

CIMAT-NLP-GTO at MentalRiskES 2023: Early Detection of Mental Disorders in Spanish Messages using Style Based Models and BERT Models

Franklin Echeverría-Barú^{1,*}, Fernando Sánchez-Vega^{1,2} and
Adrián Pastor López-Monroy¹

¹Mathematics Research Center (CIMAT), Jalisco S/N Valenciana, 36023 Guanajuato, GTO México

²Consejo Nacional de Ciencia y Tecnología (CONACYT), Av. de los Insurgentes Sur 1582, Benito Juárez, 03940, CDMX, México.

Abstract

This paper presents the methods submitted by the CIMAT-NLP-GTO team for participation in the MentalRiskES 2023 shared tasks, which focus on the early detection of eating disorders, depression, and anxiety. Our approach is based on two main ideas: First, we investigate the potential of linguistic patterns in messages and their writing styles to identify signs of mental disorders. To achieve this, we generate vector-based representations of each user by analyzing n-grams of characters. Second, we explore the use of pre-trained models, fine-tuned specifically for mental disorder detection in the Spanish language, to leverage their understanding of language. Based on these two approaches, we aim to develop effective methodologies for the early identification of mental health risks. The potential of these approaches is demonstrated, as we achieved good results in most tests and obtained the first place in the competition for the tasks of eating disorders and anxiety.

Keywords

Eating disorders, Depression, Anxiety, Writing Style Analysis, Pre-trained models

1. Introduction

Mental health is a crucial aspect of overall well-being, and early detection of potential health risks can play a vital role in the prevention and treatment of mental disorders. According to the World Health Organization [1], depression is one of the leading causes of disability, and suicide is the fourth leading cause of death among 15-29-year-olds. People with severe mental health conditions may die prematurely. With the increasing prevalence of mental health problems and the growing popularity of social networks, it is of great importance to take advantage of these platforms for rapid identification and intervention.

The MentalRiskES workshop at IberLEF [2] provides a platform for the development of methodologies and practical approaches for the early detection of various mental illnesses. In this paper, we present our methods submitted for participation in this competition. Our


IberLEF 2023, September 2023, Jaén, Spain

*Corresponding author.

✉ franklin.echeverria@cimat.mx (F. Echeverría-Barú); fernando.sanchez@cimat.mx (F. Sánchez-Vega);
pastor.lopez@cimat.mx (A. P. López-Monroy)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

approaches focus mainly on two ideas: one based on traditional machine learning models and the other on modern deep neural networks.

The first set of methods builds upon works such as [3], [4], and [5], which apply techniques that analyze writing style for Authorship Attribution. In turn, works such as [6] show that the use of character n-grams can be useful when applied to social media text. For this reason, our first approach involves generating vector-based representations of the messages of each user by analyzing character n-grams, allowing us to examine and evaluate their writing style. Our hypothesis is that these linguistic patterns can provide valuable information to identify whether a user exhibits signs of a mental disorder. Given the limited data available for this task, we developed a vector-based representation using a Bag of Character n-grams.

Additionally, as our second approach, we employed BERT models [7], particularly pre-trained BERT models. These have been observed to capture semantic and syntactic information at different levels of BERT's layers [8]. Our goal is to leverage the possible structural knowledge about language that these models possess and specifically adjust them for mental disorder detection.

The remainder of the paper is organized as follows. Section 2 describes the competition and tasks addressed, including the available dataset and the evaluation metrics. Section 3 presents the developed method, while the results achieved are shown and discussed in Section 4. Section 5 presents the main conclusions and future work lines, Finally Section 6 presents technical details.

2. Competition Overview

The MentalRiskES 2023 competition, part of the Iberian Languages Evaluation Forum (IberLEF) 2023, aims to detect early signs of mental disorders in Spanish. The organizers provide a more detailed description of the competition in [2]. In this section, we will give a brief description of the tasks that our team participated in.

2.1. Tasks

Our team participated in all 3 tasks of the competition, each of which focused on a different mental disorder: Task 1 on eating disorders, Task 2 on depression, and finally Task 3, which was an unknown disorder during the competition and was later revealed to be on anxiety. Each task was divided into different subtasks. Our team participated in the binary classification and simple regression subtasks of each task.

2.2. Datasets

The organizers built a new Spanish dataset for each task. Each task's dataset is divided as follows:

- Eating Disorders: 175 users for training, 10 for trial, and 150 for testing.
- Depression: 175 users for training, 10 for trial, and 150 for testing.
- Anxiety: Only 150 test users are provided; no corpus for train and trial is given.

3. Proposed Model

In this section, we describe the methods that were adapted and later utilized for the competition. On one hand, we adapted classical machine learning models such as the Bag of Characters and SVMs to focus on analyzing writing styles. On the other hand, we built modern neural models that leverage the use of pre-trained transformers and their knowledge of structural, syntactical, and semantic information about the language. In the following subsections, we will describe these two types of methods as well as the data augmentation technique that was used and details about the models submitted for the competition.

3.1. Models for Writing Style Analysis

Our main hypothesis is that individuals with certain mental conditions exhibit similar patterns and styles of writing. For this reason, we built a method that focused on the analysis of writing style characteristics. The method consists of three components: pre-processing, a feature extractor module to analyze writing style, and a regression or classification module depending on the subtask. Each of these parts is described below.

3.1.1. Pre-processing

First, preprocessing was applied to the data. This process consisted of removing links and stopwords, normalizing text to lowercase, and retaining all numbers and punctuation marks.

3.1.2. Feature Extractor Module

To analyze the writing style of a user, we focused on studying character-level n-grams. For this purpose, we proposed the use of a Bag of Characters as a feature extractor unit, which generates a vector representation of the message history of a user.

The weight scheme used for the Bag of Characters was the classic tf-idf weighting with sublinear tf. This module with this weight scheme was built using Sklearn's *TfidfVectorizer* class [9].

3.1.3. Regression Module

For the simple regression subtask, it is necessary to obtain a coefficient. To obtain this regression value, we decided to train a classification unit on the previous vector representations. Then, extract from it a value between 0 and 1 that can be interpreted as the model confidence level that the user belongs to the positive class.

Specifically, two variants were used for the classification unit:

- First, a linear kernel SVM [9]. In the rest of this article, we will refer to this method as **Style-SVM**.
- Secondly, a multinomial Naive Bayes classifier [9]. We will refer to this method as **Style-NB**.

3.1.4. Binary Classification Module

For the classification task, we built upon the regression module by introducing a selection threshold. In other words, a user is only considered positive if their confidence level exceeds the threshold. This approach was designed to improve overall prediction accuracy compared to classical classification methods such as SVM or Naive Bayes and to address potential class imbalances.

To determine the threshold, we first obtained the confidence levels of the regression module for the validation set. Then we searched for the optimal value between 0 and 1 that separated the users into positive or negative classes in a way that maximized the F1 score of the positive class. This value was then established as the selection threshold.

Once again, we have two method variants depending on whether an SVM or Naive Bayes is used as the classification unit. Therefore, we will refer to these methods again as **Style-SVM** and **Style-NB**, respectively. The context of the subtask in which the method is used will indicate whether the method uses a regression or binary classification module.

3.2. Transformer-Based Models

Transformer models [10], particularly BERT architectures [7], have been observed to outperform previous state-of-the-art models in several NLP tasks [11]. Studies on specific BERT models have shown that they have been able to learn certain information about the structure of the English language. Different levels of BERT layers have been found to capture different information about the language, with middle layers capturing syntactic features and upper layers capturing semantic features [8]. Based on these findings, we developed two types of architecture and two training methods that could potentially leverage the semantic, syntactic or general knowledge about language structure that BERT models trained in Spanish can possess.

3.2.1. Architecture

Two architectures have been developed. One of the architectures aims to capture detailed structural information about the language, while the other aims to capture structural information at a more macro level. These architectures are divided into four parts, which we describe next.

BERT Module

The first part of the architecture of the method and its two variants are shown in Figure 1. This module processes the input sequence using a pre-trained BERT model and extracts different information from it, giving rise to the two types of architectures developed.

- The first architecture aims to leverage the structural information captured by the language at a more macro level. For this reason, only the pooler output is extracted from BERT. This vector represents the last layer hidden-state of the classification token after being processed by the pooler module of the BERT architecture. In the rest of this article, we will refer to this architecture as BERT one output or **BERT-1Out**.

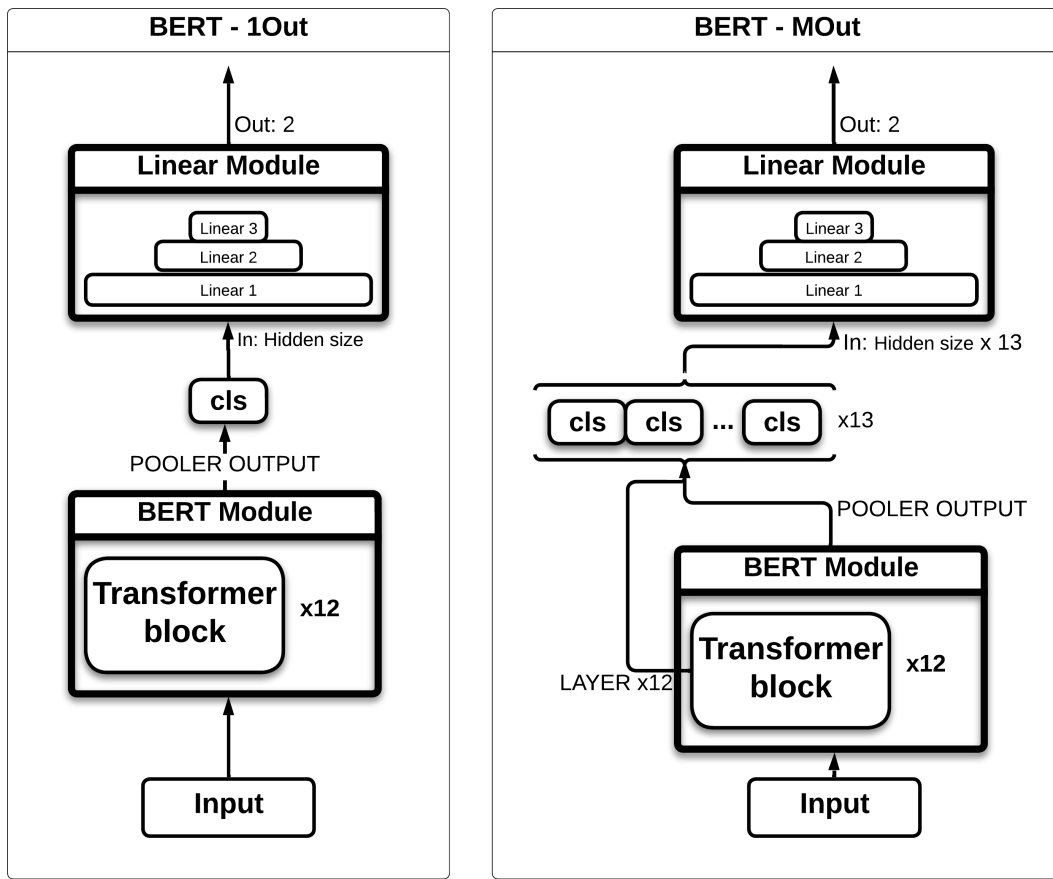


Figure 1: Variants of the BERT module architecture. On the left is the BERT-1Out architecture, note that only the pooler output is extracted from BERT. On the right is the BERT-MOut architecture, where both the pooler output and all the final hidden-states of the classification token for each of the 12 layers or transformer blocks are extracted from BERT.

- The second architecture aims to leverage more detailed structural information. As mentioned above, each layer of BERT can capture different structural information. For this reason, in addition to extracting the pooler output, all final hidden-states of the classification token for each layer of BERT were extracted. Finally, these vectors are concatenated into a single vector so that it can be processed by the next module.

We will refer to this architecture as BERT multi-output or **BERT-MOut**.

This module was implemented using the *BERT* class within the HuggingFace library [12]. Specifically, the pre-trained BERT model used for all experiments was *roBERTa* [13], [14], [15]. We decided to use this model as it is a model trained in Spanish and particularly on social media texts.

Linear Module

The second part of the architecture consists of a linear module that aims to process the linguistic information extracted by the BERT Module to obtain a probability of belonging to each class. This module consists of 3 linear layers. The first two linear layers reduce the dimension by half, while the last layer reduces it to two dimensions and is activated with a LogSoftmax function to obtain log probabilities for each class.

Maxpool Filter

Another important part of the architecture is a maximum activation filter that aims to help the method process the entire message history of a user, regardless of the number of messages they have, and to assist with early detection of mental illness.

One of the first challenges encountered when trying to process social media messages is the highly variable number of messages between users. Additionally, BERT models have a maximum sequence length that they can process in practice. The challenge then is to be able to process the entire message history for each user.

To address this difficulty, we divided the sequence corresponding to the entire message history into smaller subsequences. This allowed each subsequence to be fully processed by the BERT module and its corresponding outputs to be processed by the linear module. This resulted in a two-dimensional vector for each sub sequence, with each vector representing the probability that the model associates with each class for the respective subsequence.

We then needed a way to condense the information from each sub sequence into a single vector while also promoting early detection of mental illness. To achieve this, we introduced a maximum activation filter in which the final probability for each class is determined by selecting the maximum probability from all vectors for that class. In this way, the probability associated with a class is only updated when stronger evidence is found.

Figure 2 shows the complete flow of information, how the input sequence is processed and how the BERT module, the Linear module, and the MaxPool filter interact.

Classification and Regression module

Finally, the only part of the architecture left to describe is how classification or regression is performed according to the subtask.

The classification is performed by taking the class with the highest probability from the two-dimensional vector obtained in the previous module as the user class.

To obtain a regression coefficient, we built an MLP with 3 linear layers on top of the Maxpool Filter. The last layer is activated by a sigmoid function to obtain a value between 0 and 1, interpretable as a regression value.

3.2.2. Fine-Tuning

One of the reasons for using pre-trained BERT models was to take advantage of the structural information about the language that they can store in different layers. With this goal in mind and to improve the performance of the method, we developed two types of fine-tuning.

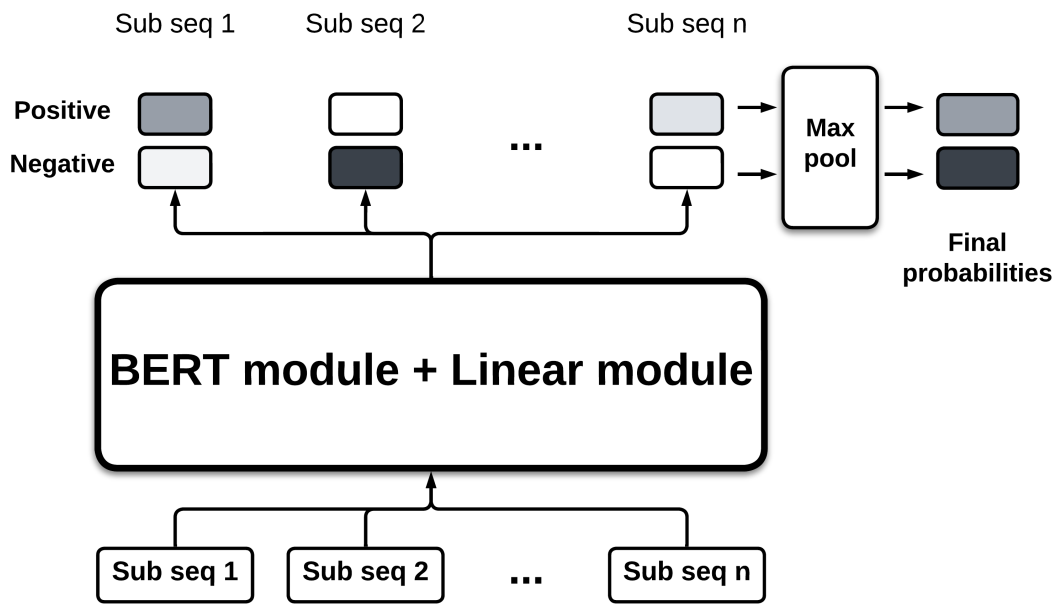


Figure 2: General visualization of the architecture. Each input sequence is divided into subsequences, each of which is independently processed by the BERT and linear modules to obtain a probability of belonging to the positive and negative classes. Subsequently, the MaxPool filter is applied to obtain the final probabilities.

Both consist of staged specialization training aimed at training both the linear module and adjusting the knowledge stored in different layers of BERT to better adapt to the task of detecting mental illness.

In detail, the two variants of fine-tuning are as follows:

- The first consists of 3 stages. In the first stage, only the Linear Module with the Maxpool is unfrozen. In the second stage, the pooler layer of the BERT Module is also unfrozen. In the third stage, the last linear layer of each layer of the BERT Module is also unfrozen. We will refer to this type of training as 3-stage Fine-Tuning or **Fn-3S**.
- The second aims to have a higher level of refinement and adjustment of knowledge by adding an extra stage to the previous 3 stages. In the fourth stage, the entire last module of linear layers of each layer of the BERT Module is also unfrozen. We will refer to this type of training as 4-stage Fine-Tuning or **Fn-4S**.

The training of the regression module was done separately and after the fine-tuning was completed. For each user, the respective 2-dimensional vectors obtained with the fine-tuned method were generated and used as training data for the regression module.

3.3. Data Augmentation

Due to the limited amount of training data available for the competition, we proposed a data augmentation. This involved combining the corpora provided for Task 1 with those of Task 2. Experiments carried out during development showed that this approach could, in some cases, improve model performance. Additionally, we believed that this could enhance the model’s robustness for the general detection of unseen mental disorders.

3.4. Details on Models for Competition

The methods submitted for the competition consist of a voting ensemble of several individual methods as described earlier. This was done to stabilize the predictions and improve the overall performance of the method.

4. Experiments

4.1. In-House Experiments

In this section, we briefly describe the experiments we conducted before submitting to the competition.

4.1.1. Experimental Setup

Since the organizers only provided a train and trial set during the development phase, we decided to combine these two sets to form a slightly larger training dataset. Additionally, we set aside 30% of the training set as a validation set.

We used 5-fold cross-validation on this dataset over 5 different seeds to evaluate the performance of each individual model. Then, the ensembles were formed by five individual models, each trained on a different fold of the cross-validation. Each individual model of the ensemble was selected as the best-performing seed.

Details on Style-Based Models

We conducted a hyperparameter search for the Bag of Character n-grams methods, as well as experimented with various text preprocessing techniques. Our observations revealed that the best performance was achieved by using n-gram ranges from 1 to 5, a minimum document frequency of 4 and the preprocessing described earlier. Similarly, we conducted a grid-search for the C or regularization parameter of the SVM used.

Details on Transformer-Based Models

Since the pre-trained model used was roBERTuito, we decided to fix the subsequence length at 125. That is, the sequence corresponding to the entire message history of a user is divided into subsequences of length 125.

Additionally, in order to maximize the utility of the small number of users in the datasets, we decided to follow a general processing strategy for the training set for all transformer-based

models. If for a user the length of the sequence corresponding to their entire message history exceeded the subsequence length (125), their history was divided into subsequences of length 125. Each of these subsequences was then taken as the complete history of a “new” training user. In this way, training focused on learning patterns over subsequences rather than on the entire message history.

The loss function used was negative logarithmic likelihood loss, implemented with PyTorch’s *NLLLoss* class [16]. The number of epochs for fine-tuning each individual model was 30 epochs with an early stopping patience of 10 epochs for each stage. We set the learning rate to 1e-5 and a batch size of 32. We use the validation set to evaluate training performance and save model with the best F1 score.

4.1.2. Results and Analysis

We experimented with various neural architectures, including CNN, LSTM, Bi-LSTM, as well as decision trees and other types of regression modules for the style-based methods. We also experimented with simpler architectures for transformer-based models, such as setting a maximum sequence length for a user’s entire message history or taking the average instead of using the maximum activation filter. However, we observed that the performance of all these models was much lower than that of the methods described earlier in this article. For this reason, and to avoid extending the article, we only describe and present results related to the methods described in previous sections.

For clarity, transformer-based models are constructed from an architecture (BERT-1Out or BERT-MOut) and a fine-tuning method (Fn-3S or Fn-4S). The +Aug label indicates that data augmentation was applied as part of the model training.

Table 1 shows the F1 score of some individual models. It can be observed that transformer-based models show the best performance. On the other hand, the style-based model with Naive Bayes shows a slight but consistent improvement over its counterpart using SVM. Another observation we made is that the combination of architecture and fine-tuning is important, as can be seen when combining 3-stage Fine-Tuning with the BERT multi-output architecture, which lowered the performance of the method considerably.

On the other hand, Table 2 shows the performance of the ensembles in the F1 and RMSE metrics. Notice how the transformer-based models showed again the best performance for these experiments. From these experiments, we noticed that data augmentation could improve the performance of a model as can be seen from the BERT-MOut Fn-4S model which improves its performance from 0.72 in Task 2 to 0.92 after data augmentation. It was also observed that the RMSE of neural models is lower than that of style-based models. This may be due to the fact that a specialized neural module was trained to obtain the regression value. It can also be noted that the ensemble manages to improve the performance of individual models, such as for style-based models for Task 1.

4.2. Participation in Competition

From the previous results, those with the best F1 score and RSME performance were selected to participate in the competition.

<i>Model</i>	<i>Task 1</i>	<i>Task2</i>
Syle-NB	0.89	0.77
Syle-SVM	0.88	0.75
BERT-MOut Fn-4S	0.98	0.72
BERT-MOut Fn-3S	0.36	0.51
BERT-1Out Fn-3S	0.84	0.97

Table 1

Summary of F1 score of different individual models. Notice that transformer-based model have the best performance.

<i>Model</i>	<i>Task 1</i>		<i>Task 2</i>	
	<i>F1</i>	<i>RMSE</i>	<i>F1</i>	<i>RMSE</i>
BERT-1Out Fn-3S	0.85	0.06	0.98	0.03
BERT-1Out Fn-4S	0.81	0.09	0.97	0.04
BERT-MOut Fn-4S	0.98	0.02	0.72	0.24
BERT-MOut Fn-4S + BERT-1Out Fn-4S	0.89	0.03	0.96	0.06
BERT-MOut Fn-4S + Aug	0.95	0.02	0.92	0.05
Style + NB	0.92	0.07	0.74	0.21
Style + NB + Aug	0.93	0.03	0.95	0.06

Table 2

Summary of performance of ensembles. Notice that ensembles improved the performance of individual models such as style-

4.2.1. Description of the Model Used on Each Task

- For Task 1, we have three runs: **Syle-NB** (Run 0), **BERT-MOut Fn-4S** (Run 1), and **BERT-MOut Fn-4S + Aug** (Run 2).
- For Task 2, we also have three runs: **Syle-NB** (Run 0), **BERT-1Out Fn-3S** (Run 1), and **BERT-1Out Fn-4S** (Run 2).
- For Task 3, we have **Syle-NB + Aug** (Run 0), **BERT-1Out Fn-3S** (Run 1), and **Run 2 T1 + Run 2 T2** (Run 2). This last model consists of an ensemble of the models sent as run 2 for task 1 (BERT-MOut Fn-4S + Aug) and for task 2 (BERT-1Out Fn-4S).

4.2.2. Results and Analysis

We present the performance results of our models in the test dataset of the competition. Additionally, we present the rankings that the models achieved in the competition. These rankings are based on a particular metric. For classification metrics, the ranking is based on Macro-F1. For latency metrics, the ranking is based on ERDE30. For regression metrics, the ranking is based on RMSE and P@30, depending on the case. Finally, we add the average performance of all contestants in the competition.

In general, it was observed that for each challenge, at least one of the runs achieved a good position in the ranking. Furthermore, at least one achieved the best or near-best performance in the competition for each metric. From Table 3, it can be seen that the style-based model

<i>Model</i>	<i>Classification</i>			<i>Latency</i>		
	<i>Rank</i>	<i>Accuracy</i>	<i>Macro-F1</i>	<i>Rank</i>	<i>ERDE 30</i>	<i>Latency-Weighted F1</i>
Syle-NB	1	0.967	0.966	1	0.018	0.863
BERT-MOut Fn-4S	8	0.847	0.847	3	0.065	0.761
BERT-MOut Fn-4S + Aug	18	0.720	0.715	16	0.119	0.676
Average of competition	-	0.764	0.748	-	0.127	0.702

Table 3

Results on Task1 subtask of binary classification. Style-based models achieved the best performance in most of the evaluation metrics.

<i>Model</i>	<i>Rank</i>	<i>Regression</i>		<i>Precision</i>	
		<i>RMSE</i>	<i>Pearson Coeff</i>	<i>Rank</i>	<i>P@30</i>
Syle-NB	15	0.348	0.906	2	0.867
BERT-MOut Fn-4S	2	0.192	0.885	16	0.633
BERT-MOut Fn-4S + Aug	4	0.200	0.864	4	0.867
Average of competition	-	0.295	0.704	-	0.713

Table 4

Results on Task1 subtask of simple regression. Style-based methods achieved the best pearson coefficient and our transformer-based models got near to the best RSME.

<i>Model</i>	<i>Classification</i>			<i>Latency</i>		
	<i>Rank</i>	<i>Accuracy</i>	<i>Macro-F1</i>	<i>Rank</i>	<i>ERDE 30</i>	<i>Latency-Weighted F1</i>
Syle-NB	13	0.651	0.635	9	0.175	0.665
BERT-1Out Fn-3S	15	0.638	0.621	13	0.187	0.666
BERT-1Out Fn-4S	20	0.624	0.602	15	0.199	0.662
Average of competition	-	0.617	0.579	-	0.232	0.599

Table 5

Results on Task2 subtask of binary classification. The style-based methods got the best results among our models.

achieved excellent performance on binary classification for Task 1. Furthermore, from Table 5, it can be seen that it was the best model among the 3 runs submitted. This provides evidence to support the study of writing patterns and styles to capture relevant information that can help identify users with mental disorders.

On the other hand, transformer-based models also performed well, as shown in Table 4, achieving rankings 2 or 4 in the regression subtask for Task 1. Another example can be seen in Table 6, which shows rankings 2, 3, and 5 for the regression subtask for Task 2.

In general, machine learning-based models showed very good performance for both Tasks 1 and 2. However, they struggled to generalize the detection of mental disorders, as can be observed in Table 7 and Table 8. It is in this challenge where no training set was provided that transformer-based models demonstrate a greater ability to generalize. In particular, Run 2 for Task 3 was designed with the objective of mixing different types of training and datasets used

<i>Model</i>	<i>Rank</i>	<i>Regression</i>		<i>Precision</i>	
		<i>RMSE</i>	<i>Pearson Coeff</i>	<i>Rank</i>	<i>P@30</i>
Syle-NB	9	0.367	0.632	3	0.567
BERT-1Out Fn-3S	2	0.292	0.645	5	0.567
BERT-1Out Fn-4S	3	0.294	0.630	8	0.533
Average of competition	-	0.367	0.444	-	0.362

Table 6

Results on Task2 subtask of simple regression. Notice that both style and transformer based models achieve near to the best RMSE and P@30.

<i>Model</i>	<i>Rank</i>	<i>Classification</i>		<i>Rank</i>	<i>Latency</i>	
		<i>Accuracy</i>	<i>Macro-F1</i>		<i>ERDE 30</i>	<i>Latency-Weighted F1</i>
Syle-NB + Aug	7	0.663	0.593	11	0.283	0.654
BERT-1Out Fn-3S	9	0.653	0.516	8	0.232	0.703
Run 2 T1 + Run 2 T2	1	0.773	0.740	3	0.188	0.757
Average of competition	-	0.671	0.556	-	0.234	0.716

Table 7

Results on Task3 subtask of binary classification. Notice that the ensemble of Run 2 from previous subtasks achieved the best Accuracy and Macro-F1 score and near to the best ERDE30.

<i>Model</i>	<i>Rank</i>	<i>Regression</i>		<i>Precision</i>	
		<i>RMSE</i>	<i>Pearson Coeff</i>	<i>Rank</i>	<i>P@30</i>
Syle-NB + Aug	6	0.367	0.385	9	0.467
BERT-1Out Fn-3S	3	0.329	0.479	7	0.533
Run 2 T1 + Run 2 T2	5	0.348	0.576	3	0.667
Average of competition	-	0.394	0.348	-	0.549

Table 8

Results on Task3 subtask of simple regression. Our methods achieved near to the best metrics on the competition.

for training. It is observed that this model achieved the best ranking among competing teams for the classification subtask, as well as rankings 3 and 5 for regression subtask.

5. Conclusions

We presented the methods adapted by the CIMAT-NLP-GTO team for the MentalRiskES 2023 Workshop tasks, which focused on the early detection of eating disorders, depression, and anxiety. Our approach involved analyzing writing style features using machine learning models and utilizing pre-trained transformer models for deep learning-based detection.

Our best-performing approach for writing style analysis involved using a Bag of Characters as a feature extractor and Naive Bayes classifiers. This approach showed promising results in the binary classification tasks. In addition, we employed the roBERTuito model with a specific

architecture to leverage the structural, syntactic, and semantic knowledge that the model may possess, fine-tuning it for each task. The results showed that our models effectively captured relevant information from user messages for mental classification.

In conclusion, our study demonstrates the potential of analyzing writing style features and leveraging pre-trained transformer models for early detection of mental disorders. The combination of machine learning and deep learning approaches can provide valuable insights for identifying individuals at risk and facilitating timely intervention.

6. Ethical issues

The development of automated methods for detecting mental health deterioration holds significant value in facilitating timely interventions and identifying relapses during the progression of mental health conditions. However, it is crucial to ensure that the employment of these methods adheres to stringent ethical guidelines. Careful monitoring is necessary to prevent potential misuses, such as the preemptive identification of individuals who may be prone to developing short-term mental illnesses. The inappropriate deployment of these methods may result in population segmentation and perpetuate biases against already vulnerable communities. Such biases could manifest in various forms, including discrimination during job interviews or when accessing essential services such as medical insurance.

7. Technical details

Regarding the efficiency metrics, unfortunately, our team encountered compatibility issues between the libraries used for our models and Code Carbon [17]. Due to time constraints, we were unable to resolve these incompatibilities and therefore, we are unable to report efficiency metrics.

The hardware used for the development and training of the methods includes an HPE ProLiant XL270d Gen 10 server with 40 cores and 256 GB of RAM memory, as well as an NVIDIA Tesla V100 32GB SXM2 card.

Acknowledgments

Echeverría-Barú acknowledges *Consejo Nacional de Ciencia y Tecnología* (CONACyT) for its master's degree grant (CVU 1064186) that funded this research. The authors thank to CONACyT, CIMAT and *Instituto Nacional de Astrofísica, Óptica y Electrónica* (INAOE) for the computer resources provided through the INAOE Supercomputing Laboratory's Deep Learning Platform for Language Technologies (*Plataforma de aprendizaje profundo para tecnologías del lenguaje*) and CIMAT Bajío Supercomputing Laboratory (#300832). Sanchez-Vega acknowledges CONACyT for its support through the Program "Investigadoras e Investigadores por México" by the project "Desarrollo de Inteligencia Artificial aplicada a la prevención de violencia y salud mental." (ID.11989, No. 1311).

References

- [1] W. H. Organization, Mental health, <https://www.who.int/health-topics/mental-health>, 2023. Accessed on June 28, 2023.
- [2] A. M. Mármol-Romero, A. Moreno-Muñoz, F. M. Plaza-del-Arco, M. D. Molina-González, M. T. Martín-Valdivia, L. A. Ureña-López, A. Montejo-Ráez, Overview of MentalriskES at IberLEF 2023: Early Detection of Mental Disorders Risk in Spanish, *Procesamiento del Lenguaje Natural* 71 (2023).
- [3] A. Rocha, W. J. Scheirer, C. W. Forstall, T. Cavalcante, A. Theophilo, B. Shen, A. Carvalho, E. Stamatatos, Authorship attribution for social media forensics, *IEEE Trans. Inf. Forensics Secur.* 12 (2017) 5–33. URL: <https://doi.org/10.1109/TIFS.2016.2603960>. doi:10.1109/TIFS.2016.2603960.
- [4] E. Stamatatos, N. Fakotakis, G. Kokkinakis, Computer-based authorship attribution without lexical measures, *Comput. Humanit.* 35 (2001) 193–214. URL: <https://doi.org/10.1023/A:1002681919510>. doi:10.1023/A:1002681919510.
- [5] J. Houvardas, E. Stamatatos, N-gram feature selection for authorship identification, in: J. Euzenat, J. Domingue (Eds.), *Artificial Intelligence: Methodology, Systems, and Applications*, 12th International Conference, AIMS 2006, Varna, Bulgaria, September 12–15, 2006, Proceedings, volume 4183 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 77–86. URL: https://doi.org/10.1007/11861461_10. doi:10.1007/11861461_10.
- [6] Q. Han, J. Guo, H. Schütze, Codex: Combining an SVM classifier and character n-gram language models for sentiment analysis on twitter text, in: M. T. Diab, T. Baldwin, M. Baroni (Eds.), *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2013*, Atlanta, Georgia, USA, June 14–15, 2013, The Association for Computational Linguistics, 2013, pp. 520–524. URL: <https://aclanthology.org/S13-2086/>.
- [7] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. *arXiv:1810.04805*.
- [8] G. Jawahar, B. Sagot, D. Seddah, What does BERT learn about the structure of language?, in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Florence, Italy, 2019, pp. 3651–3657. URL: <https://aclanthology.org/P19-1356>. doi:10.18653/v1/P19-1356.
- [9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, *J. Mach. Learn. Res.* 12 (2011) 2825–2830.
- [10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017. *arXiv:1706.03762*.
- [11] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, GLUE: A multi-task benchmark and analysis platform for natural language understanding, in: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 353–355. URL: <https://aclanthology.org/W18-5446>. doi:10.18653/v1/W18-5446.
- [12] H. F. Inc., Hugging face - the ai community building the future, <https://huggingface.co/>, 2021.

- [13] J. M. Pérez, D. A. Furman, L. Alonso Alemany, F. M. Luque, RoBERTuito: a pre-trained language model for social media text in Spanish, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 7235–7243. URL: <https://aclanthology.org/2022.lrec-1.785>.
- [14] J. M. Pérez, J. C. Giudici, F. Luque, pysentimiento: A python toolkit for sentiment analysis and socialnlp tasks, 2021. [arXiv:2106.09462](https://arxiv.org/abs/2106.09462).
- [15] M. García-Vega, M. Díaz-Galiano, M. García-Cumbreras, F. Del Arco, A. Montejo-Ráez, S. Jiménez-Zafra, E. Martínez Cámara, C. Aguilar, M. Cabezudo, L. Chiruzzo, et al., Overview of tass 2020: Introducing emotion detection, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020) Co-Located with 36th Conference of the Spanish Society for Natural Language Processing (SEPLN 2020), Málaga, Spain, 2020, pp. 163–170.
- [16] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, 2019. [arXiv:1912.01703](https://arxiv.org/abs/1912.01703).
- [17] B. Courty, V. Schmidt, Goyal-Kamal, MarionCoutarel, B. Feld, J. Lecourt, SabAmine, kn-goyal, M. Léval, A. Cruveiller, S. Luccioni, ouminasara, F. Zhao, A. Joshi, A. Bogroff, H. de Lavoreille, N. Laskaris, LiamConnell, A. Saboni, D. Blank, Z. Wang, A. Catovic, inimaz, M. Stęchły, alencon, JPW, MinervaBooks, SangamSwadiK, H. M., brotherwolf, mlco2/codecarbon: v2.2.4, 2023. URL: <https://doi.org/10.5281/zenodo.8063401>. doi:10.5281/zenodo.8063401.