Data Augmentation via Generative LLMs for the Detection of Gambling Disorders and Type of Addiction in Social Media Threads

Ibai Sologuestoa[1,*], Xabier Larrayoz[2], Maite Oronoz[2] and Alicia Pérez[1,2]

[1] *Escuela de Ingeniería de Bilbao, University of the Basque Country UPV/EHU, Spain*
[2] *HiTZ Center - Ixa, University of the Basque Country UPV/EHU (http://www.hitz.eus), Spain*

Abstract

This work addresses natural language processing in Spanish-language social media. The goal is twofold: first, the early detection of users at high risk of having gambling disorders, based on a set of their messages; second, determining the type of gambling addiction (Betting, Online Gaming, Trading, and Lootboxes). We employed supervised classification approaches; however, these methods inherently depend on the availability of large annotated corpora. In an attempt to cope with this challenge, we explored the incorporation of artificial user message threads generated using Generative Large Language Models, creating synthetic counterparts by user-segment for high or low risk users. For the early detection of user's gambling disorder risk, we proposed a hybrid Bi-LSTM model trained with the GroupDRO loss and enhanced by dual attention mechanisms—capturing learned and lexicon-based risk—alongside data augmentation. To support high-risk alarm decisions based on the classifier's output, we investigated the use of dynamic thresholds. Dynamic thresholds were intended as a trade-off between earliness in decision making and unfair alarm raising for users with long-enough message sequences. The approach achieved a competitive Macro-F1 score of 0.475. For classifying the type of addiction, the classification model employs a hierarchical Bidirectional Long Short-Term Memory (Bi-LSTM) achieving a Macro-F1 score of 0.856. In addition, we designed our approaches to be lightweight and versatile, capable of running on modest hardware without requiring GPU acceleration.

Keywords

Generative LLMs, Early Detection, Dual Attention, Group Loss Function, Data Augmentation,

1. Introduction

Gambling activities have been interwoven with human society throughout history, but recent decades have witnessed an unprecedented acceleration in their proliferation and accessibility. Of particular concern is the emergence of novel gambling modalities such as online casinos and in-game lootbox mechanisms [1], which have expanded the gambling ecosystem beyond traditional boundaries.

At the same time, sophisticated marketing strategies have emerged, using digital influencers on platforms like Twitch, where streamers promote gambling activities in ways that reduce the perceived risks associated with gambling [2]. Perhaps more concerning is the early exposure of younger demographics to gambling adjacent mechanics through colourful, engaging lootbox systems in video games potentially boosting gambling behaviours that may develop into clinical disorders later in life.

The relationship between exposure to gambling mechanics and the development of problematic gambling behaviours [3] highlights the critical importance of early detection and intervention systems across all demographic groups. Identifying at-risk individuals becomes essential for implementing timely preventive measures and appropriate support resources.

Given the critical importance of early detection for problematic gambling behaviours, two **tasks** are distinguished with the following specifications:

IberLEF 2025, September 2025, Zaragoza, Spain

[*] Corresponding author.

isologuestoa001@ikasle.ehu.eus (I. Sologuestoa); xabier.larrayoz@ehu.eus (X. Larrayoz); maite.oronoz@ehu.eus (M. Oronoz); alicia.perez@ehu.eus (A. Pérez)
http://ixa.si.ehu.eus (A. Pérez)
0009-0003-7308-5965 (X. Larrayoz); 0000-0001-9097-6047 (M. Oronoz); 0000-0003-2638-9598 (A. Pérez)

1. Detection of risk: early detection of gambling disorder risk for each user consists of seizing the risk of developing a gambling-related disorder for each user employing as fewer messages as possible. This can be approached as a binary classification task with two risk levels: low and high.
2. Type of addiction: regardless of the risk-level (either low or high risk), the type of addiction has to be determined, one out of these ones: Betting, Online Gaming, Trading, and Lootboxes. This can be approached as a multi-class classification in a mono-label setting, as each user has assigned one and only one of the labels as if the labels were mutually exclusive. The specifications state that the the type of addiction is estimated in the last round.

While these task definitions establish our objectives, effective approaches require consideration of previous work in this domain. The following section examines related research that informs our methodology.

2. Related work

To address this work we focused on antecedents in similar activities, such as the latest MentalRiskES [4, 5] and the CLEF eRisk related editions [6, 7]. In eRisk 2023, pathological gambling was already explored, but the focus remained exclusively on early detection, without addressing the specific type of addiction. In contrast, the gambling domain is introduced for the first time in MentalRiskES 2025, expanding the thematic scope of the shared task. While the addiction-type classification task is novel in this context, the underlying formulation as a multiclass classification problem has been present in previous MentalRiskES editions, albeit applied to other mental health conditions such as depression, eating disorders, or suicidal ideation.

In the previous edition of MentalRiskES, most of the participant systems focused on Transformer-based models, frequently relying on Spanish pre-trained variants such as RoBERTa [8] and BETO [9]. Several teams combined these models with diverse strategies to enhance classification performance. For instance, Ixa-Med [10] introduced a heuristic message-level re-labelling based on embedding similarity, while ELiRF-VRAIN [11] explored both classical methods like Support Vector Machines and Transformer-based architectures, including RoBERTa and Longformer, to manage long texts. UNED-GELP [12] adopted a two-step approach using BETO and ANN models, and UnibucAI [13] integrated RoBERTuito with the use of the Long Short-Term Memory (LSTM) architecture and experimented with context-aware strategies. These approaches reflect a growing trend towards combining deep learning with task-specific preprocessing and representation techniques.

3. Methods

In a set of **users**, $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$, each user publishes their messages in different time-stamps. In this task, the messages are released sequentially, one message per round by each user (regardless of the real time-stamp). Formally, in round r for user $u \in \mathcal{U}$, a **message** $X_u^{(r)}$ is released. Let us denote by $M_u^{[1,r]} = \{X_u^{(1)}, X_u^{(2)}, \ldots, X_u^{(r)}\}$ the **sequence of messages** at hand by round r for user u, that is, the history or accumulated set of messages. Naturally, the total number of messages published differs from user to user. Let us denote by R_u the total number of messages published by user u; in other words, R_u represents the 'last round' and is dependant on the user.

In the detection of user risk, the optimization criteria involve both Macro-F1 and ERDE in an attempt to keep balance between prediction reliability and speed, while in the type of addiction task the optimization criterion is just Macro-F1. Regarding the optimization criteria, while for the detection task earliness is a key factor in the specifications, for the determination of the type of addiction, oddly enough, the prediction is received in the last round. As a result, for the detection task, the input information available is $M_u^{[1,r]}$ with r as small as possible; by contrast, to determine the type of addiction, the input information available is $M_u^{[1,R_u]}$, all the messages up to the last round for the user.

3.1. Input vectorization

Each message has to be represented as a numeric feature-vector that serves as the input to the classifier. The generation of this vector is critical as it has to convey semantics into a multidimensional numeric space ($\mathbb{R}^n$). That is, semantic relatedness has to be encoded and computed in a vector space.

In an attempt to get the text represented as numeric feature-vectors we turned to available encoders with the aim of not investing time in task-specific fine-tuning. With this, just a shallow **preprocessing** was applied to avoid encoding incompatibilities, as follows: atypical and infrequent characters were substituted by a space character; emojis within the UTF-16 encoding spectrum were also replaced for a space character.

With regard to the text encoding, our approach turned to Google's **embeddings**, to be precise, to the `text-embedding-004` model[1] [14]. We opted for these embeddings for several reasons: easy out-of-the-box deployment, competitive in general domains, multilingual approach and, above all, a large-context window, allowing inputs up to 2048 tokens without truncation. The number of tokens in the input is, needless to say, a bottleneck in large text processing and the development is being rapid. For example, the default limit was set to 512 tokens in BERT [15] and in 8000 in Gemini Embeddings [16, 17]. In this task, each message, $X_u^{(r)}$, tends to be short with an average of $\Delta 9.58 \pm 12.85$ tokens per message.

Nevertheless, since we need to guarantee the ability to tackle all messages, we need to ensure that the sequence $M_u^{[1,R_u]}$ can be vectorized. The number of tokens by user (with all the messages concatenated) is, on average, $|M_u^{[1,R_u]}| = 615.53 \pm 477.74$. Further details about the token distribution per user are shown in Figure 1. Assuming roughly one token corresponds to four characters [18], a text of 120 characters would correspond to about 30 tokens, for example.

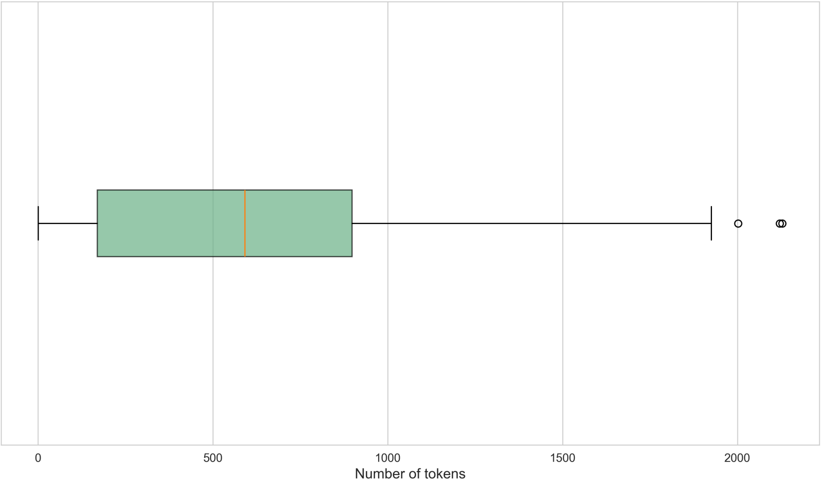

Figure 1: Distribution of total tokens per user i.e. $|M_u^{[1,R_u]}|$.

In any case, the strategy implemented to deal with **input token limitation** inherent to each transformer tool could be described as follows: the sequence of all messages $M_u^{[1,R_u]}$ was processed in consecutive sub-sequences $\{M_u^{[1,j]}, M_u^{[j+1,k]}, \ldots, M_u^{[l+1,R_u]}\}$. Next, each sub-sequence is embedded as in expression (1), with Σ denoting the input message vocabulary. With the number of chunks in each batch, k, being dynamically chosen not to surpass the default limit of tokens.

$$\begin{array}{rccl} e: & \Sigma^* & \longrightarrow & \mathbb{R}^n \\ & M_i^{[j+1,k]} & \longrightarrow & e(M_i^{[j+1,k]}) \end{array} \tag{1}$$

[1] `text-embedding-004` model is publicly available on https://aistudio.google.com/apikey

Finally, the mean of the embedded representations is obtained, as in (2).

$$e(M_i^{[1,R_i]}) = Avg\left(e(M_i^{[1,j]}), e(M_i^{[j+1,k]}), \ldots, e(M_i^{[l,R_i]})\right) \tag{2}$$

To sum up, we envisaged a mechanism to deal with lengthy messages which, needless to say, involves two arguable decisions: on the one hand, the way in which the chunks are chosen so that each sub-sequence fits the input token limitation and, on the other hand, the decision to average them all as a means to merge the meaning from all the chunks. All in all, with the input length size in these tasks, seldom will we need to resort to this averaged sub-sequence embeddings but in case it happens, at least, we count on a simple mechanism to deal with the exception.

So far we mentioned the means in which the inputs were vectorized. However, each input did not merely bring the message, instead, each input comprises four elements: the id_message identifier, the message, the date, and the platform. In our approach we decided to include, the date concatenated to the text message due to the fact that in preliminary experiments the date seemed to bring relevant nuances for both classification tasks. To sum up, each **input instance vectorization** was described as a vector embedding with both the date and message together. In an attempt to refrain from repetitive computations, the embeddings where cached locally. Hence, for cases where the text already existed, it only needed to be loaded from disk, saving time.

3.2. Data augmentation via generative LLMs

Data scarcity is a common challenge in mental health text classification tasks. While the original dataset provides a foundation, it lacks sufficient examples of varied linguistic expressions for robust model training. Our architecture benefits from augmented data through improved generalization.

Not only sufficient but also representative supervised data are the key cornerstone to train robust classification models. The absence of corpora is a challenge for inferred approaches. To tackle this, we turned to Generative Large Language Models (LLMs). Eventually, this resulted in the development of one of the most productive strategies in this work, despite its simplicity.

Using the original training set as a basis, we generated artificial variants of the original messages focusing, primarily, on these aspects: verb tenses, register (formal or informal) and, orthographic and grammatical errors. The motivation was to get semantically similar sentences keeping the label (both risk level on the first task or addiction type, on the second). We discarded generating negative or speculative variants not to risk the reference label. Shallowly speaking, we are generating not a complete digital twin but somehow a synthetic counterpart of given users.

Formally, given the original supervised set of sequences of messages and user label, $\mathcal{T}$ as in (3), an artificial alternative generated by message, $g(M_u^{[1,R_u]}) = \widetilde{M}_u^{[1,R_u]}$, while keeping the original user label, leading to an artificially generated set, $\widetilde{\mathcal{T}}$ as in (4). In the artificial set, we distinguished two segments, mutually exclusive by user label, as in (5) bound to $\widetilde{\mathcal{T}} = \widetilde{\mathcal{T}}_0 \cup \widetilde{\mathcal{T}}_1$. This generative approach led us to explore different alternative settings for each run (with an impact in Tables 2 and 6).

$$\begin{aligned}
\mathcal{T} &= \{(M_u^{[1,R_u]}, C_u)\}_{u=1}^{N} && (3)\\
\widetilde{\mathcal{T}} &= \{(\widetilde{M}_u^{[1,R_u]}, C_u)\}_{u=1}^{N} && (4)\\
\widetilde{\mathcal{T}}_i &= \{(\widetilde{\mathcal{M}}_u^{[1,R_u]}, C_u) \in \widetilde{\mathcal{T}} : C_u = i\} \quad \text{with } i \in \{0,1\} && (5)
\end{aligned}$$

With regard to the specific generative model selected, we opted for Gemini Flash[2], to be precise, `gemini-2.0-flash-thinking-exp` [19]. We configured it for data augmentation by instructing it to generate Spanish text variations that preserve the original meaning and key information while utilizing different sentence structures and synonyms. The system varies formality in register and tone without altering the core message and retains the same emotional content and sentiment. It ensures that the

[2] Gemini Flash, developed by Google available at https://ai.google.dev/gemini-api/docs/thinking

output remains natural and fluent to enhance friendliness, which may include emojis, colloquial slang, and intentionally varied orthographic and grammatical mistakes (though it avoids repeating identical errors). The system preserves any clinical significance and risk indicators in mental-health-related texts. Furthermore, it avoids reusing original words, phrases, punctuation, or sentence patterns, as it prioritizes data augmentation as its primary objective. The output format is presented as a JSON array that mirrors the input structure and maintains the class labels. For each original message, we generated one variant; We could have augmented artificially the corpus with more variants per user-message, but decided not to, in an attempt to avoid overfitting.

At this point the question arising was whether augment the texts of both types of users (high and low risk) or augment only the segment on which the classifiers were more error-prone. Preliminary experiments were conducted augmenting, in turns, each segment and jointly augmenting both of them. These results were helpful and shed light to make the decision.

3.3. Early detection of user gambling disorders

In this section we provide details of the approaches involved to cope with **Task 1** i.e. early detection of user gambling risk (either low or high): the message classifier, the strategy to optimize the classifier and the strategy to make a decision to generate the user label with the information provided by the message classifier.

3.3.1. Classifier

The estimated risk level of gambling disorders was addressed as supervised binary classification. The core of the approach rests in a Bidirectional Long Short-Term Memory (Bi-LSTM) [20] network (made available in Pytorch) augmented with attention mechanisms following the idea by Zhou et al. [21] but improved with a more modern transformer based self-attention [22] together with a lexicon based attention [23]. The bidirectional nature of the LSTM allows the model to capture dependencies in both forward and backward directions, a thing that is essential when tackling complex tasks like determining the risk of developing gambling disorders.

Let us provide some technical details to favour reproducibility. The **architecture** starts with an input normalization and regularization stage, applying moderate dropout (20%) and layer normalization to stabilize training. Followed by two stacked Bi-LSTM layers with a pretty aggressive dropout rate of 50% between layers. Each layer processes the input bidirectionally, capturing sequential dependencies in both forward and backward directions. This bidirectional design enables the model to incorporate contextual information from the entire sequence, crucial for identifying complex gambling risk patterns that may manifest throughout the text. What distinguishes this model is its specialized attention mechanism, implemented through a custom module where the manually-curated lexicon attention and the inferred attention are combined:

- Inferred Attention: A standard single head attention, as described by Vaswani et al. [22], automatically discovers which parts of the input sequence are most relevant for a successful detection.

- Lexicon Attention: A predefined set of risk-related terms (e.g., “debt”, “chasing losses”, “addiction”) is embedded into vectors using the Google’s embedding model as describe in section 3.1. These vectors guide the attention mechanism by computing a risk-relevance score for each hidden state, boosting the influence of gambling-specific features. Approximately, the risk-related term-vocabulary includes above 200 terms generated via LLMs and subsequently refined through manual review. The terms fall broadly into the following categories: *Clinical and diagnostic terms; Problematic gambling behaviours; Financial distress indicators; Psychological and emotional indicators; Social and family impact; Treatment and self-help terms; Terms for betting, trading, lootboxes, online gambling and crypto; Rationalization and denial phrases; Colloquial high-risk expressions; Extreme consequence indicators; and Severe addiction indicators.*

The combination of the two attention mechanisms is encompassed as in (6), where the lexicon attention is used to weight the terms that the inferred attention obtains or shed light to potentially highly relevant terms not detected by it. Consequently, the vector is passed through a normalization layer and then the mean of the vector is returned.

$$\text{Combined Attention} = \text{Inferred attention} \odot (1 + \text{Lexicon attention}) \tag{6}$$

In the **inference** stage, the settings were as follows:

- Initialization: The weights of the classifier are initialized making use of Xavier initialization [24]; regarding the optimizer, `AdamW` [25] was employed, that is, an improved version of Adam approach; a learning rate of 0.05 was decided with the help of the `ReduceLROnPlateau` [26] scheduler. Ending with a risk probability $p_u \in [0, 1]$ for binary classification (high/low risk), calculated for each round as in (8).
- Optimization criteria: A challenge in model inference with unbalanced class distributions tends to be a bias towards majority class. In this case, given that the differences to perceive either low or high risk in language might be subtle, in the learning stage we turned to Group Distributionally Robust Optimization (GroupDRO) [27, 28]. The loss function aims to improve model robustness by focusing on the performance of the worst-performing subgroups within the data. Instead of minimizing the simple average loss across all samples, it dynamically adjusts weights for different data groups based on their error magnitude during training and minimizes a weighted average loss.
  - Group formation: In our implementation, 5 groups are formed dynamically based on the error quantiles of the predictions inspired by [29]. Instances are ranked by decreasing error and split by error percentiles (leading to 5 groups). During training stage, the model would progressively allocate more attention to groups with higher errors.
  - Weight update mechanism: As a contribution in our work, in favour of gradual weight updates based on persistent performance patterns (rather than strong weight fluctuations due to temporary or sporadic differences), instead of employing the original GroupDRO, in our case the group weights are updated using an exponential moving average approach [30].
- Training corpus: Each training instance in the training corpus was built as the concatenation of all the messages (together with the date) by user with its corresponding user-risk label (C_u), namely, $\mathcal{T} = \{(M_u^{[1,R_u]}, C_u)\}_{u=1}^{N}$. Thus, the size of the training corpus was bound to the number of users ($|\mathcal{T}| = N = |\mathcal{U}|$), a small set of $N = 357$ users (350 in the supervised train set and 7 in the trial set), this is why we enhanced the corpus with artificially generated inputs as mentioned in section 3.2. In different runs, we explored augmenting only users with $C_u = 0$ or both $C_u = 0$ and $C_u = 1$, as stated in the settings of each task (Tables 2 and 6), accordingly, thereby doubling the size of the supervised dataset. Needless to say, we could have generated more than a single synthetic counterpart for each user and generate more instances to assess the impact of the size of artificially generated data in the learning stage, but this point remains open for further research.

3.3.2. Decision strategy for user gambling risk

The decision strategy determines whether a user u is classified as either high or low risk. In our case, this decision is based on the cumulative risk assessment across rounds. For each round r, and each user u, the classifier provides the likelihood of high risk evidences in the sequence of messages $M_u^{[1,r]}$, $p_u^{(r)}$, as in (7).

$$\begin{array}{rccl} l: & \Sigma^* & \longrightarrow & [0,1] \\ & M_u^{[1,r]} & \longrightarrow & l(M_u^{[1,r]}) = p_u^{(r)} \end{array} \tag{7}$$

We could have decided to raise the alarm based on $p_u^{(r)}$, by contrast, preliminary experimental results led us to make the decision in a different way. Arguably enough, we accounted for the accumulated risk score for user u up to round r ($s_u^{(r)}$), as a simple sum, expressed in (8). This, somehow implies that first impressions result inertial in the decision.

$$s_u^{(r)} = \sum_{t=1}^{r} p_u^{(t)} \tag{8}$$

At round r the system makes the decision to label each user as high risk if $s_u^{(r)}$ exceeds a **dynamic threshold** $\theta^{(r)}$. The threshold increases over time (or more specifically, over rounds) as defined in (9) with θ_0 being an offset or initial-state threshold and κ being a scaling factor. In plain words, the threshold is a mere floor division of r divided by κ constant with the offset θ_0. Arbitrarily, we chose a floor division in an attempt to refrain us from varying the threshold every round and, instead, limit variations to regular batches of κ rounds (leading to a stair-shaped threshold function). This decision was adopted for both run 0 and 1, by contrast, a common division was employed for run 2 (varying the threshold dynamically, in every round indeed, leading to a lineal shaped function).

$$\theta^{(r)} = \theta_0 + \left\lfloor \frac{r}{\kappa} \right\rfloor \tag{9}$$

The motivation to use a lineally increasing threshold is to reward the gain of a stronger evidence as more data become available so that user decision label is not benefitted for the mere fact of having a longer chat history (R_u). In other words, if the threshold had been constant (e.g. θ_0) then $s_u^{(r)}$ would have always surpassed it with sufficient enough messages (with r large enough) and, thus, end up always raising the alarm of high risk. In order to prevent this and put the focus on more realistic situations we opted for a dynamically increasing threshold. Our approach mitigates raising alarms by mere exposure to further messages and becomes slightly more realistic than with a constant threshold setting. In an attempt to avoid raising the alarm by some r, we could have also addressed the user accumulated risk score (8) instead of the dynamic threshold. Finally, let us mention that even though these parameters (θ_0 and κ) are trainable, in practice, we decided them on a preliminary experimental basis.

To sum up, the decision strategy to determine user-label risk in round r is as in (10) where 1 indicates high risk and 0 indicates low risk.

$$\text{Risk}\,(u, r) = \begin{cases} 1 & \text{if } s_{u_i}^{(r)} \geq \theta^{(r)}, \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Note that, while this dynamic threshold is a step ahead towards handling realistic situations, it is also true that it promotes conservative decisions detrimental to detection speed or earliness. We consider the proposal of the threshold an open research question worth of further research as it is a cornerstone in the task.

3.4. Type of addiction estimation

In this section we provide details of the methodology followed to cope with **Task 2** that is, to determine the type of addiction. The methods proposed are bound to the difference in criteria from Task 1 to Task 2 in what earliness and available information regards (mentioned in page 2).

3.4.1. Classifier

The classification model employs a hierarchical Bidirectional Long Short-Term Memory (Bi-LSTM) [20] architecture (with the help of the Pytorch library) for predicting gambling disorder risk across four categories. The approach begins with a LockedDropout (20%) mechanism [31](by contrast to the approach described in section 3.3.1). This mechanism applies consistent masking across sequence dimensions, preserving temporal coherence during training. It was used because this approach typically works best with RNNs.

The core of the architecture consists of two stacked Bi-LSTM layers [20]. Each layer processes the sequence bidirectionally, capturing sequential dependencies in the text in both forward and backward directions, while a more aggressive dropout of 40% was employed between the layers, this was possible as having Bi-LSTM layers practically duplicate the network's dimension, e.g. a 64 dimension input will transform into a 128 dimension output after passing a Bi-LSTM layer. With a unidirectional LSTM this would not have been an issue. At last a final dropout layer of 30% precedes the classification head. The terminal hidden state from the second Bi-LSTM layer serves as the sequence representation, which is then projected through a fully connected layer to produce a multi-dimensional output (as many as class-labels in this mono-label and multi-class classification task i.e. 4).
In the **inference** stage, the settings were as follows:

- Initialization: The model employs Xavier Normal initialization [24] (as it was the case of the approach described in section 3.3.1 for Task 1) for all weights, while biases are initialized to zero. This initialization strategy helps to maintain appropriate activation magnitudes throughout the network during forward propagation. As for the optimizer and scheduler are the same ones used in (3.3.1) AdamW [25] and ReduceLROnPlateau [26].

- Optimization criteria: This classifier employed the well-known CrossEntropyLoss [32] function for the optimization stage.

- Training corpus: The training corpus is identical to the one defined in 3.3.1.

3.4.2. Decision strategy on type of addiction

Given the sequence of messages available from the 1st round to round r by user u, i.e. $M_u^{[1,r]}$, the classifier provides the confidence of each addiction type a (with $a \in$ {Betting, Online Gaming, Trading, Lootboxes}) as $p(M_u^{[1,r]}, a) \in [0,1]$ such that $\sum_a p(M_u^{[1,r]}, a) = 1$ in (8). The sequence of messages available in the current round (r) is embedded and next the multi-class classification decision made. We explored two alternative means of estimating the addiction. The first one, $Addiction_{All}()$, is a straightforward decision that concatenates all the messages and provides the addiction type with highest score in the last round (R_u) as in (11).

$$Addiction_{All}(M_u^{[1,R_u]}) = \arg\max_a p(M_u^{[1,R_u]}, a) \tag{11}$$

The second means, $Addiction_{FIM}()$, exploits the first impressions mostly (denoted as FIM), as in (12). FIM gets the confidence with each sequence of messages available, sums them all and gets the addiction with the highest overall confidence score.

$$Addiction_{FIM}(M_u^{[1,R_u]}) = \arg\max_a \sum_{r=1}^{R_u} p(M_u^{[1,r]}, a) \tag{12}$$

The FIM approach entails the first messages in all subsequent rounds to get the confidence and eventually sums them all with the consequence of providing cumulated relevance to early messages. The FIM decision was implemented as a mere toy for this challenge and performed equal to the same model with the so-called All, meaning that it had little impact overall. Ultimately, FIM was ranked 6th out of 32 approaches having a tie with the All run at 5th out of 32.

With the limitation that the total number of messages (last round, R_u) is user dependant, in the real scenario our system submitted the decision made thus far, $\hat{a}_u{}^{(r)}$, every round ($1 \leq r \leq R_u$) while the specifications stated that the approaches would be ranking just with the predictions sent on the final round ($r = R_u$). This might have an undesirable impact in carbon emissions.

4. Experimental results

Registered as SoloResearch team, denoted after the family name of the first author, Sologuestoa, we attained a rank of 11 over 38 in Task 1 and 5 over 32 in Task 2. Below we analyze the performance of the methods presented.

4.1. Task 1: Early detection performance

Our methods were assessed employing the data made available in MentalRiskES 2025 [33, 34, 35]. Some details about the corpus are shown in Table 1.

Table 1
Data made available in MentalRiskES for the early detection of user gambling risk.

	Train	**Trial**	**Total**
Low risk	178	4	182
High risk	172	3	175
Total	350	7	357

Three runs were submitted each of which with the specifications stated in Table 2. The settings differ on the dynamic threshold set to make the decision, $\theta^{(r)}$ described in (9), and on the data augmentation strategy, adding to the original training set ($\mathcal{T}$) data generated for either only low-risk ($\widetilde{\mathcal{T}}_0$) or both ($\widetilde{\mathcal{T}}_0 \cup \widetilde{\mathcal{T}}_1$) types of messages, as described in section 3.2. The settings provide versatility, with the target involved both the latency (closely related to the threshold) and the accuracy (via data augmentation).

Table 2
Run settings for Task 1.

Run	Threshold	Data augmentation
Run 0	$2+\lfloor \frac{r}{20} \rfloor$	$\mathcal{T} \cup \widetilde{\mathcal{T}}_0 \cup \widetilde{\mathcal{T}}_1$
Run 1	$2+\lfloor \frac{r}{20} \rfloor$	$\mathcal{T} \cup \widetilde{\mathcal{T}}_0$
Run 2	$3+r/6$	$\mathcal{T} \cup \widetilde{\mathcal{T}}_0$

On the one hand, classification accuracy metrics achieved in this task together with the final rank are shown in Table 3 and, on the other hand, their corresponding speed and latency-weighted accuracy are given in Table 4.

Table 3
Performance metrics for each run in Task 1.

Run	Rank	Accuracy	Macro-P	Macro-R	Macro-F1
Run 0	15/38	0.494	0.476	0.483	0.446
Run 1	13/38	0.500	0.486	0.490	0.455
Run 2	11/38	0.506	0.497	0.498	0.475

Table 4
Speed and latency metrics for each run in Task 1.

Run	ERDE5	ERDE30	LatencyTP	Speed	Latency-weighted F1
Run 0	0.639	0.424	11	0.904	0.550
Run 1	0.674	0.458	20	0.820	0.501
Run 2	0.691	0.487	26	0.769	0.464

Note that, in Table 3 Run 2 achieved better Macro-F1 than the other two runs, with a score of 0.475 (Rank 11 over a total of 38 runs taking part in this task) in comparison to Run 0 and 1 (ranked in 15th and 13th positions respectively). Nevertheless, detection speed is a key factor in this task and, as shown in Table 4, Run 0 resulted in the fastest approach, leading to the best Latency-weighted F1.

4.2. Task 2: Type of addiction estimation performance

The methods presented were assessed by means of the corpus employed in MentalRiskES 2025 [33, 34, 35]. Some details are summarized in Table 5.

Table 5
Distribution of type of addiction in the joint Train and Trial data made available in MentalRiskES for the estimation of type of addiction.

	Betting	Online Gaming	Trading	Lootboxes	Total
Low risk	41	51	76	14	182
High risk	46	55	61	13	175
Total	87	106	137	27	357

The specifications for the runs involved in Task 2 are described in Table 6, with the decision rules, All and FIM, described in, respectively, (11) and (12). Again we made use of data augmentation by segment, as presented in section 3.2.

Table 6
Run settings for Task 2.

Run	Decision	Data augmentation
Run 0	All	$\mathcal{T} \cup \widetilde{\mathcal{T}_0} \cup \widetilde{\mathcal{T}_1}$
Run 1	FIM	$\mathcal{T} \cup \widetilde{\mathcal{T}_0} \cup \widetilde{\mathcal{T}_1}$
Run 2	All	$\mathcal{T} \cup \widetilde{\mathcal{T}_0}$

Table 7 presents the performance metrics for Task 2, focused only on accurate detection capability. In Task 2, we demonstrated strong detection capabilities. Our models achieved high Macro-F1 scores (0.856) , with run 0 and run 1 achieving identical performance metrics, meaning that the voting system had little relevance since run 0 and run 1 were identical expecting the voting system. Notably, our run 2 showed a slight decrease in Macro-F1 marking the importance of our data augmentation since the runs 0 and 1 were made with the augmented data from both risk classes 0 and 1 and the last run used only augmented data from the low risk class. Overall, we can appreciate our efforts, which placed us as the 3rd team in the competition rankings for this task.

Table 7
Performance metrics for each run in Task 2.

Run	Rank	Accuracy	Macro-P	Macro-R	Macro-F1
Run 0	5/32	0.900	0.928	0.850	0.856
Run 1	6/32	0.900	0.928	0.850	0.856
Run 2	7/32	0.900	0.933	0.846	0.850

4.3. Computational efficiency

In connection to the SDGs, the carbon emissions and computational efficiency of each run were also assessed. Table 8 presents our hardware configuration and emissions data. The results demonstrate exceptional efficiency together with competitive performance across both tasks.

Table 8
Hardware configuration and emissions data.

Measurement	Value
CPU Count	12
GPU Count	None
CPU Model	AMD Ryzen 5 5600X 6-Core Processor
GPU Model	N/A
RAM Total Size	15.924 GB
Average Duration	2.69E+01 ms
Energy Consumed (mean)	2.44E-04
CPU Energy (mean)	2.43E-04
Emissions (mean)	4.24E-05

4.4. Discussion

Having presented the experimental results and in an attempt to provide further discussion, we delved into evaluating both performance and resource consumption, considering the straightforward efficiency ratios in (13). Bearing in mind that for the two baselines some consumption metrics were not made available, they will not be taken into account in the following discussion as it is not possible to rank the baselines in this way.

$$\begin{aligned} \text{Eff}_{\text{RAM}} &= \frac{\text{Macro F1}}{\text{RAM Size}}, & \text{Eff}_{\text{GPU}} &= \frac{\text{Macro F1}}{\text{GPU Energy}}, \\ \text{Eff}_{\text{Energy}} &= \frac{\text{Macro F1}}{\text{Energy Used}}, & \text{Eff}_{\text{Emissions}} &= \frac{\text{Macro F1}}{\text{Emissions}} \end{aligned} \tag{13}$$

Seizing both performance and consumption, the following points are noteworthy:

- **No need of GPUs:** We were among the participants who have not made use of GPUs (aka GPU Count = None), relying solely on CPU computation. Between these contestants we ranked first in both Task 1 and Task 2 and performed at a high margin from the next approach when ranked by Eff_{GPU} (13) (being the energy consumed for non GPU users a very small number 1×10^{-13} to avoid a 0 division). Serving as an overly simplistic estimator of GPU power, the next best models having to employ graphics cards were far behind at 13/38 in Task 1 and 22/32 for Task 2. Furthermore, our CPU model (AMD Ryzen 5) is by far simpler and more economical than the others, which rely on GPUs such as the "NVIDIA L4" and "NVIDIA GeForce RTX 4090".

- **Small RAM size**: Our system required minimal RAM (15.924 GB) compared to other high-ranking teams. Overall, in terms of Macro F1-Score, the best system with rank 1/32 in Task 2, employed 31.350 GB, that is, twice as much RAM as our approach did and two top of the line NVIDIA Tesla T4. Using Eff_{RAM} (13) as the ranking criterion, our system achieved rankings of 4/38 in Task 1 and 1/32 in Task 2, while utilizing only 15.924 GB of RAM. Notably, the previously mentioned model, which ranked first in terms of Macro F1-Score, was placed 10/32 under this ranking scheme. We should highlight the disparity in computational power and resource allocation from approach to approach.

- **Energy consumption:** With an average consumption of just $2.44E-04$ our models performed pretty robustly, taking into consideration that a GPU is usually more energy efficient for large calculations than a CPU. Using a structure similar to the points before, sorting by $\text{Eff}_{\text{Energy}}$ (13) we achieved less impressive result ranking 19/38 in Task 1 and 12/32 in Task 2, highlighting the energy efficiency of a GPU against a CPU.

- **Emissions:** In terms of $\text{Eff}_{\text{Emissions}}$ (13), the results are, as expected, similar to those of energy consumption, albeit slightly better—ranking 12/38 in Task 1 and 7/32 in Task 2.

- **Trade-off between hardware and attained ranking:** Despite resource constraints, we obtained competitive results in both Task 1 and Task 2, ranking 11/38 and 5/32, respectively. In Task 2, our performance was particularly noteworthy, with evaluation metrics comparable to those of teams employing significantly more resource-intensive configurations. This positions our approach among the most effective—if not the most effective—for lightweight prediction scenarios, considering it was executed on mid- to lower-tier hardware: a system with 16 GB of RAM and an AMD Ryzen 5 5600 6-core processor, which is significantly less powerful than GPU-based setups.
- **Relevance of low risk users**: Curiously, data augmentation of low risk users had led to the biggest increment in classification performance for Task 1. The classifier seems more sensitive to gain classification ability from the low risk (labelled as 0). A conjecture to explain this might be that having more samples of the low risk class helped in having a better understanding of what is not really a high risk indicator. As a result, we submitted two variant models inferred with further data generated over the 0 class. This phenomenon did not happen on Task 2 as this classifier thrived in more data no matter if the user was high or low risk. In any case, we felt curious, and decided to submit, as well, a model trained with both data segments augmented for Task 1 and with only low risk for Task 2.

5. Conclusions

In this work we describe some approaches to deal with the early detection of gambling disorder risks and classification of addiction types proposed in MentalRiskES 2025. We restricted ourselves to approaches that could run in a regular laptop without GPUs or intensive computation resources.

We resorted to available multi-lingual approaches and employed data augmentation by segment via generative LLMs. For Task 1, the most challenging early detection scenario, we developed a hybrid approach incorporating Bi-LSTM adjusted with GroupDRO loss function and dual attention mechanisms (with both learned-risk and lexicon-risk awareness) and augmenting training data. We opted for a conservative risk detection strategy even at the expense of increasing latency with the aim to get a realistic decision strategy rather than challenge-tailored. The approach delivered competitive results with a Macro-F1 score of 0.475. For Task 2, our hierarchical Bi-LSTM architecture proved highly effective, achieving a Macro-F1 score of 0.856 and positioning us the third-best team in the competition.

As for the lessons learned, in what the impact of the artificial data obtained with generative LLMs is regarded, we observed an interesting phenomenon, that is, augmenting the low-risk class segment of data yielded the most substantial improvements. This suggests that enhancing information of what "normal" (or low-risk) behaviour is, paradoxically, improves high-risk identification, the conjecture is that clearer boundaries between classes are learned this way.

It is worth highlighting that our approach demonstrated fairly good computational efficiency, operating without GPU resources and with minimal RAM requirements (16GB) while maintaining competitive performance. This efficiency-performance balance is particularly relevant for potential real world deployment in mental health monitoring systems where computational resources may be limited.

For future work, we find several core aspects worth exploring e.g. (i) learnable decision making strategies (i.e. learnable thresholds) together with a thorough experimental framework; (ii) the impact of each attention mechanism (inferred and lexicon) and, particularly, terms that should be involved in the lexicon-attention; (iii) alternative loss functions for both tasks with systematic comparisons. The task itself, could be re-considered as multi-label given that the addiction types are not mutually-exclusive and could co-appear in real settings.

To sum up, integrating well-established LSTM architectures with current techniques as strategic data augmentation and specialized attention mechanisms has demonstrated powerful for mental health risk and addiction detection, particularly in scenarios with limited labelled data.

Acknowledgments

This work was partially funded by LOTU grant (TED2021-130398B-C22 funded by MICIU/AEI/ 10.13039/501100011033 and by the European Union NextGenerationEU/PRTR) and also by the Spanish Ministry of Science and Innovation (EDHIA PID2022-136522OB-C22) with the second author being awarded a grant from the Spanish Ministry of Education within *Formación de Profesorado Universitario* program (FPU23/01068); it has been also funded by the Basque Government (IXA IT1570-22).

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT in order to improve the writing style and perform grammar and spelling checks. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] R. K. L. Nielsen, P. Grabarczyk, Are loot boxes gambling?: Random reward mechanisms in video games, Transactions of the Digital Games Research Association 4 (2019) 171–207.

[2] H. Pitt, S. McCarthy, M. Randle, M. Daube, S. L. Thomas, Young people's views about the use of celebrities and social media influencers in gambling marketing, Health Promotion International 39 (2024) daae012. URL: https://doi.org/10.1093/heapro/daae012. doi:10.1093/heapro/daae012.

[3] S. M. Gainsbury, Online gambling addiction: the relationship between internet gambling and disordered gambling, Current addiction reports 2 (2015) 185–193.

[4] 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 2024: Early detection of mental disorders risk in Spanish, Procesamiento del Lenguaje Natural 73 (2024) 435–448.

[5] 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-Raéz, Overview of MentalRiskES at Iberlef 2023: Early detection of mental disorders risk in Spanish, Procesamiento del Lenguaje Natural 71 (2023) 329–350.

[6] 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.

[7] J. Parapar, P. Martín-Rodilla, D. E. Losada, F. Crestani, eRisk 2024: Depression, anorexia, and eating disorder challenges, in: European Conference on Information Retrieval, Springer, 2024, pp. 474–481.

[8] A. Gutiérrez-Fandiño, J. Armengol-Estapé, M. Pàmies, J. Llop-Palao, J. Silveira-Ocampo, C. P. Carrino, C. Armentano-Oller, C. R. Penagos, A. Gonzalez-Agirre, M. Villegas, Maria: Spanish language models, Procesamiento del Lenguaje Natural 68 (2021) 39–60. URL: https://api.semanticscholar.org/CorpusID:252847802.

[9] 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.

[10] X. Larrayoz, A. Casillas, M. Oronoz, A. Pérez, Mental disorder detection in Spanish: Hands on skewed class distribution to leverage training, in: IberLEF (Working Notes). CEUR Workshop Proceedings, 2024.

[11] A. M. Andreu Casamayor, Vicent Ahuir, L.-F. Hurtado, ELiRF-VRAIN at MentalRiskES 2024: Using longformer for early detection of mental disorders risk, in: IberLEF (Working Notes). CEUR Workshop Proceedings, 2024.

[12] J. Fernandez-Hernandez, H. Fabregat, A. Duque, L. Araujo, J. Martinez-Romo, UNED-GELP

at MentalRiskES 2024: Transformer-Based Encoders and Similarity Techniques for Early Risk Prediction of Mental Disorders, in: IberLEF (Working Notes). CEUR Workshop Proceedings, 2024.

[13] C. D. Păduraru, I. M. Anghelina, Early risk detection for mental health disorders: UnibucAI at MentalRiskES 2024, anxiety 88 (2024) 5.

[14] J. Lee, F. Chen, S. Dua, D. Cer, M. Shanbhogue, I. Naim, G. H. Ábrego, Z. Li, K. Chen, H. S. Vera, et al., Gemini embedding: Generalizable embeddings from Gemini, arXiv preprint arXiv:2503.07891 (2025).

[15] 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, volume 1 (long and short papers), 2019, pp. 4171–4186.

[16] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, et al., Gemini: a family of highly capable multimodal models, arXiv preprint arXiv:2312.11805 (2023).

[17] Y. Li, Y. Zhang, C. Wang, Z. Zhong, Y. Chen, R. Chu, S. Liu, J. Jia, Mini-gemini: Mining the potential of multi-modality vision language models, arXiv preprint arXiv:2403.18814 (2024).

[18] G. AI, Understand and count tokens, 2025. URL: https://ai.google.dev/gemini-api/docs/tokens?lang=python.

[19] Google, Gemini api: Documentation for developers, 2025. URL: https://ai.google.dev/gemini-api/docs.

[20] H. Sak, A. Senior, F. Beaufays, Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition, 2014. URL: https://arxiv.org/abs/1402.1128. arXiv:1402.1128.

[21] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, B. Xu, Attention-based bidirectional long short-term memory networks for relation classification, in: Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers), 2016, pp. 207–212.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2023. URL: https://arxiv.org/abs/1706.03762. arXiv:1706.03762.

[23] Y. Zou, T. Gui, Q. Zhang, X. Huang, A lexicon-based supervised attention model for neural sentiment analysis, in: E. M. Bender, L. Derczynski, P. Isabelle (Eds.), Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 868–877. URL: https://aclanthology.org/C18-1074/.

[24] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Y. W. Teh, M. Titterington (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, volume 9 of *Proceedings of Machine Learning Research*, PMLR, Chia Laguna Resort, Sardinia, Italy, 2010, pp. 249–256. URL: https://proceedings.mlr.press/v9/glorot10a.html.

[25] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. URL: https://arxiv.org/abs/1711.05101. arXiv:1711.05101.

[26] P. Contributors, Reducelronplateau — pytorch 2.7 documentation, https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html, 2025.

[27] J. C. Duchi, H. Namkoong, Learning models with uniform performance via distributionally robust optimization, The Annals of Statistics 49 (2021) 1378–1406.

[28] 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. URL: https://arxiv.org/abs/1911.08731. arXiv:1911.08731.

[29] C. Eastwood, A. Robey, S. Singh, J. von Kügelgen, H. Hassani, G. J. Pappas, B. Schölkopf, Probable domain generalization via quantile risk minimization, in: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (Eds.), Advances in Neural Information Processing Systems, volume 35, Curran Associates, Inc., 2022, pp. 17340–17358. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/6f11132f6ecbbcafafdf6decfc98f7be-Paper-Conference.pdf.

[30] S. Seo, J.-Y. Lee, B. Han, