Text Classification For Early Detection of Eating Disorders and Depression in Spanish

Pablo Turón[*,†], David Cabestany[†], Naiara Pérez and Montse Cuadros

Vicomtech, Member of BRTA, Mikeletegi Pasealekua, 57, 20009 Donostia-San Sebastián (Spain)

Abstract

This paper presents the participation of the Vicomtech NLP team in the MentalRiskES shared task about the early detection of mental disorders in Spanish comments from Telegram users. We participate in two tasks: Task 1a, related to eating disorders, and Task 2a related to depression. For both tasks we propose a set of approaches based on supervised text classifiers using Transformers. We prioritise our experimentation in building low resource demand systems with the minimum low carbon footprint. With those highlighted features, our systems are developed to detect disorders as early as possible, involving an initial phase that automatically projects stream labels at message level, since not all messages contained in a stream are equally representative of the stream class. We obtain the best ERDE_5 result in depression detection (0.27) and second-best in eating disorders (0.17).

Keywords

BERT, Mental Disorders, Disorders Detection, Early Risk

1. Introduction

The rising prevalence of mental health disorders, including eating disorders, dysthymia, anxiety, depression, and suicidal ideation, has become a significant global concern in recent years. For example, the global incidence of eating disorders was estimated to affect approximately 14 million individuals between 1990 and 2019—with nearly 3 million being children and adolescents—[1]. In light of this, there has been a growing interest in utilising social media as a tool to detect mental health disorder signals among the general population, to better understand mental health trends and potentially facilitate early detection and intervention. However, these efforts have predominantly concentrated on the English language.

To address this gap, the MentalRiskES [2] challenge was introduced as part of the evaluation campaign IberLEF 2023. The primary objective of this challenge is the early detection of mental disorders in Spanish comments from Telegram users. The challenge was conducted online,

IberLEF 2023, September 2023, Jaén, Spain

[*] Corresponding author.

[†] These authors contributed equally.

pturon@vicomtech.org (P. Turón); dcabesma@gmail.com (D. Cabestany); nperez@vicomtech.org (N. Pérez); mcuadros@vicomtech.org (M. Cuadros)

https://www.vicomtech.org/es/vicomtech/equipo/1053 (P. Turón);
https://www.linkedin.com/in/david-cabestany/ (D. Cabestany);
https://www.vicomtech.org/es/vicomtech/equipo/551 (M. Cuadros)

0000-0002-5563-1120 (P. Turón); 0000-0002-0122-7931 (D. Cabestany); 0000-0001-8648-0428 (N. Pérez); 0000-0002-3620-1053 (M. Cuadros)

requiring participants to identify potential risks as early as possible within a continuous stream of data. That is, participants were presented with user messages sequentially and required to emit a prediction at each step, thus emulating the dynamics of real-time analysis. Accordingly, systems were evaluated both for their risk detection correctness and speed.

This paper describes the approach developed by the Vicomtech NLP team to address two tasks of the MentalRiskES challenge:

- **Binary classification of eating disorders (Task 1a)**: the goal of this task was to detect as soon as possible whether users suffer from anorexia or bulimia.
- **Binary classification of depression (Task 2a)**: the goal of this task was to detect as soon as possible whether users suffer from depression.

Our team participated in these tasks with supervised Transformer-based [3] text classifiers, with which we obtained the best ERDE_5 results in depression detection (0.27) and second-best in eating disorders (0.17). The novelty of our proposal lies in the preprocessing of the training data, whereby we automatically labelled streams at message level. As will become clearer in subsequent sections, this preprocessing step was motivated by the observation that the training data came labelled at stream level, but not all messages within a stream are equally representative of the corresponding stream category.

The remainder of the paper is structured as follows: Section 2 provides a brief overview of the related work; Section 3 introduces the challenge data; Section 4 presents our approach to the tasks, including a detailed explanation of the aforementioned data preprocessing methodology; Section 5 reports and analyses the obtained results, both in the development phase and in the official evaluation; finally, Section 6 concludes the paper by summarising the key findings and suggesting avenues for future research.

2. Related work

Many efforts have been made to detect different mental disorders through written messages on the internet, based on a text classification problem [4, 5, 6, 7]. Initiatives like the Cross-Lingual Evaluation Forum (CLEF) have been promoting the Early-Risk Identification task by analysing social media posts [8] to detect different disorders in English since 2017. This motivated the MentalRiskES organisers to create, to the best of our knowledge, the first Early-Risk Identification task in the Spanish language.

The solutions that have been developed by the CLEF participants cover a wide range of different approaches. For example, the UNSL team at eRisk 2022 [9] developed a system that split the solution in two different problems: *i)* classifying partial information, and *ii)* deciding the moment of classification. For the development of their system, they exploited feature engineering schemas (bag of words, TF-iDF) to train both classic models (such as support vector machines, regression classifiers, etc.) and models based on Transformers [3]. Other solutions, like Bucur et al. [10], relied on massive data crawling to train a BERT [11] model and apply a high threshold when classifying.

A different solution, which inspired our work, was the on proposed by the UNED-NLP team [12] in the eRisk 2022 shared task. They proposed the use of Approximated Nearest Neighbours

(ANN) to tag the entire corpus to a message level. This process was performed adjusting text embeddings with ANN algorithms, where they start assuming that all the messages from a negative stream are negative, and all the messages from a positive stream are positive. Once they adjust their first model, they repeat the process iteratively until convergence, considering that a message is positive only if its 20 nearest neighbours are positive. Once the model has been adjusted, they would use the same criteria with the last ANN model to consider a message as positive.

3. Data

The MentalRiskES dataset is a novel collection of content sourced from public groups on the messaging platform Telegram. Each training example consists of a sequence of messages or *stream* from one Telegram user, and the corresponding mental disorder risk label. Specifically, the streams were labelled as SUFFER or CONTROL by 10 annotators, SUFFER being the positive class; the final label for each stream is SUFFER if 5 or more annotators voted thus, and CONTROL otherwise. Streams vary in length, containing typically 20 to 40 messages, although many of them consist of up to 100 messages. We refer the reader to the challenge overview article [2] for further details about the dataset and its curation process.

The organisers provided 175 training examples and 10 trial examples for each task. Guided by early experiment results and data analysis, we preprocessed these datasets as follows:

1. **Resample** the available data in order to obtain two development sets, with 15% and 10% of the available examples each—one for development purposes proper (Dev_d), and the other for testing purposes (Dev_t), respectively.

2. **Filter out streams with 4, 5 or 6 votes** from the resulting training set, as such controversial examples could introduce noise in the learning process. Consequently, we discarded 18 and 27 examples from Task 1 and Task 2 data respectively.

3. **Detect and erase verbalised forms of emojis** from the messages. As part of the data curation process, some emojis had apparently been transformed to their verbalised form (see examples in Table 7, Appendix A), which rendered some messages ungrammatical or even nonsensical. This preprocessing step was only applied to depression data (Task 2), seeing as the problem seemed to occur more often in that case.

No additional data was used other than that provided by the organisers. Thus, the final example distribution is as shown in Table 1, which also includes the official test. All references to train and/or development data should be hereafter interpreted as referring to these samples.

As can be seen, MentalRiskES is challenging in many respects, namely, the reduced size of the dataset and its imbalanced class distribution, but also the subjectivity of the problem, as evidenced by the spread-out annotator vote distribution.

Table 1
Distribution of streams over category and annotator votes per task and split

		Task 1a: Eating disorders				**Task 2a: Depression**			
Category	**Votes**	**Train**	**Dev$_d$**	**Dev$_t$**	**Test**	**Train**	**Dev$_d$**	**Dev$_t$**	**Test**
SUFFER	10	39	8	7	46	30	3	5	35
(positive)	9	9	0	0	10	13	4	1	17
	8	3	0	0	3	15	3	1	6
	7	1	0	0	3	10	1	3	6
	6	-	1	0	2	-	3	0	3
	5	-	3	1	0	-	1	0	1
	Total	52	12	8	64	68	15	10	68
CONTROL	4	-	0	1	10	-	2	3	14
(negative)	3	10	1	1	20	11	4	2	19
	2	12	3	2	24	11	1	1	15
	1	22	6	3	25	11	3	0	18
	0	30	6	4	7	19	3	3	15
	Total	74	16	11	86	52	13	9	81

4. Methodology

While the provided challenge data comes annotated at stream level, early risk detection systems must be able to emit predictions at each step, that is, by having seen only partial stream sequences. Intuitively, then, a classifier trained with complete streams could have a hard time emitting correct predictions for partial streams, given that not all messages in a stream are bound to be equally representative of the stream category (e.g., some messages in positive streams could deal with topics unrelated to the target mental disorders).

In accordance with this premise, our approach to MentalRiskES has consisted of two distinct phases: first, automatically annotating streams at message level as SUFFER or CONTROL; second, exploiting the resulting silver corpus to train early detection systems. These phases are illustrated in Figure 1 and explained in detail in subsequent sections (4.1 and 4.2 respectively).

Both phases involve training BERT-like text classification models based on DiagTrast-Berto [13], a BETO [14] model post-trained on the synthetic corpus about mental disorders DiagTrast [15]. The architecture and hyperparameters are the same for both, as described in Section 4.3, the training setup differing only on the data used in each case.

4.1. Message-level annotation

The goal of this phase was to automatically detect the most representative messages of the SUFFER class in SUFFER training streams, and assign the CONTROL label to all other messages. That is, did not involve CONTROL streams, whose messages were directly assigned the CONTROL label. This phase corresponds to the leftmost box of Figure 1. In what follows, we introduce the key concepts of our proposal, namely, the average confidence Δ (ACΔ) and its threshold.

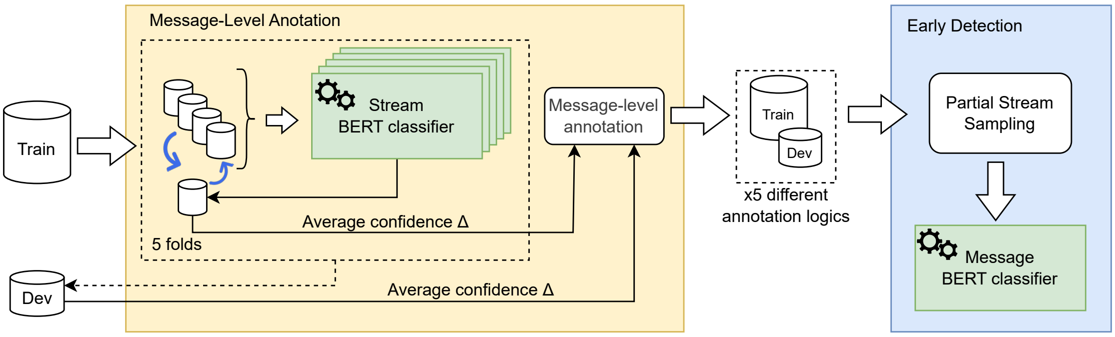

Figure 1: Methodology overview. Our approach consisted of 2 phases: first, automatic message-level annotation of the training and development data (Section 4.1); second, training of early risk detection models (Section 4.2).

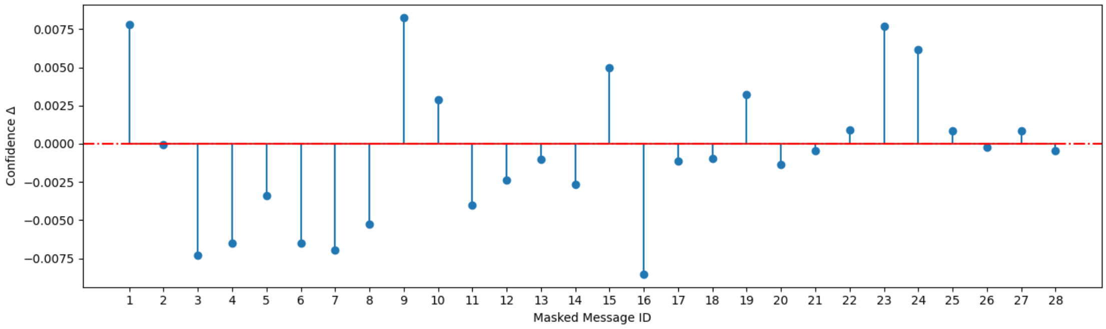

Figure 2: Confidence Δ scores by message in an eating disorders stream (see Appendix A)

4.1.1. Average confidence Δ (ACΔ)

Our proposal builds on the intuition that, given an early risk classification model and an input message stream, if the confidence of the model for the SUFFER category is markedly higher with a particular message in the stream than without it, then it is sound to assume that the message must be representative of the SUFFER class.

Here, **model confidence** is defined as the result of applying a sigmoid function to the output logits of a BERT-like binary classification model, so that a value of 1 corresponds to a 100% SUFFER stream, and 0 to a 100% CONTROL stream. The **confidence Δ of a message or messages** is then the difference between the confidence of the model for the entire stream and the confidence of the model for the stream without said message(s), whereby a positive confidence Δ would imply that the message(s) contributed towards the SUFFER class. That is, the confidence Δ of a message is always relative to the stream it occurs in.

Figure 2 illustrates the confidence Δ scores of each message in a real stream of Task 1, eating disorders. The peaking messages are as follows (translations to English are given below each message; the whole stream can be consulted in Appendix A):

(1) Ana Peso actual: 68 Peso meta: 50 Edad: 20

An[orexi]a Current weight: 68 Goal weight: 50 Age: 20

(9) Es posible vomitar las calorías?
Is it possible to vomit the calories?

(15) Tips para bajar de peso en 3 días por favor urgente
Tips to lose weight in 3 days please urgent

(19) Estoy tan gorda que hice reventar una falda Me siento muy mal
I'm so fat that I burst a skirt I feel so bad

(23) Sabes que no tienes hambre y simplemente comes porque te sientes estresad@
You know you are not hungry and you just eat because you feel stressed

(24) No puedes detenerte ni aunque pienses en qué vas a engordar y te verás horrible
You can't stop even if you think you're going to get fat and look awful

The confidence Δ scores of training set messages were computed in a k-fold validation fashion: the training dataset was divided into 5 folds with approximately the same number of streams; then, the messages of each held-out fold were processed with the confidence scores given by binary classification models trained on the other 4 folds. More specifically, we trained 5 models per fold with different seeds, for a total of 25 models, which allowed us to calculate an **average confidence** Δ (ACΔ) per message. The ACΔ of the development set messages (Dev$_d$) were obtained by averaging the scores of all the 25 models. Details about model architecture and training setup can be consulted in Section 4.3 and Appendix C.

4.1.2. ACΔ threshold

Given a stream of messages and their ACΔ scores, we annotated messages as SUFFER if their score surpassed a given threshold, and CONTROL otherwise. In this work, we designed and tested multiple types of thresholds:

Naive A message is considered SUFFER if its ACΔ is positive.

Unigram Local Maximum (LocMax$_1$) A message is considered SUFFER if its ACΔ is greater than 20% of the maximum ACΔ of the stream.

N-gram Local Maximum (LocMax$_n$) A message is considered SUFFER if LocMax$_1$ applies or if its dropping alongside n consecutive messages (2 or 3) yields an ACΔ that is greater than 22% of the maximum ACΔ of the stream, in which case all the involved messages are tagged as SUFFER.

Unigram Global Maximum (GloMax$_1$) A message is considered SUFFER if its ACΔ is greater than 20% of the average maximum ACΔ of all the streams.

N-gram Global Maximum (GloMax$_n$) A message is considered SUFFER if GloMax$_1$ applies or if its dropping alongside n consecutive messages (2 or 3) yields an ACΔ that is greater than 22% of the average maximum ACΔ of all the streams, in which case all the involved messages are tagged as SUFFER.

The ACΔ ratios (20% and 22%) we set empirically, along with a general minimum threshold of 0.004 (except for the naive approach).

Statistics about the resulting silver corpora (one per threshold strategy) can be found in Table 8 of Appendix B. Overall, less than 5% of the messages in each stream were labelled as SUFFER by the different annotation strategies, the first message labelled as SUFFER being on average in the third or fourth position. Appendix A includes a complete example of a stream annotated at message level.

4.2. Early risk detection

In the second phase of the experimentation, we trained binary classification models that are able to recognise SUFFER cases of the target disorder seeing as few messages as possible. To that end, we sampled partial stream training and development sequences from SUFFER and CONTROL streams, leveraging our message-level annotations, in order to emulate the actual inference scenario of the challenge. This phase is represented in the rightmost box of Figure 1.

4.2.1. Partial stream sampling (PSS)

The goal of partial stream sampling (PSS) is to obtain sub-sequences or *partial streams* of each stream, with the label for a sub-sequence being SUFFER if any of the messages in it is SUFFER, and CONTROL otherwise. The process differs depending on the type of stream being sampled.

On the one hand, **CONTROL streams** only yield **CONTROL samples**. In this case, we extracted samples semi-randomly, assigning more likelihood to shorter stream lengths. In 10% and 5% of the cases, we sampled 2 and 3 sub-sequences respectively instead of just one.

On the other hand, **SUFFER streams** yield both SUFFER and CONTROL samples:

- **SUFFER samples** extend from the beginning of the stream up to and including a SUFFER message. We extracted one sample per stream from the beginning of the stream up to the *first* SUFFER message. Additionally, we generated samples that extend up to the second or third SUFFER message in 20% of the streams, so as to maximise classifier recall.

- **CONTROL samples** extend up to but excluding the first SUFFER message. These samples are meant to maximise classifier precision and were generated from 10% of the SUFFER streams.

This process was applied once per stream, obtained what we will henceforth refer to as **"1-time sample" or 1s** datasets. By definition, 1s datasets only include content from the start of each stream. Furthermore, we applied the sampling method iteratively in a sliding-windows fashion (**"exhaustive sampling" or $\forall$s**), effectively obtaining more training and development samples.

These two sampling techniques, paired with the five message-level annotation versions (see Section 4.1) produced a total of ten dataset versions for early risk detection training, the sizes of which are reported in Table 9 (Appendix B).

4.2.2. Early risk detection models

In order to measure the impact of the different ACΔ thresholds and partial stream sampling methods, we trained and tested early risk detection models using the ten dataset versions

(Table 9). Specifically, we trained 5 models per dataset with different random seeds. Then, their averaged results were studied to select the final models for submission to the challenge.

We also implemented baseline systems consisting of binary classification models trained directly on entire streams (i.e., on the kickoff Train and Dev_d partitions described initially in Section 3). These baselines allowed us to measure the gains of our proposed method about automatically labelling streams at message level.

Details about model architecture and training setup can be consulted in Section 4.3 and Appendix C. Section 5 reports and analyses the results of all these models in our development test partition (Dev_t in Table 1), as well as the results in the official test set (Test) of the models selected for submission.

4.3. Classifier architecture and input handling

All the classifiers implemented throughout our reported experiments consist of BERT-based binary classifiers. We followed the standard layer stack recipe: a BERT encoder, whose output for the token [CLS] is pooled and fed to a dropout layer, followed by a dense linear layer that produces the logits for the target categories—namely, SUFFER and CONTROL. During training, the cross-entropy loss is back-propagated to fine-tune the models. In inference, the final label for the given input is simply the most probable one.

Streams were fed to the encoder by joining all the messages through the special token [SEP] and passing this text through the corresponding BERT tokeniser. The cases where the resulting subword sequence surpassed the maximum allowed length of the encoder were handled differently in each of the experimentation phases:

- For **message-level annotation** models (Section 4.1.1), we simply truncated the stream to the maximum allowed length. Given the observation that SUFFER streams usually contain relevant information for the class already in initial messages, we considered this a good compromise between implementation simplicity and expected performance.

- In the case of **early risk detection** models (Section 4.2.2), on the other hand, the most important message of the input sequence is by design the latest one—as the classifier receives messages from a live stream incrementally, it is the latest message that contains new information to consider. Thus, if necessary, the input sequences were truncated from the left, that is, discarding older content. Of note, the maximum input length was limited to 130 tokens during training and 512 in inference, as this setting produced better results in our preliminary experiments.

All the models are fine-tuned versions of DiagTrast-Berto [13], a Spanish BERT_{Base} (a.k.a., BETO) [14] post-trained on the synthetic corpus about mental disorders DiagTrast [15]. This base model was chosen during early experiments, where it obtained better results than BETO itself, Multilingual BERT [16] and a Spanish Longformer [17].

This decision was also conditioned by concerns related to efficiency, as MentalRiskES organisers encouraged participants to implement systems with low resource demand and carbon footprint. For this reason, we did not consider experimenting with larger Transformer-based models, despite their potential superior performance.

Specific model hyperparameters and training setup can be consulted in Appendix C.

5. Results

In this section, we will describe the results obtained in our development experiments and in the official challenge evaluation. It must be noted that despite our efforts to replicate the organisers' evaluation methodology, we have not been able to obtain exactly the same metrics as those officially reported on the test set. That is, the results obtained in our local experiments do not represent exactly the same metrics used in the final evaluation.

5.1. Development results

The results obtained by our models on the development test (Dev_t) are reported in Tables 2 and 3, for Task 1 (eating disorders) and Task 2 (depression) respectively. It must be noted that the reported results correspond in each case to the average of 5 models.

As can be seen, many of our models manage to surpass the simpler baselines trained on entire streams. With our proposed stream preprocessing method, we have managed to gain 11 and 15 F1-score points in Task 1a and Task 2a, respectively, despite the fact that the message-level datasets were labelled automatically, while the baselines are trained on gold labels.

Furthermore, it can be observed that models trained with the exhaustive datasets (∀s) managed to obtain better results than those of the 1-step datasets (1s) in almost all the cases. This observation applies both to binary classification metrics and, to a lesser extent, to early detection metrics. This is despite the fact that ∀s datasets contain information that was not sampled exclusively from the initial parts of the streams; on the other hand, it should be noted that ∀s datasets do contain many more training and development examples than 1s.

No such general remark can be made with respect to the impact of the different threshold heuristics. Although our models obtained the best results in both tasks with the GloMax_1 threshold and ∀s dataset, the differences are not particularly significant nor systematic. It is possible that a bigger test dataset is needed to measure the real impact of the different thresholds.

These results led us to submit the following models for official evaluation:

Task 1a (eating disorders)

- Run 0: ∀s + GloMax_1
- Run 1: ∀s + GloMax_n
- Run 2: ∀s + LocMax_n

Task 2a (depression)

- Run 0: Baseline
- Run 1: 1s + GloMax_1
- Run 2: ∀s + LocMax_n

5.2. Official results

The official results of our 3 runs per task can be consulted in Tables 4 and 5 for Task 1a and 2a respectively. For benchmarking purposes, these tables also report the results of the organisers' baselines and those of the best participants in terms of F1-score and/or F1_{LatW}. Carbon footprint measurements have been compiled in Appendix D.

Table 2
Results of Task 1a, early eating disorders risk detection, on our test dataset (Dev$_t$). Each row reports the average metrics of 5 models trained on the corresponding dataset. The best results per metric are highlighted in boldface.

Dataset		**Binary classification**				**Early Detection**				
PSS	**ACΔ th**	**Acc**	**P**	**R**	**F1**	**ERDE$_5$**	**ERDE$_{30}$**	**Lat**	**Speed**	**F1$_{LatW}$**
1s	naive	0.67	0.75	0.71	0.67	0.16	0.15	**1**	**1**	0.71
	LocMax$_1$	0.68	0.77	0.72	0.68	0.15	0.14	**1**	**1**	0.72
	LocMax$_n$	0.74	0.81	0.77	0.73	0.13	0.12	**1**	**1**	0.76
	GloMax$_1$	0.64	0.69	0.67	0.64	0.19	0.18	**1**	**1**	0.67
	GloMax$_n$	0.75	0.78	0.77	0.75	0.14	0.13	**1**	**1**	0.75
$\forall$s	naive	0.59	0.66	0.65	0.55	0.18	0.17	**1**	**1**	0.68
	LocMax$_1$	0.72	0.80	0.75	0.71	0.15	0.12	**1**	**1**	0.75
	LocMax$_n$	**0.81**	**0.85**	0.83	**0.81**	0.12	0.09	**1**	**1**	**0.82**
	GloMax$_1$	**0.81**	**0.85**	**0.84**	**0.81**	**0.11**	**0.08**	**1**	**1**	**0.82**
	GloMax$_n$	0.77	0.83	0.79	0.76	0.15	0.11	**1**	**1**	0.79
Baseline		0.71	0.78	0.74	0.70	0.18	0.14	1.25	**1**	0.72

Table 3
Results of Task 2a, early depression risk detection, on our test dataset (Dev$_t$). Each row reports the average metrics of 5 models trained on the corresponding dataset. The best results per metric are highlighted in boldface.

Dataset		**Binary classification**				**Early Detection**				
PSS	**ACΔ th**	**Acc**	**P**	**R**	**F1**	**ERDE$_5$**	**ERDE$_{30}$**	**Lat**	**Speed**	**F1$_{LatW}$**
1s	naive	0.54	0.41	0.52	0.42	0.28	0.26	**1**	**1**	0.68
	LocMax$_1$	0.57	0.38	0.54	0.42	0.25	0.23	**1**	**1**	0.71
	LocMax$_n$	0.55	0.47	0.52	0.39	0.26	0.24	**1**	**1**	0.70
	GloMax$_1$	0.62	0.63	0.60	0.53	**0.24**	**0.21**	**1**	**1**	**0.73**
	GloMax$_n$	0.56	0.41	0.54	0.42	0.27	0.24	**1**	**1**	0.70
$\forall$s	naive	0.61	0.65	0.59	0.54	0.25	0.22	**1**	**1**	0.72
	LocMax$_1$	0.57	0.61	0.55	0.49	0.28	0.25	**1**	**1**	0.69
	LocMax$_n$	0.63	**0.73**	0.61	0.57	0.25	**0.21**	**1**	**1**	**0.73**
	GloMax$_1$	0.64	0.68	0.63	0.60	0.29	0.22	**1**	**1**	0.72
	GloMax$_n$	**0.65**	0.67	**0.64**	**0.62**	0.33	0.22	1.20	**1**	0.72
Baseline		0.58	0.72	0.56	0.47	0.29	0.23	**1**	**1**	0.71

These results reveal that our systems are among the fastest of the participants, with some of the best trade-offs between F1-score and speed of the entire task. We obtained the second-best ERDE$_5$ value in Task 1a, having only been slightly outperformed by the Roberta$_{Large}$ baseline (a model with double the parameters than ours). We also obtained the second-best F1$_{LatW}$ value. The best F1$_{LatW}$ value was achieved by the team *CIMAT-NLP-GTO*, who have a

Table 4
Official results for Task 1a, detection of eating disorders risk, sorted by descending F1$_{LatW}$. Our systems are highlighted in a grey background. The best results per metric are highlighted in boldface.

	Binary classification				**Early Detection**				
	Acc	**P**	**R**	**F1**	**ERDE$_5$**	**ERDE$_{30}$**	**Lat**	**Speed**	**F1$_{LatW}$**
CIMAT-NLP-GTO	**0.97**	**0.96**	**0.97**	**0.97**	0.33	**0.02**	6	0.90	**0.86**
∀s + LocMax$_n$	0.88	0.88	0.89	0.88	0.17	0.07	3	0.96	0.83
∀s + GloMax$_n$	0.86	0.86	0.87	0.86	0.22	0.09	3	0.96	0.81
∀s + LocMax$_n$	0.85	0.85	0.85	0.85	0.23	0.11	3	0.96	0.79
Baseline Roberta$_{Large}$	0.81	0.82	0.83	0.81	**0.16**	0.10	**2**	**0.98**	0.79
Baseline Deberta	0.81	0.84	0.84	0.81	0.31	0.08	5	0.92	0.75
Baseline Roberta$_{Base}$	0.70	0.78	0.74	0.69	0.19	0.13	**2**	**0.98**	0.72

Table 5
Official results for Task 2a, detection depression risk, sorted by descending F1$_{LatW}$. Our systems are highlighted in a grey background. The best results per metric are highlighted in boldface.

	Binary classification				**Early Detection**				
	Acc	**P**	**R**	**F1**	**ERDE$_5$**	**ERDE$_{30}$**	**Lat**	**Speed**	**F1$_{LatW}$**
SINAI-SELA	0.73	0.78	0.74	0.72	0.40	**0.14**	4	0.95	**0.72**
Baseline Deberta	0.66	**0.79**	0.69	0.64	0.30	0.15	**2**	**0.98**	**0.72**
∀s + LocMax$_n$	0.65	0.75	0.68	0.63	**0.27**	0.17	**2**	**0.98**	0.71
1s + GloMax$_1$	0.64	0.74	0.66	0.62	0.28	0.18	**2**	**0.98**	0.70
Baseline Roberta$_{Large}$	0.70	0.76	0.72	0.69	0.29	0.16	4	0.95	0.70
Baseline	0.59	0.69	0.62	0.56	0.29	0.20	**2**	**0.98**	0.67
Baseline Roberta$_{Base}$	0.63	0.74	0.66	0.61	0.34	0.18	4	0.95	0.67
UMUTeam	**0.74**	0.76	**0.75**	**0.74**	0.55	0.36	30	0.56	0.42

remarkably better F1-score but double our latency. On a minor note, it should be reported that we encountered a problem upon submitting the predictions of the first round, which affected our speed-related metrics.

The trends are similar for Task 2a, although the absolute metrics are overall worse than for Task 1a, in line with our development results. We hypothesise that this task may have been more challenging for all participants because the problem being modelled is more subjective or ambiguous, as the annotator votes in Table 1 suggest.

In general, our ERDE$_{30}$ and F1-score values suggest that our systems are relatively premature in the classification. Other participants obtained better F1-score and ERDE$_{30}$ values than us by consuming more messages, which, in return, allowed them to be more accurate. However, most of our errors were committed in streams with a greater disagreement than average between the annotators, as we explain in the next section.

5.3. Error analysis

The false positive (FP) and false negative (FN) error counts on the Test are reported by task and run in Table 6. Run 0 ($\forall$s + GLOMAX$_1$) and Run 1 ($\forall$s + GLOMAX$_n$) are quite balanced in Task 1a, but Run 2 ($\forall$s + LOCMAX$_n$) makes twice FPs than FNs. This behaviour could be explained by the fact that LOCMAX$_n$ is a laxer threshold than the former two. The error types are much more imbalanced in Task 2a, where our systems have made more than 50 FP errors each. As explained before, this task turned out to be markedly more challenging for all the task participants.

Interestingly, the errors committed by the different systems involve to a great extent the same set of streams. Having manually analysed them, we observed that many of the repetitive FP errors were triggered by messages that mentioned words or phrases related to eating disorders (example E1) or depression (E2), but that do not necessarily imply that the user themselves is at risk of suffering those. In contrast, the FN errors generally involve streams that clearly convey mental health risk but were outright missed by the systems (e.g., E3 and E4).

E1) Que opinan sobre hacer ayuno para bajar de peso
What do you think about fasting to lose weight

E2) Y estoy aquí porqué mi Esposa fue diagnosticada con depresión y ansiedad
And I'm here because my Wife was diagnosed with depression and anxiety

E3) Yo me siento horrible con el cuerpo que tengo [...] intenté bomitar pero nunca pude
I feel horrible with the body I have [...] I tried to vomit but I never could

E4) Solo quiero ayudar y que los demás estén más felices [...] ya que yo no puedo [...]
I just want to help and make others happier [...] since I can't [...]

Given the subjective nature of the task, we also analysed the annotator vote distribution specifically on the instances that induced the errors on our systems. Table 6 includes the average number of votes received by stream: in the case of FP streams (that is, true CONTROL streams), the annotator agreement is better as the vote average gets closer to 0; conversely, the annotator agreement for FN streams (that is, true SUFFER streams) is better as the vote average gets closer to 10. As we compare these values to the total average votes of the dataset (the bottom row of the table), we observe that the annotator agreement is somewhat worse, overall, for the streams that induced errors in our systems. That is, this results suggest that our systems committed errors on instances that present more difficulties than average even to human annotators.

6. Conclusions and future work

This article described the participation of the Vicomtech NLP team in the MentalRiskES shared task. Specirfically, we have tackled the binary classification task for eating disorders and depression disorders, proposing Trasnformer-based systems that did not involve training with any external data nor the use of larger language models. Our proposal includes a novel message-level annotation algorithm based on the variation of confidence when dropping message from streams, and the training of DiagTrast-Berto models over these message-level annotated datasets.

Table 6
Error analysis on the official Test set, including a breakdown of the annotator votes on the instances that induced the errors

	Task 1a: Eating disorders (n=150)				Task 2a: Depression (n=149)			
	# FP	Votes$_{FP}$	# FN	Votes$_{FN}$	# FP	Votes$_{FP}$	# FN	Votes$_{FN}$
Run 0	11	2.72 ± 1.35	12	9.25 ± 0.75	56	2.14 ± 1.28	5	8.20 ± 2.05
Run 1	11	2.72 ± 1.35	12	9.25 ± 0.75	50	2.30 ± 1.33	7	8.29 ± 1.89
Run 2	14	2.43 ± 1.22	7	9.00 ± 0.57	53	2.11 ± 1.34	4	8.75 ± 1.89
Test		2.01 ± 1.15		9.48 ± 1.00		1.99 ± 1.38		9.06 ± 1.27

Our systems managed to obtain competitive metrics in comparison to the results by other participants, having achieved the second-best F1$_{LatW}$ and ERDE$_5$ values in Task 1a, and the best ERDE$_5$ and the third-best F1$_{LatW}$ in Task 2a These metrics measure how fast models recognise the desired pattern. Our systems might have been slightly penalised in terms of ERDE$_{30}$ or F1-score since we encouraged our models to be as fast as possible; however, further error analysis revealed that many of the errors were induced by data that produced notable disagreement among annotators.

Further work could focus on modifying our annotation algorithm, obtaining a BERT-like embedding by message, and looking at the attention layers of a trained a BERT model with a concatenation of those embeddings. Other studies could be leveraged training our systems with some external data from different sources, exploring different threshold heuristics, or modifying our system to prevent it from making premature classifications.

Acknowledgments

We would like to thank Aitor García Pablos for his insightful comments and recommendations. This work was partially funded by the Basque Business Development Agency, SPRI, under the ESAN project (grant agreement KK-2021/001179).

References

[1] A. Haakenstad, J. A. Yearwood, N. Fullman, C. Bintz, K. Bienhoff, M. R. Weaver, V. Nandakumar, J. N. Joffe, K. E. LeGrand, M. Knight, et al., Assessing Performance of the Healthcare Access and Quality Index, Overall and by Select Age Groups, for 204 Countries and Territories, 1990–2019: a Systematic Analysis from the Global Burden of Disease Study 2019, The Lancet Global Health 10 (2022) e1715–e1743.

[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). TBP.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is All you Need, in: Advances in Neural Information Processing Systems 30 (NIPS 2017), Curran Associates Inc., 2017, pp. 6000–6010.

[4] M. M. Tadesse, H. Lin, B. Xu, L. Yang, Detection of Depression-Related Posts in Reddit Social Media Forum, IEEE Access 7 (2019) 44883–44893.

[5] A. Yates, A. Cohan, N. Goharian, Depression and Self-Harm Risk Assessment in Online Forums, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2017, pp. 2968–2978.

[6] M. J. Vioulès, B. Moulahi, J. Azé, S. Bringay, Detection of Suicide-Related Posts in Twitter Data Streams, IBM Journal of Research and Development 62 (2018) 7:1–7:12.

[7] S. Ji, X. Li, Z. Huang, E. Cambria, Suicidal Ideation and Mental Disorder Detection With Attentive Relation Networks, Neural Computing and Applications 34 (2021) 10309–10319.

[8] J. Parapar, P. Martín, D. E. Losada, F. Crestani, Overview of eRisk 2022: Early Risk Prediction on the Internet, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction Proceedings of the Thirteenth International Conference of the CLEF Association (CLEF 2022), Springer International Publishing, 2022, pp. 233–256.

