Early Risk Detection for Mental Health Disorders: UnibucAI at MentalRiskES 2024

Cristian Daniel Păduraru[1], Ion Marian Anghelina[1],*

[1]University of Bucharest, 14 Academiei St, Bucharest, 010014, Romania

Abstract

As the number of mental health disorders has risen in the recent years, so has the interest in detecting the signs of these disorders as early as possible, which is also the topic of the MentalRiskES shared task. This paper presents our team's solutions to the three proposed tasks in the 2024 edition of MentalRiskES. By relying on deep pretrained encoders, task specific data augmentations and an optimization strategy from the literature of subpopulation shifts, we obtained the best results in terms of Macro_F1 score in tasks 2 and 3 and competitive ones in the first task.

Keywords

Early Risk Detection, Data Augmentations, GroupDRO, LSTM

1. Introduction

Mental disorders have become rather common in the recent years [1] and global events, such as the COVID-19 pandemic, have lead to an increase in the demand for mental health [2]. People suffering from certain disorders are also at an increased risk of suicide, which is among the leading causes of death for people aged 15-29 [1, 3]. As effective prevention and treatment options exist [1], there is also a growing interest in detecting these disorders (or signs of developing them) as early as possible, the most common type of data used in this regard being a person's activity on social media.

Since 2017, as part of the Conference and Labs of the Evaluation Forum (CLEF) [4], the task of Early Risk Prediction on the Internet (eRisk) [5] has focused on the early detection of diverse mental health conditions from social media posts of people. The MentalRiskES (MRES) workshop, which started in 2023 [6] and is part of the IberLEF [7] evaluation campaign, targets similar problems, but focuses on texts that are written in Spanish rather than English.

This paper presents the solutions of our team, UnibucAI, for the tasks of the 2024 edition of MRES [8]. The rest of the article is structures as follows: Section 2 contains a description of the three tasks that were proposed, the data provided by the organizers for each one of them and the evaluation procedure for the systems developed by the participants. Section 3 describes the systems that we used to make submissions (pretrained networks, trained classifiers and training procedures, hyperparameter and model selection criteria). The best results that we have obtained in each task are presented in Section 4, along with the results of other participating teams and baselines provided by the organizers. Finally, we talk in Section 5 about the conclusions of our work for these tasks. Our implementation of the described methods will be published at the following link.

2. Tasks

The 2024 edition of MRES [8] comprises three classification tasks, related to the early detection of anxiety, depression and suicidal ideation from sequences of messages sent on Telegram groups. Predictions are judged based on classification performance (using metrics such as F_1 score, precision etc.), latency in

IberLEF 2024, September 2024, Valladolid, Spain

*Corresponding author.

cristian.paduraru@s.unibuc.ro (C. D. Păduraru); ion.anghelina@s.unibuc.ro (I. M. Anghelina)

Table 1
Class label distribution for Task1 data.

Class	Train	Trial
none	213	10
anxiety	88	5
depression	164	5
Total	465	20

Table 2
Number of positive occurrences for each context in the train and trial sets for Task2

Context	Train	Trial
addiction	12	0
emergency	17	0
family	61	0
work	17	0
social	88	3
other	56	3
none	74	4

detecting the potential mental health disorders (ERDE5, ERDE30) as well as the computational burden of the system (carbon emission, energy consumption, necessary memory).

2.1. Task1

The first task is a multiclass classification problem where subjects have to be labeled with *depression*, *anxiety* or *none*, depending on the symptoms (or lack thereof) that they are showing through their activity. The organizers have also noted that certain individuals may show signs of both depression an anxiety, but one of them is more dominant than the other and thus gives the final label.

Data For this task the organizers have provided a train dataset which consists of sequences of messages from 465 different subjects and an additional trial set with 20 more users. We present in Table 1 the distribution of class labels for the two sets. These messages also come with an additional metadata - the timestamp at which they have been sent.

2.2. Task2

In the second task participants have to determine the context in which a subject labeled as suffering from anxiety or depression has developed said condition. This is modeled as a multilabel problem, with possible contexts being *addiction*, *emergency*, *family*, *work*, *social*, *other* or *none* if there is no specific context detected.

Data As this task is tied to the previous one, the same train and trial samples as before were provided, but with additional context labels for those subjects that were marked as suffering from depression or anxiety. As it can be noted from the distribution of labels in Table 2, there are few positive examples for each context and the instances are also not mutually exclusive (for certain subjects there are multiple factors which have contributed to the development of their disorder).

2.3. Task3

Task3 is a binary classification problem where individuals having *suicidal* thoughts have to be identified. For this task no training data was provided, but the test data is known to be in the same format as that of Task1.

2.4. Evaluation

The evaluation of the solutions to these tasks is done in an online fashion, based on rounds. At each round, the participants receive a new message from each subject and must submit their predictions before they can move on to the next round. This makes the task more difficult as the current message of a person may as well be his last one, which does not allow the participants to be patient in their decision and first gather all the data before giving a prediction based on all the messages. Also, the provided train and trial data is labeled at the level of sequences and not at message level, so it is unknown from which timestep onwards a subject should be confidently classified as suffering from one of the possible disorders. For tasks 1 and 3 only the first positive prediction is taken into account (e.g. if in Task1 a user is labeled as suffering from depression at a certain round then any further predictions will be ignored) and in task 2 only the contexts predicted at the first positive prediction of Task1 (anxiety or depression) are considered. The test data for each round was received from an HTTPS server through GET requests and predictions had to be submitted by POST requests to the server. A full description of the training corpus can be found at [9].

Besides the common classification metrics (accuracy, F1 etc.) the early risk detection error (ERDE) is also used to characterise how timely the solutions to tasks 1 and 3 can detect the signs of mental disorders. Another aspect that the organizers wished to evaluate was the computational efficiency of the proposed solutions. Participants were thus asked to measure the total RAM needed to run their solution, the CPU usage, number of FLOPS, processing time and carbon emissions with the help of the Code Carbon [10] tool.

3. Method

As the number of samples provided in tasks 1 and 2 seemed rather low, we decided that data augmentation would be necessary in order to reduce the risk of overfitting on the training data. As it has been noted in the previous section, the data is also imbalanced, which should be a common occurrence in the medical field where many conditions are rare compared to the size of the entire population. This imbalance poses a problem for classifiers trained with simple Empirical Risk Minimization (ERM), as they can perform poorly on underrepresented classes or subgroups of a class in exchange for a high overall accuracy [11]. Considering the specifics of the tasks, such a classifier is not desirable as the minority groups are more critical to detect (especially in the case of Task3, where a timely detection could potentially save the life of a person).

Some common approaches in dealing with this imbalance are using a balanced subset of the data [12, 13] in the training of the classifier or weighting the loss of samples based on the size of the class that they are a part of [12]. As the number of samples is already quite low, we did not consider following the first approach, while the second one does not take into account the fact that samples in a certain class may be, in general, harder to learn than the others. We have made this assumption as we did not fine-tune the encoders on the provided data, so features which are meaningful for the classification task may not all be extracted, even if the encoder was fine-tuned on a similar task, due to the specifics of each dataset (distribution shifts).

For tasks 1 and 3 we thus opted to train our classifiers by Group Distributionally Robust Optimization (GroupDRO) [14], using the class labels also as group labels and following the implementation of the algorithm from [11]. GroupDRO assigns for each group label g a weight q_g that is uniformly initialized and updated during the training process. Formally, let (X, Y, G) be a batch of training samples and their corresponding class and group labels, $\mathcal{Y}$, $\mathcal{G}$ the sets of all class and group labels in the dataset and $q^{(t)}$ the vector of group weights at time t ($q_g^{(0)} = 1/|\mathcal{G}|$ for each $g \in \mathcal{G}$). We denote by $S_g = \{(x_i, y_i) | (x_i, y_i, g_i) \in (X, Y, G),\ g_i = g\}$ the subset of samples with group label g and by $\mathcal{L}(f_\theta, S)$ the loss of a classifier f_θ on a set S of training examples. At each timestamp t we update the group weights as follows:

$$q'_g = q_g^{(t-1)} \exp(\eta \mathcal{L}(f_\theta, S_g))$$

$$q_g^{(t)} = q'_g \,/ \sum_{g' \in \mathcal{G}} q'_{g'}$$

, where η is a hyperparameter of the algorithm, which we set to 0.1 in all our experiments. The loss $\mathcal{L}_{GDRO}$ used in optimizing the parameters θ of the classifier is then computed as:

$$\mathcal{L}_{GDRO}(f_\theta, (X, Y)) = \sum_{g \in \mathcal{G}} q_g^{(t)} \mathcal{L}(f_\theta, S_g)$$

In our case we have that $\mathcal{Y} = \mathcal{G}$ and $y_i = g_i$ for each sample in the training set. This formulation of the loss prevents the classifier from disregarding any of the classes by adapting the weights to favor the learning of classes on which the performance is poor. The fact that this is an online algorithm also allows for slight adaptation of the weights for each batch of samples (if a class has a low average loss but it contains some hard examples - not outliers or misclassified examples - then those samples will receive a higher weight than other samples from the same class).

Defining the notion of group for the second task, a multilabel classification, is more difficult, unless we interpret it as separate binary classifications tasks, one for each context. In this case we chose to simply train the classifiers with ERM, fixed class weights and data augmentations.

In all our solutions we used deep encoders, pretrained on Spanish texts, and trained a classifier on top of the features extracted by these encoders from individual messages of a user. We present in the following subsections all the task specific details of training the classifiers used in our submissions.

3.1. Preprocessing

Since our text processing method is based on pretrained transformer models, removal of affixes, accents or punctuation would only lead to information loss and a poorer performance. Considering the fact that the source of the samples is known, our goal was finding a series of preprocessing methods which would increase the language norms strictness and information density of text messages.

The first irregularity we had to deal with, was the presence of multiple *emoji*, both as *Unicode* characters, and *ASCII* descriptions. While the former category of *emoji* were dealt with relatively easily using the support of Python's *emoji* library for Spanish language, the latter category was harder to tackle, since it involved manual pattern matching. For this task, we analyzed 18 of the most common *emoji* in Spanish text , such as *":)"* for *"cara sonriente"*, and manually translated them to Spanish. The next step consisted of eliminating repeating substrings in words, such as multiple adjacent vowels or the repetition of the "laughing" formulations: *"jaja"* or *"jsjs"* of arbitrary lengths. As a final step, after manually analyzing data, we observed the corruption of multiple *"o"* characters, them being replaced with the sequence *" @"*. This decoding error was manually solved.

3.2. Task1

The text encoder that we have used in the first task is a RoBERTuito [15] model that was fine-tuned on the TASS 2020 [16] corpus and is available on HuggingFace under the name of *pysentimiento/robertuito-sentiment-analysis*. As for the classifiers, we trained single layer LSTMs [17] by processing the embeddings of messages and applying a linear layer only on the last hidden state from each sequence, as it is unknown where the source of the label for the anxiety and depression classes lies in the sequence.

Based on the organizers' observation that certain individuals may show signs of both depression and anxiety, we have decided to use soft labels for these two classes, setting the values to 0.9 for the annotated class and 0.1 for the other one.

While in general we have only sequence level annotation, for the subjects that are not suffering from any disorder we know that they must to predicted as such at each round. Adding to the training set every possible subsequence of consecutive messages for these individuals would greatly reduce the expected number of positive sample in each batch during training, so instead we decided to randomly pick these subsequences. At each epoch and for each subject labeled with *none* we uniformly pick a

Table 3
Hyperparameter values over which we have performed a grid search in Task1.

Hyperparameter	Values
Batch size	{32, 64, 96}
Hidden size	{64, 96, 128, 160}

Table 4
Hyperparameters used in the training of LSTM classifiers for our submissions in Task1.

Run	Preprocessing	Batch size	Hidden size
0	No	96	96
1	No	64	128
2	Yes	32	128

random number n between 1 and the total number of messages for the current subject and use only his first n messages to make the prediction. We thus enforce correct predictions earlier in the sequences without changing the ratio of class samples seen in an epoch by the classifier. Meanwhile, for a positive subject we randomly introduce up to 10 messages from a negative subject at the beginning of his sequence of messages, with a probability of 30%. We had observed that certain sequences started off with the person explicitly saying that he is dealing with depression or anxiety. This augmentation does not affect the class of a given subject and is meant to ensure that the classifier is not biased towards the first few message in the sequence.

We trained the classifiers with the Adam optimizer, a learning rate of 1e-3 and cross entropy loss for 100 epochs, saving the checkpoint with the best Macro_F1 scores on a validation set that was i.i.d. sampled from the combined train and trial sets. We performed a grid search over multiple values for the batch size and hidden state size (see Table 3 for the complete set of values). We also experimented training with and without the data preprocessing previously mentioned. In the end, we selected the hyperparameter combinations that yielded the best Macro_F1 scores on validation. The explicit combinations are presented in Table 4, together with the index of the run that they represent in the official results. For more technical details about Task1 solutions and other unsuccessful approaches, please refer to Appendix A.

