. The obtained embeddings achieve respectively 98%, 95%, 96% and 96% of overall accuracy using the first, second, third and fourth approaches for COVIDSENTI. 97%, 92%, 95% and 96% of overall accuracy were obtained using the first, the second, the third and the fourth approaches, respectively for COVIDSENTI-A. 96%, 93%, 95% and 95% of overall accuracy were obtained using the first, the second, the third and the fourth approaches, respectively for COVIDSENTI-B. The obtained embeddings achieve respectively 96%, 94%, 95% and 96% of overall accuracy using the first, second, third and fourth approaches for COVIDSENTI-C.

For experiments performed on the balanced dataset using AugLy, the results of all models in terms of accuracy, precision, recall, F1 score and Macro average are shown in Tables 9, 10, 11 and 12.

Table 9: Results using the combined static embeddings with PCA and the AugLy technique for COVIDSENTI

Models	**Accuracy**	**Class**	**Precision**	**Recall**	**F1 score**
The first proposed approach	**0.98**	0	0.98	0.98	0.98
		-1	0.97	0.99	0.98
		1	0.99	0.97	0.98
		Macro avg	0.98	0.98	0.98
The second proposed approach	0.95	0	0.96	0.94	0.95
		-1	0.95	0.95	0.95
		1	0.95	0.96	0.95
		Macro avg	0.95	0.95	0.95
The third proposed approach	0.96	0	0.96	0.96	0.96
		-1	0.96	0.96	0.96
		1	0.96	0.96	0.96
		Macro avg	0.96	0.96	0.96
The fourth proposed approach	0.96	0	0.97	0.96	0.96
		-1	0.96	0.96	0.96
		1	0.96	0.97	0.97
		Macro avg	0.96	0.96	0.96
BERT only	0.95	0	0.95	0.95	0.95
		-1	0.95	0.96	0.95
		1	0.96	0.96	0.96
		Macro avg	0.95	0.95	0.95
Combined Static embeddings only	0.91	0	0.89	0.91	0.89
		-1	0.91	0.86	0.87
		1	0.92	0.93	0.92
		Macro avg	0.91	0.91	0.91

Table 10: Results using the combined static embeddings with PCA and the AugLy technique for COVIDSENTI-A

Models	**Accuracy**	**Class**	**Precision**	**Recall**	**F1 score**
The first proposed approach	**0.97**	0	0.97	0.95	0.96
		-1	0.95	0.97	0.96
		1	0.97	0.98	0.98
		Macro avg	0.96	0.97	0.97
The second proposed approach	0.92	0	0.96	0.90	0.93
		-1	0.90	0.93	0.91
		1	0.90	0.95	0.92
		Macro avg	0.92	0.92	0.92
The third proposed approach	0.95	0	0.94	0.95	0.95
		-1	0.96	0.95	0.95
		1	0.97	0.96	0.96
		Macro avg	0.96	0.95	0.95
The fourth approach	0.96	0	0.97	0.95	0.96
		-1	0.95	0.96	0.95
		1	0.96	0.97	0.97
		Macro avg	0.96	0.96	0.96
BERT only	0.94	0	0.95	0.92	0.94
		-1	0.93	0.94	0.94
		1	0.94	0.96	0.95
		Macro avg	0.94	0.94	0.94
Combined static embeddings only	0.90	0	0.88	0.90	0.89
		-1	0.90	0.85	0.87
		1	0.91	0.93	0.92
		Macro avg	0.90	0.90	0.90

Table 11: Results using the combined static embeddings with PCA and the AugLy technique for COVIDSENTI-B

Models	**Accuracy**	**Class**	**Precision**	**Recall**	**F1 score**
The first proposed approach	**0.96**	0	0.95	0.95	0.95
		-1	0.96	0.95	0.95
		1	0.96	0.97	0.97
		Macro avg	0.96	0.96	0.96
The second proposed approach	0.93	0	0.95	0.93	0.94
		-1	0.96	0.89	0.92
		1	0.87	0.96	0.91
		Macro avg	0.93	0.93	0.93
The third proposed approach	0.95	0	0.96	0.94	0.95
		-1	0.94	0.96	0.95
		1	0.95	0.96	0.95
		Macro avg	0.95	0.95	0.95
The fourth proposed approach	0.95	0	0.96	0.94	0.95
		-1	0.94	0.96	0.95
		1	0.95	0.95	0.95
		Macro avg	0.95	0.95	0.95
BERT only	0.93	0	0.94	0.91	0.93
		-1	0.92	0.94	0.93
		1	0.93	0.94	0.94
		Macro avg	0.93	0.93	0.93
Combined static embeddings only	0.87	0	0.85	0.89	0.87
		-1	0.89	0.85	0.87
		1	0.89	0.89	0.89
		Macro avg	0.88	0.87	0.87

The second series of studies involves utilizing PCA for combining static embeddings and BERT for contextual embeddings on unbalanced data. Experimental results are provided in Tables 13, 14, 15 and 16. On the balanced dataset, the performance of four DL models for merging the combined static embeddings and contextual ones was exam-

Table 12: Results using the combined static embeddings with PCA and the AugLy technique for COVIDSENTI-C

Models	Accuracy	Class	Precision	Recall	F1 score
The first proposed approach	**0.96**	0	0.95	0.95	0.95
		-1	0.97	0.94	0.96
		1	0.95	0.98	0.97
		Macro avg	0.96	0.96	0.96
The second proposed approach	0.94	0	0.98	0.91	0.94
		-1	0.92	0.96	0.94
		1	0.93	0.96	0.94
		Macro avg	0.94	0.94	0.94
The third proposed approach	0.95	0	0.95	0.94	0.95
		-1	0.95	0.95	0.95
		1	0.94	0.95	0.94
		Macro avg	0.95	0.95	0.95
The fourth proposed approach	**0.96**	0	0.95	0.95	0.95
		-1	0.95	0.96	0.95
		1	0.97	0.95	0.96
		Macro avg	0.96	0.96	0.96
BERT only	0.94	0	0.93	0.92	0.92
		-1	0.93	0.93	0.93
		1	0.93	0.95	0.94
		Macro avg	0.93	0.93	0.93
Combined static embeddings only	0.88	0	0.86	0.90	0.88
		-1	0.90	0.86	0.88
		1	0.90	0.90	0.90
		Macro avg	0.89	0.88	0.88

ined. The first proposed approach achieved the highest accuracy score of 0.98, 0.97, 0.96 and 0.96 for COVIDSENTI, COVIDSENTI-A, COVIDSENTI-B and COVIDSENTI-C. Using the imbalanced data without AugLy, the highest accuracy obtained using the first model is 0.83, 0.85, 0.83, 0.83 for COVIDSENTI, COVIDSENTI-A, COVIDSENTI-B and COVIDSENTI-C, respectively. This indicates significantly worse performance as compared to results on the balanced dataset. Class balance is responsible for the huge improvement in model performance while utilizing AugLy. The use of AugLy for data balancing also minimizes the likelihood of the model over-fitting on the majority class and helps in performance improvement.

Moreover, results show that the first proposed approach outperforms the CNN network (the same configuration adopted for our proposed hybrid approach) applied on BERT only and the combined static embeddings only. The first proposed approach achieves 0.98 accuracy score as compared to 0.95 and 0.91 using BERT only and the combined static embeddings only, respectively for COVIDSENTI. The first proposed approach achieves 0.97 accuracy score as compared to 0.94 and 0.90 using BERT only and the combined static embeddings only, respectively for COVIDSENTI-A. The first proposed approach achieves 0.96 accuracy score as compared to 0.93 and 0.87 using BERT only and the combined static embeddings only, respectively for COVIDSENTI-B. The first proposed approach achieves 0.96 accuracy score as compared to 0.94 and 0.88 using BERT only and the combined static embeddings only, respectively for COVIDSENTI-C. These findings might be explained by the fact that the first proposed technique makes use of the combined static embedding model to capture the characteristics associated with COVID-19-related terms. Furthermore, the suggested approach makes use of the BERT model, which has shown exceptional results on a range of NLP-related problems across several languages.