[9] J. M. Loyola, H. Thompson, S. Burdisso, M. Errecalde, UNSL at eRisk 2022: Decision Policies With History for Early Classification, in: G. Faggioli, N. Ferro, A. Hanbury, M. Potthast (Eds.), Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, Bologna, Italy, September 5th - to - 8th, 2022, volume 3180 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 947–960.

[10] A.-M. Bucur, A. Cosma, L. Dinu, Early Risk Detection of Pathological Gambling, Self-Harm and Depression Using BERT, in: Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, 2021, pp. 1–12.

[11] J. Devlin, M.-W. Chang, K. Lee, 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 2019), Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186.

[12] H. Fabregat, A. Duque, L. Araujo, J. Martínez-Romo, UNED-NLP at eRisk 2022: Analyzing Gambling Disorders in Social Media using Approximate Nearest Neighbors, in: Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum, 2022, pp. 1–12.

[13] A. M. Garrido, E. Mencia, M. Ángel Solís Orozco, J. C. V. Villegas, Model Card for "DiagTrast-Berto", 2023. URL: https://huggingface.co/hackathon-somos-nlp-2023/DiagTrast-Berto.

[14] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish Pre-Trained BERT Model and Evaluation Data, in: PML4DC at ICLR 2020, 2020, pp. 1–9.

[15] A. M. Garrido, E. Mencia, M. Ángel Solís Orozco, J. C. V. Villegas, Dataset Card for 'DiagTrast", 2023. URL: https://huggingface.co/datasets/hackathon-somos-nlp-2023/DiagTrast.

[16] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Multilingual Models (Google Research - BERT), 2019. URL: https://github.com/google-research/bert/blob/master/multilingual.md.

[17] Text Mining Unit (TeMU) at the Barcelona Supercomputing Center, Longformer Base Trained With Data from the National Library of Spain (BNE), 2022. URL: https://huggingface.co/PlanTL-GOB-ES/longformer-base-4096-bne-es.

[18] I. Loshchilov, F. Hutter, Decoupled Weight Decay Regularization, in: Proceedings of the 7th International Conference on Learning Representations (ICLR 2019), 2019, pp. 1–18.

[19] Mila, DataForGood, BCG GAMMA, Comet.ml, Haverford College, Codecarbon, 2021. URL: https://pypi.org/project/codecarbon/.

A. Message-level annotation example

Table 7 contains the complete message stream of a real training instance for Task 1, eating disorders. This is the same stream from which Figure 2 was computed. The table also shows the message-level annotations produced by 3 different average confidence Δ (ACΔ) thresholds (see Section 4.1).

B. Quantification of silver dataset versions

Table 8 describes the five versions of message-level annotations obtained by applying the different ACΔ thresholds (see Section 4.1) to the original streams. Table 9, in turn, describes the 10 dataset versions obtained after partial stream sampling (PSS) to the message-level annotated streams (see Section 4.2.1).

C. Model hyperparameters and training setup

Our models were trained on a NVIDIA GeForce RTX 2080 GPU with 11GB of memory using the AdamW optimiser [18] and the hyperparameters listed in Table 10. The models were implemented in Python 3.9 with `torch` (version 1.8.0) and Huggingface's `transformers` (version 4.21.2). Any hyperparameter no reported here should be interpreted as being set to the default values of the aforementioned libraries. The architecture of the models is described in Section 4.3.

D. Carbon footprint

Table 11 and Table 12 describe the estimated hardware electricity power consumption submitted by the participants that we cite in Section 5.2 (official baselines were excluded from these tables, as no footprint metrics were provided for those by the organisers) for the eating disorders task and the depression task, respectively. These metrics were obtained using the Code Carbon Python library [19], which estimates the carbon footprint according to the hardware usage, and the amount of carbon emissions used to produce that amount of energy in the configured country (see its documentation for more information).

Table 7
Complete eating disorders stream with the message-level annotations obtained with different threshold strategies. Verbalised emojis are highlighted in italics.

	Message	Naive	LocMax$_n$	GloMax$_1$
1	Ana Peso actual: 68 Peso meta: 50 Edad: 20	**SUFFER**	**SUFFER**	CONTROL
2	Algún tip para las uñas débiles y quebradizas?	CONTROL	CONTROL	CONTROL
3	Muchas gracias *cara feliz con ojos sonrientes cara feliz con ojos sonrientes*	CONTROL	CONTROL	CONTROL
4	Por las mañanas me tomo un café o como un poco de avena Entre desayuno y comida suele comer frutas (manzana, plátano, naranja, uvas, etc) A la hora de la comida como un poco de carne y arroz o sopa y vuelvo a comer fruta en la tarde	**SUFFER**	**SUFFER**	CONTROL
5	Pero ya como a las 10 u 11 quiero algo dulce como dulces, chocolates y todas esas cosas	**SUFFER**	**SUFFER**	CONTROL
6	Muchas gracias, ya he hecho yoga pero intentaré también lo del agua fría	CONTROL	CONTROL	CONTROL
7	Tal vez podrías usar un poco de maquillaje	CONTROL	CONTROL	CONTROL
8	También puedes masticar chicle	CONTROL	CONTROL	CONTROL
9	Es posible vomitar las calorías?	**SUFFER**	**SUFFER**	**SUFFER**
10	Mismo peso y misma meta *cara de por favor*	CONTROL	CONTROL	CONTROL
11	Haz ejercicio de respiración	CONTROL	CONTROL	CONTROL
12	Inhala y exhala lentamente	CONTROL	CONTROL	CONTROL
13	Aunque te cueste trata de respirar tranquilamente con ritmo	CONTROL	CONTROL	CONTROL
14	No tienes alguna canción que te relaje?	CONTROL	CONTROL	CONTROL
15	Tips para bajar de peso en 3 días por favor urgente	**SUFFER**	**SUFFER**	CONTROL
16	Muchas gracias *cara sonriente cara sonriente cara sonriente*	CONTROL	CONTROL	CONTROL
17	Okey solo agüita de piña	CONTROL	CONTROL	CONTROL
18	Eso ya lo tengo controlado	CONTROL	CONTROL	CONTROL
19	Estoy tan gorda que hice reventar una falda Me siento muy mal	**SUFFER**	**SUFFER**	CONTROL
20	No lo creo, se rompió con todo y cierre	CONTROL	CONTROL	CONTROL
21	Lo se, me la puse con la esperanza de que me quedaría *cara desanimada cara desanimada*	CONTROL	CONTROL	CONTROL
22	Y solo termino hecho un desastre	CONTROL	CONTROL	CONTROL
23	Sabes que no tienes hambre y simplemente comes porque te sientes estresad@ o con ansiedad	**SUFFER**	**SUFFER**	**SUFFER**
24	No puedes detenerte ni aunque pienses en qué vas a engordar y te verás horrible	**SUFFER**	**SUFFER**	**SUFFER**
25	Dónde viste esa dieta?	**SUFFER**	CONTROL	CONTROL
26	Y a mí parecer está bien	CONTROL	CONTROL	CONTROL
27	Solo será una semana	**SUFFER**	CONTROL	CONTROL
28	Además de complementarlo con mucha agua, no lo olvides *cara sonriente*	CONTROL	CONTROL	CONTROL

Table 8
Summary of silver annotations at message level in suffer streams per ACΔ threshold strategy

		Naive	LocMax$_1$	LocMax$_n$	GloMax$_1$	GloMax$_n$
Task 1: Eating disorders						
Train	# suffer messages	530	322	292	219	263
	# control messages	3,659	3,867	3,897	3,970	3,926
	% suffer per stream	4.21 ± 6.49	2.56 ± 4.18	2.32 ± 4.03	1.74 ± 2.96	2.09 ± 3.49
	Index of 1st suffer	1.63 ± 3.27	2.27 ± 3.94	2.41 ± 4.16	4.02 ± 6.40	3.64 ± 6.64
Dev$_d$	# suffer messages	162	109	106	92	89
	# control messages	860	913	916	930	933
	% suffer per stream	5.79 ± 7.94	3.89 ± 5.33	3.79 ± 5.14	3.29 ± 5.07	3.18 ± 4.99
	Index of 1st suffer	1.67 ± 3.27	1.83 ± 3.24	2.08 ± 3.20	3.58 ± 5.92	3.58 ± 5.92
Task 2: Depression						
Train	# suffer messages	751	474	427	357	386
	# control messages	3,343	3,620	3,667	3,737	3,708
	% suffer per stream	6.26 ± 7.45	3.95 ± 4.95	3.56 ± 4.64	2.98 ± 4.16	3.22 ± 4.41
	Index of 1st suffer	1.27 ± 2.17	2.72 ± 4.70	2.72 ± 4.40	3.38 ± 5.54	3.13 ± 4.99
Dev$_d$	# suffer messages	227	118	117	114	108
	# control messages	1.107	1.216	1.217	1.220	1.226
	% suffer per stream	8.11 ± 10.72	4.21 ± 7.48	4.18 ± 7.38	4.07 ± 7.87	3.86 ± 7.85
	Index of 1st suffer	1.47 ± 2.94	3.80 ± 6.33	3.47 ± 4.66	4.87 ± 10.67	6.13 ± 11.47

Table 9
Partial stream datasets for early risk detection training, per sampling and ACΔ threshold strategy (suf=suffer; ctr=control; Tot=Total)

		Task 1: Eating disorders						Task 2: Depression					
Dataset		Train			Dev$_d$			Train			Dev$_d$		
PSS	ACΔ th	suf	ctr	Tot	suf	ctr	Tot	suf	ctr	Tot	suf	ctr	Tot
1s	naive	63	91	154	13	20	33	81	65	146	20	15	35
	LocMax$_1$	61	97	158	13	20	33	83	64	147	20	16	36
	LocMax$_n$	61	92	153	14	19	33	82	65	147	20	17	37
	GloMax$_1$	59	94	153	13	20	33	77	69	146	18	16	34
	GloMax$_n$	60	98	158	13	20	33	74	73	147	18	15	33
∀s	naive	368	407	775	100	81	181	492	333	825	115	87	202
	LocMax$_1$	241	416	657	91	106	197	361	382	743	81	102	183
	LocMax$_n$	221	418	639	70	102	172	301	393	694	78	90	168
	GloMax$_1$	185	437	622	73	109	182	267	371	638	73	88	161
	GloMax$_n$	205	435	640	70	111	181	310	396	706	74	103	177

Table 10
Hyperparameters for message-level annotations models (MLA; Section 4.1) and final early risk detection models (ERD; Section 4.2)

	MLA		ERD	
	Train	**Infer**	**Train**	**Infer**
Max input length	512	512	130	512
Batch size	8	8	8	1
Gradient clipping	2		2	
Learning rate	1e-05		1e-05	
Warm-up epochs	2		2	
Dropout rate	0.4		0.3	
Max epochs	80		80	
Early stopping patience	5		5	

Table 11
Mean carbon footprint metrics obtained for Task 1a sorted in ascending order by emissions.

Team	Duration (s)	Emissions (Kg)	CPU Energy (kW)	GPU Energy (kW)	RAM Energy (kW)
∀s + LocMax$_n$	3.62	**4.56e-05**	**8.86e-05**	**1.50e-04**	**1.41e-06**
∀s + GloMax$_n$	3.62	4.66e-05	8.85e-05	1.55e-04	1.41e-06
∀s + LocMax$_n$	3.61	4.71e-05	8.83e-05	1.58e-04	1.42e-06
CIMAT-NLP-GTO	**3.28**	2.55e-04	1.80e-04	3.42e-04	5.67e-07

Table 12
Mean carbon footprint metrics obtained for Task 2a sorted in ascending order by emissions.

Team	Duration (s)	Emissions (Kg)	CPU Energy (kW)	GPU Energy (kW)	RAM Energy (kW)
UMUTeam	19.49	**5.52e-08**	**1.02e-07**	**1.88e-07**	**7.73e-10**
SINAI-SELA	30.58	9.17e-06	8.68e-06	1.13e-05	3.01e-07
∀s + LocMax$_n$	**3.01**	3.79e-05	7.35e-05	1.24e-04	1.18e-06
1s + GloMax$_1$	3.27	3.86e-05	7.90e-05	1.23e-04	1.27e-06
Baseline	3.38	4.13e-05	8.13e-05	1.34e-04	1.35e-06