3.3. Task2

For this task, the chosen model for encoding the texts is based on the BERT model pretrained on a Spanish language corpus, BETO [18]. The version used is finetuned on the IMDB Movie Review Spanish corpus [19] and is available on HuggingFace under the name of *ignacio-ave/beto-sentiment-analysis-spanish.*

Following the extraction of the text embeddings, the main classifier model consists of one unidirectional LSTM layer, followed by a Fully Connected dense layer applied to the last hidden state of the LSTM.

For this task, each of the seven context labels, namely *addiction, emergency, family, work, social, other* and *none*, was labeled independently, as one subject's disease might be caused by multiple contexts. In this case, hard labelling was used for each of the 7 dimensions, 0 denoting the lack of causality between the respective context and the disease, and 1 the causality.

For enhancing the number of samples on which our model is trained, we performed data augmentation by concatenating random different input samples over their temporal dimension. The concatenation augmentation was done for each particular sample with a probability of $p < 1$, so that a part of the initial samples remain unchanged.

Namely, if a contexts influences at least one of the original samples, its label will be set to 1, with the exception of the *none* context label, which will only be positive if both the constituent samples have no context associated.

For training, we used the *BCEWithLogitsLoss* loss function. Since there exists a heavy imbalance between the number of positive samples for each context, visible in table 2, we used a weighting

Table 5
Hyperparameter values over which we have performed a grid search in Task2.

Hyperparameter	Values
Learning Rate	{1e-4, 5e-3, 1e-3}
Hidden size	{64, 96, 128}

Table 6
Hyperparameters used in the training of LSTM classifiers for our submissions in Task2.

Run	Learning Rate	Hidden size	No. Classifiers
0	1e-4	96	7
1	3e-4	128	7
2	1e-4	96	1

technique, assigning each positive sample a weight inversely proportional to its frequency in the training set. The optimizer used was *Adam*, with a learning rate of 1e-4 for a number of 100 epochs, keeping the best intermediary results in terms of validation loss. Multiple values of the hyperparameters, including hidden size for the LSTM layer and the learning rate were experimented with, all of them being documented in table 5.

For our final submissions, we chose two main approaches:

- Using an independent model for each context, thus maximizing the $F1$ score for each class independently.
- Using a single model for all contexts, thus maximizing the average $F1$ score over all classes.

The aim of the former approach was scoring higher on the accuracy metrics, while the purpose of the latter was to provide a reasonably accurate answer by consuming a lower amount of energy, and, thus, lower carbon emissions. More details about the submission can be found in table 6.

3.4. Task3

As we did not have any explicit training data for the third task, we looked for available datasets that addressed the same task of suicidal ideation detection in Spanish texts. We selected the dataset from [20] and the one from the 2023 SomosNLP Hackathon [21], which we considered to be of better quality, compared to those that were scraped from Reddit and contained false positive examples. These datasets had only individual text messages instead of sequences from the same person. While we could have created synthetic ones by randomly adding a positive example from these datasets in the sequences from Task1 and then train a classifier for sequential data as before, we opted for the simpler solution of predicting only based on the latest message of a subject. We considered that with this type of synthetic data the classifier would learn to detect abrupt changes in the topic of consecutive messages or differences in the style of writing between messages from the provided dataset in Task1 and those from the other datasets. On the same note, we always applied the data preprocessing steps described in the previous section to messages from all sources in order to reduce the differences in the style of writing. From the two external datasets we only took the positive examples and for the negative ones we reused the messages from the first two tasks. In our first solution we only took the messages of individuals that were not suffering from depression or anxiety and in the other ones we included all the messages from Task1 as negative examples. We tried this second solutions as the first classifier that we have trained detected many individuals from Task1 as potentially having suicidal thoughts, but we expected such cases to be a lot less frequent.

We used the same encoder as in Task1 to obtain embeddings for individual messages and then trained a linear layer over them using the Adam optimizer and the GroupDRO loss formulation for 50 epochs. We performed a grid search (see Table 7) for other training hyperparameters and selected the ones that lead to the best Macro_F1 score on a validation set, i.i.d. sampled from the set of all messages that we

Table 7
Hyperparameter values choices over which we performed a grid search for Task3.

Hyperparameter	Values
Batch size	{32, 64, 128}
Learning rate	{1e-3, 5e-3, 5e-2}

Table 8
Hyperparameter choices for the classifier used in the 3 submission to Task3

Run	Batch size	Learning rate
0	128	5e-3
1	32	1e-3
2	64	1e-3

Table 9
Task1 results of the top 3 participating teams, ordered by the Macro_F1 score of their best run. We also added the best baseline results provided by the organizers. **LatencyTP** is the median round number at which a true positive predictions is made. The best results for each metric are marked in bold.

Rank	**Team**	**Run**	**Accuracy**	**Macro_F1**	**ERDE5**	**ERDE30**	**LatencyTP**
1	ELiRF-UPV	2	**0.890**	**0.874**	0.405	0.045	8
3	BaseLine - Roberta Base [8]	2	0.853	0.834	0.162	**0.042**	3
5	UnibucAI	0	0.828	0.808	0.308	0.078	5
8	UNED-GELP	0	0.797	0.785	**0.138**	0.065	2

have used. When we used all the messages from Task1 as negative examples (runs 1 and 2) we noticed that all the classifiers had very similar Macro_F1 score, regardless of the hyperparameters, so we chose for run 1 the ones that lead to the best Macro_Precision, while for run 2 we picked the ones for the highest Macro_Recall. The explicit hyperparameters chosen for each run are in Table 8.

4. Results and Discussion

4.1. Task1

Table 9 contains the best results obtained on Task1 for the top three participants. Our first submission places us second in the ranking of teams and is overall the fifth best submission. The best solution in terms of Macro_F1 came from the team ELiRF-UPV, while the team UNED-GELP obtained the best ERDE5 score among all the participants. Our submission has a slightly better Macro_F1 than that of UNED-GELP, but at the cost of a higher ERDE5. This signifies that our solution is making more accurate predictions, but some are done rather late in the sequence of messages.

In figure 1 we present the confusion matrix of our best submission on the test set. As it can be noticed, the *depression* class has many false positives, from both of the other classes. This might indicate a possible spurious correlation that the classifier has learned from the training data or a bias towards the *depression* class.

4.2. Task2

In the second task, our team's second submission obtained the best Macro_F1 score (Table 10), followed by the solutions of UMUTeam, ELiRF-UPV and a baseline of the organizers. We also obtained the highest Macro_Recall, which is substantially higher than the score obtained by the baseline of the organizers and that of team ELiRF-UPV. On the downside, our solution has worse Macro_Precision and Accuracy scores compared to these two.

Figure 1: Confusion matrix on the test set of our best submission in Task1.

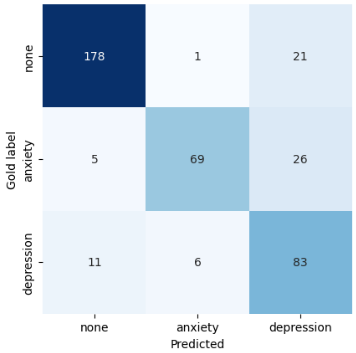

Table 10
Task2 results of the top 3 participating teams, ordered by the Macro_F1 score of their best run on the context detection task. The best results for each metric are marked in bold.

Rank	Team	Run	Accuracy	Macro_P	Macro_R	Macro_F1
1	UnibucAI	1	0.022	0.194	**0.508**	**0.268**
3	UMUTeam	0	0.007	0.166	0.408	0.224
5	ELiRF-UPV	0	0.065	0.262	0.177	0.208
6	BaseLine - Roberta Base [8]	2	0.075	**0.358**	0.168	0.181

Table 11
Task3 results of the top 3 participating teams, ordered by the Macro_F1 score of their best run. The best results for each metric are marked in bold.

Rank	Team	Run	Accuracy	Macro_P	Macro_R	Macro_F1	ERDE5	ERDE30
1	UnibucAI	0	0.655	**0.556**	**0.539**	**0.534**	0.511	0.238
4	UNED-GELP	0	0.618	0.465	0.480	0.456	0.326	0.215
5	Baseline (all positives)	1	**0.691**	0.345	0.500	0.409	**0.226**	**0.214**
6	V team	0	**0.691**	0.345	0.500	0.409	0.261	**0.214**
11	Baseline (all negatives)	0	0.309	0.155	0.500	0.236	0.691	0.691

4.3. Task3

Our first submission in Task3 has scored the highest Macro_F1, Macro_Precision and Macro_Recall scores of all participating teams (see Table 11). For this task the organizers provided only two baselines that had all predictions as either positive or negative. Team UNED_GELP placed second in terms of Macro_F1 score, but their solution has better ERDE5 and ERDE30 scores than ours.

While our results in this task are good, we acknowledge the fact that the use of a model which does not take into account past messages is prone to many false negatives and false positives due to a lack of context. For example, a person might cite the words of someone else dealing with suicidal thoughts, but if this information is not captured in the same message then the person could be detected as a false positive. Similarly, a person might show his thoughts by referencing past messages (or replying to the messages of other people), but these situations would go unnoticed by our classifier. Using a classifier that can process sequences of messages should thus be the preferred solution, but a proper validation procedure is also necessary, as the results on a synthetic dataset may not be representative for the performance of the classifier on real world data.

4.4. Practical considerations

In our submissions we did not specifically optimize for the efficiency metrics at inference time (emissions, processing time etc.). As per the remarks of the organizers, systems that could run on mobile devices or personal computers, with a low carbon footprint, would be of great interest for practical applications. Analyzing our system, it is obvious that most of the computational burden is caused by the encoder, so distilling these deep models into ones with fewer parameters should be the starting point in optimizing the systems for inference. The only problem that this poses is that it would require a lot more resources at training time.

Another thing to consider is the calibration of the classifiers, which we did not cover in this work. Applying any calibration technique, such as the Platt temperature scaling [22], would be important in order to regulate the confidence of classifiers before using them in a practical application.

5. Conclusions

In this edition of MentalRiskES we have relied on pretrained encoders to extract deep features from texts and trained simple classifiers with ERM and GroupDRO [14], a method stemming from the literature of subpopulation shifts, obtaining the best results in terms of Macro_F1 score in tasks 2 and 3 and competitive ones in the first task. Data augmentations where also essential in obtaining these results, but we have noticed that slight changes in hyperparameters can have a large impact on the final results, which is why a good validation set is needed in order to select proper values. Preprocessing the texts on the other hand had less of an impact and if one were to fine tune the pretrained encoders the right choice might be not to apply them, so that the network can learn to extract meaningful features from the style of writing (emojis, phrasing or slang).

On the downside, our solutions do not perform as well in terms of early detection for tasks 1 and 3. We consider that improving the latency in giving the right responses is of major importance for the underlying motivation behind the tasks, but such an objective is harder to track in the absence of more fine-grained labeling.

References

[1] World Health Organization, Mental disorders, 2022. https://www.who.int/news-room/fact-sheets/detail/mental-disorders [Accessed: 19/05/2024].

[2] World Health Organization, Covid-19 disrupting mental health services in most countries, who survey, 2020. https://www.who.int/news/item/05-10-2020-covid-19-disrupting-mental-health-services-in-most-countries-who-survey [Accessed: 19/05/2024].

[3] World Health Organization and others, Suicide worldwide in 2019: global health estimates (2021).

[4] A. Arampatzis, E. Kanoulas, T. Tsikrika, S. Vrochidis, A. Giachanou, D. Li, M. Aliannejadi, M. Vlachos, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction - 14th International Conference of the CLEF Association, CLEF 2023, Thessaloniki, Greece, September 18-21, 2023, Proceedings, volume 14163 of *Lecture Notes in Computer Science*, Springer, 2023. URL: https://doi.org/10.1007/978-3-031-42448-9. doi:10.1007/978-3-031-42448-9.

[5] J. Parapar, P. Martín-Rodilla, D. E. Losada, F. Crestani, Overview of eRisk 2023: Early risk prediction on the internet, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2023, pp. 294–315.

[6] A. M. Mármol-Romero, A. Moreno-Muñoz, F. M. P. del Arco, M. D. Molina-González, M. T. Martín-Valdivia, L. A. Ureña-López, A. Montejo-Raéz, Overview of MentalRiskES at IberLEF 2023: Early Detection of Mental Disorders Risk in Spanish, Procesamiento del Lenguaje Natural 71 (2023) 329–350. URL: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6564.

[7] Chiruzzo, L. and Jiménez-Zafra, S. M. and Rangel, F., In Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2024), co-located with the 40th Conference of the Spanish Society for Natural Language Processing (SEPLN 2024), CEUR-WS.org (2024).

[8] A. M. Mármol-Romero, A. Moreno-Muñoz, F. M. P. del Arco, M. D. Molina-González, M.-T. Martín-Valdivia, L. A. Ureña-López, A. Montejo-Ráez, Overview of MentalRiskES at IberLEF 2024: Early Detection of Mental Disorders Risk in Spanish, Procesamiento del Lenguaje Natural 73 (2024).

[9] 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, MentalRiskES: A new corpus for early detection of mental disorders in Spanish, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 11204–11214. URL: https://aclanthology.org/2024.lrec-main.978.

[10] K. Lottick, S. Susai, S. A. Friedler, J. P. Wilson, Energy Usage Reports: Environmental awareness as part of algorithmic accountability, 2019. arXiv:1911.08354.

[11] Y. Yang, H. Zhang, D. Katabi, M. Ghassemi, Change is hard: A closer look at subpopulation shift, arXiv preprint arXiv:2302.12254 (2023).

[12] B. Y. Idrissi, M. Arjovsky, M. Pezeshki, D. Lopez-Paz, Simple data balancing achieves competitive worst-group-accuracy, 2022. arXiv:2110.14503.

[13] P. Kirichenko, P. Izmailov, A. G. Wilson, Last Layer Re-Training is Sufficient for Robustness to Spurious Correlations, arXiv preprint arXiv:2204.02937 (2022).

[14] S. Sagawa, P. W. Koh, T. B. Hashimoto, P. Liang, Distributionally Robust Neural Networks for Group Shifts: On the Importance of Regularization for Worst-Case Generalization, 2020. arXiv:1911.08731.

[15] 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.

[16] 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.

[17] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.

[18] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained bert model and evaluation data, arXiv preprint arXiv:2308.02976 (2023).

[19] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 142–150. URL: http://www.aclweb.org/anthology/P11-1015.

[20] J. V.-R. y Sara Lana-Serrano y Eugenio Martínez-Cámara y José Carlos González-Cristóbal, TASS - Workshop on Sentiment Analysis at SEPLN, Procesamiento del Lenguaje Natural 50 (2013) 37–44. URL: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/4657.

[21] SomosNLP, 20203 hackathon - sucide comments dataset, 2023. https://huggingface.co/datasets/somosnlp-hackathon-2023/suicide-comments-es [Accessed: 18/05/2024].

[22] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger, On calibration of modern neural networks, 2017. arXiv:1706.04599.

[23] J. D. la Rosa, E. G. Ponferrada, M. Romero, P. Villegas, P. G. de Prado Salas, M. Grandury, Bertin: Efficient pre-training of a spanish language model using perplexity sampling, Procesamiento del Lenguaje Natural 68 (2022). URL: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6403.

A. Task1

The following encoders from HuggingFace were considered for this task:

- *ignacio-ave/beto-sentiment-analysis-spanish*
- *lxyuan/distilbert-base-multilingual-cased-sentiments-student*

- *pysentimiento/robertuito-sentiment-analysis*
- *edumunozsala/bertin_base_sentiment_analysis_es* [23]

In order to select the best one we have done a simple test where we averaged the embeddings of all messages for each subject and then trained a linear layer on top of these averaged embeddings with ERM and fixed class weights. The encoder that lead to the best validation Macro_F1 score, *pysentimiento/robertuito-sentiment-analysis*, was selected.

Regarding the soft labels probability distribution, we had also tested on a single set of hyperparameters two other options, giving the correct class (anxiety or depression) only a probability of 0.8 or 0.7 (the complement was assigned to the other disorder), but observed a significant decrease in Macro_F1 score. While signs of both disorders (anxiety and depression) may be present for certain individuals (as the organizers have mentioned), it seems that trying to capture this in the general loss of samples does not lead to stable results. In cases where an individual does not show signs of the other disorder, these soft labels may induce a bad optimization target.

In order to reduce the number of false negatives and false positives we have also attempted to add to the training set some hard examples. We first trained a classifier with the mentioned procedure, performed predictions on each sequence from the training set at every time step and then added all messages in the sequence, up to the point of a mistake, to the training set, with the correct label. Unfortunately, this approach has actually lead to a decrease in the performance on the validation set so we did not investigate any further on this technique.