VI. Conclusion

Due to the rise in COVID-19 cases, governments have limited people's movement, causing a wave of fear and worry to sweep the world. As a result, a significant portion of the general population relies on social media to keep informed, particularly social media network platforms such as Facebook, Twitter, and Instagram. Coronavirus information published on social media has a direct impact on people's lives. Sometimes, it was handled positively by people and sometimes, it posed a negative impact on the daily routine. This paper proposes a new SA system. First, we proposed a new technique for dealing with the imbalanced dataset problem. Next, we examine several techniques for combining static word embeddings such as PCA. Finally, we introduced a strategy based on an attention mechanism for combining static word embeddings with contextual embeddings generated by a BERT-based language model to classify positive, negative, and neutral reviews using several DL architectures. The first proposed DL approach perform very well and achieved an accuracy of 0.98, 0.97, 0.96 and 0.96 with AugLy for COVIDSENTI, COVIDSENTI-A, COVIDSENTI-B and COVIDSENTI-C, respectively. While the same approach achieved an accuracy of 0.83, 0.85, 0.83 and 0.83 without AugLy for COVIDSENTI, COVIDSENTI-A, COVIDSENTI-B and COVIDSENTI-C, respectively. These findings show that employing data balancing with AugLy improves classification accuracy. Furthermore, since BERT is trained on Wikipedia and cannot handle emerging keywords such as trending hashtags, recent topics, and so on, the experimental findings show that combining contextual vectors (BERT in our case) and the combined static ones using PCA plays a crucial role in this context.

Table 13: Results using the combined static embeddings with PCA on the original COVIDSENTI

Models	Accuracy	Class	Precision	Recall	F1 score
The first proposed approach	0.83	0	0.97	0.97	0.97
		-1	0.93	0.94	0.94
		1	0.15	0.14	0.15
		Macro avg	0.68	0.68	0.69

Table 14: Results using the combined static embeddings with PCA on the original COVIDSENTI-A

Models	Accuracy	Class	Precision	Recall	F1 score
The first proposed approach	0.85	0	0.95	0.95	0.95
		-1	0.58	0.60	0.59
		1	0.68	0.65	0.66
		Macro avg	0.74	0.73	0.73

Table 15: Results using the combined static embeddings with PCA on the original COVIDSENTI-B

Models	Accuracy	Class	Precision	Recall	F1 score
The first proposed approach	0.83	0	0.93	0.97	0.95
		-1	0.54	0.41	0.47
		1	0.72	0.70	0.71
		Macro avg	0.73	0.69	0.71

Table 16: Results using the combined static embeddings with PCA on the original COVIDSENTI-C

Models	Accuracy	Class	Precision	Recall	F1 score
The first proposed approach	0.83	0	0.96	0.93	0.95
		-1	0.45	0.51	0.48
		1	0.70	0.72	0.71
		Macro avg	0.70	0.72	0.71

VII. Declarations

Funding We acknowledge financial support for this research from the Ministry of Higher Education, Scientific Research

and Innovation, the Digital Development Agency (DDA) and the CNRST of Morocco (Alkhawarizmi/2020/01).

Availability of data and materials Data sharing is not applicable to this article as no new data were created or analyzed in this study.

References

[1] G. I. Ahmad, J. Singla, A. Ali, A. A. Reshi, and A. A. Salameh. Machine Learning Techniques for Sentiment Analysis of Code-Mixed and Switched Indian Social Media Text Corpus: A Comprehensive Review. *Machine Learning*, 13(2), 2022.

[2] A. Balahur. Sentiment analysis in social media texts. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 120–128, 2013.

[3] C. Baraniuk. What the Diamond Princess taught the world about covid-19. *Bmj*, 369, 2020.

[4] N. Bensalah, H. Ayad, A. Adib, and A. I. E. Farouk. Combining Static and Contextual Features: The Case of English Tweets. In *Emerging Trends in Intelligent Systems & Network Security*, pages 168–175, 2023.

[5] N. Bensalah, H. Ayad, A. Adib, and A. I. e. farouk. Sentiment Analysis in Drug Reviews Based on Improved Pre-trained Word Embeddings. In *Innovations in Smart Cities Applications Volume 6*, pages 87–96, 2023.

[6] N. Bensalah, H. Ayad, A. Adib, and A. Ibn El Farouk. Arabic Sentiment Analysis Based on 1-D Convolutional Neural Network. In *International Conference on Smart City Applications,SCA20*, 2020.

[7] N. Bensalah, H. Ayad, A. Adib, and A. Ibn el Farouk. Arabic Machine Translation Based on the Combination of Word Embedding Techniques. In *Proceedings of the 14th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2022)*, pages 60–69, 2023.

[8] S. Bird and E. Loper. NLTK: the natural language toolkit. Association for Computational Linguistics, 2004.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[10] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching Word Vectors with Subword Information. *Trans. Assoc. Comput. Linguistics*, 5:135–146, 2017.

[11] S. E. Clark, M. C. Bledsoe, and C. J. Harrison. The role of social media in promoting vaccine hesitancy. *Current opinion in pediatrics*, 34(2):156–162, 2022.

[12] S. Das, D. Das, and A. K. Kolya. Sentiment classification with GST tweet data on LSTM based on polarity-popularity model. *Sādhanā*, 45(1):1–17, 2020.

[13] C. Del Rio and P. N. Malani. 2019 novel coronavirus—important information for clinicians. *Jama*, 323(11):1039–1040, 2020.

[14] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, pages 4171–4186, 2019.

[15] A. O. Docea, A. Tsatsakis, D. Albulescu, O. Cristea, O. Zlatian, M. Vinceti, S. A. Moschos, D. Tsoukalas, M. Goumenou, N. Drakoulis, et al. A new threat from an old enemy: Re-emergence of coronavirus. *International journal of molecular medicine*, 45(6):1631–1643, 2020.

[16] W. Etaiwi, D. Suleiman, and A. Awajan. Deep Learning Based Techniques for Sentiment Analysis: A Survey. *Informatica*, 45(7), 2021.

[17] J. Han, K. Qian, M. Song, Z. Yang, Z. Ren, S. Liu, J. Liu, H. Zheng, W. Ji, T. Koike, et al. An early study on intelligent analysis of speech under COVID-19: Severity, sleep quality, fatigue, and anxiety. *arXiv preprint arXiv:2005.00096*, 2020.

[18] D. Huremović. Brief history of pandemics (pandemics throughout history). In *Psychiatry of pandemics*, pages 7–35. 2019.

[19] H. Ide and T. Kurita. Improvement of learning for CNN with ReLU activation by sparse regularization. In *2017 International Joint Conference on Neural Networks, IJCNN*, pages 2684–2691, 2017.

[20] H.-J. Kim, Y. K. Jeong, Y. Kim, K. Kang, and M. Song. Topic-based content and sentiment analysis of Ebola virus on Twitter and in the news. *Journal of Information Science*, 42, 2015.

[21] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR*, 2015.

[22] M. Koroteev. BERT: a review of applications in natural language processing and understanding. *arXiv preprint arXiv:2103.11943*, 2021.

[23] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *8th International Conference on Learning Representations, ICLR*, 2020.

[24] S. Loria. textblob Documentation. *Release 0.15*, 2, 2018.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR*, 2013.

[26] V. Nair and G. E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[27] U. Naseem, K. Musial, P. W. Eklund, and M. Prasad. Biomedical Named-Entity Recognition by Hierarchically Fusing BioBERT Representations and Deep Contextual-Level Word-Embedding. In *2020 International Joint Conference on Neural Networks, IJCNN*, pages 1–8, 2020.

[28] U. Naseem, I. Razzak, M. Khushi, P. W. Eklund, and J. Kim. COVIDSenti: A large-scale benchmark Twitter data set for COVID-19 sentiment analysis. *IEEE Transactions on Computational Social Systems*, 8(4):1003–1015, 2021.

[29] L. Nemes and A. Kiss. Social media sentiment analysis based on COVID-19. *Journal of Information and Telecommunication*, 5(1):1–15, 2021.

[30] A. Ortis, G. M. Farinella, G. Torrisi, and S. Battiato. Visual sentiment analysis based on on objective text description of images. In *2018 international conference on content-based multimedia indexing (CBMI)*, pages 1–6, 2018.

[31] Z. Papakipos and J. Bitton. AugLy: Data Augmentations for Robustness. *CoRR*, abs/2201.06494, 2022.

[32] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[33] M. Shah Jahan, H. U. Khan, S. Akbar, M. Umar Farooq, S. Gul, and A. Amjad. Bidirectional Language Modeling: A Systematic Literature Review. *Scientific Programming*, 2021, 2021.

[34] T. Singhal. A review of coronavirus disease-2019 (COVID-19). *The indian journal of pediatrics*, 87(4):281–286, 2020.

[35] Y. Sun. A Systematic Review of Celebrity Effect and Its Impact On the Consumer Economy. In *2021 International Conference on Social Development and Media Communication (SDMC 2021)*, pages 777–782. Atlantis Press, 2022.

[36] S. Tuli, S. Tuli, R. Tuli, and S. S. Gill. Predicting the growth and trend of COVID-19 pandemic using machine learning and cloud computing. *Internet of Things*, 11, 2020.

[37] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference ICML*, 2008.

[38] C. Wang, P. Nulty, and D. Lillis. A Comparative Study on Word Embeddings in Deep Learning for Text Classification. In *NLPIR 2020: 4th International Conference on Natural Language Processing and Information Retrieval*, pages 37–46, 2020.

[39] D. Wang, B. Hu, C. Hu, F. Zhu, X. Liu, J. Zhang, B. Wang, H. Xiang, Z. Cheng, Y. Xiong, et al. Clinical characteristics of 138 hospitalized patients with 2019 novel coronavirus–infected pneumonia in Wuhan, China. *Jama*, 323(11):1061–1069, 2020.

[40] C. O. WHO et al. World health organization. *Responding to Community Spread of COVID-19. Reference WHO/COVID-19/Community_Transmission/2020.1*, 2020.

[41] J. Wu, T. Zhu, J. Zhu, T. Li, and C. Wang. A Optimized BERT for Multimodal Sentiment Analysis. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2023.

[42] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *CoRR*, abs/1609.08144, 2016.

[43] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS*, pages 5754–5764, 2019.

Author biographies

Nouhaila Bensalah received her M.Sc. degree in 2018 in Informatics and Telecommunications from the Department of Physics, Faculty of Sciences, Mohammed V University, Rabat, Morocco. She is currently working toward a Ph.D. degree from the LIM Laboratory of Informatics, Faculty of Sciences and Techniques, Mohammedia, Morocco. She has eight publications at international and national conferences. Her current research interests include Machine Learning and Natural Language Processing, especially Arabic Machine Translation.

Habib Ayad received his Ph.D. in Computer Science from the National School of Applied Sciences of Marrakech, CADI AYYAD University in 2013. He is currently an associate professor at Hassan II University in Casablanca. His research focuses on data science, Artificial Intelligence, machine learning, deep learning, Natural Language Processing. Habib Ayad is also a member of the ACM Casablanca Chapter.

Abdellah Adib received the Doctorat de 3rd Cycle and the Doctorat d'Etates-Sciences degrees in Statistical Signal Processing from the Mohammed V University, Rabat, Morocco, in 1996 and 2004, respectively. Since 1997, he has been an assistant professor at the Scientific Institute of Rabat and a professor of higher education at the Faculty of Science and Technology of Mohammedia since 2008. He was Head of the Department between 2012 and 2015. He was a member of the Scientific Committee of FSTM for two terms, 2015-2018 and 2018-2021. He was also a member of the CNRST scientific committees as well as an expert evaluator for information technologies for two consecutive terms, 2013-2016 and 2016-2020. Since 1993, his research has focused on automatic information processing, source separation, and applications (seismic, biomedical, and speech signals). He is also the author or co-author of more than 30 papers in international journals and more than 80 papers in international conferences. He has been a member of several technical committees of IEEE, EURASIP and SPRINGER, ... He has supervised more than 20 theses in different fields related to his favorite areas.

Abdelhamid Ibn El Farouk is a full professor at Hassan II University in Casablanca. Currently, he serves as the dean of the Faculty of Letters and Humanities in Mohammedia. He is a linguist, sociolinguist, and translator, holding a doctorate thesis from the University of René Descartes in 1994 on the verbal system of Written Arabic and a Doctorat d'Etat in 1996 in the functional grammar of written Arabic from the Hassan II of Casablanca. He has several publications in the fields of his specialization, linguistics, sociolinguistics, and translation. since March 2018, Dr. Ibn El Farouk Abdelhamid has held the position of Dean of the Faculty of Letters and Human Sciences in Mohammedia. Additionally, he serves as the Director of the Research Center for Translation in Humanities since December 2020, where he oversees research projects related to translation in the field of human sciences.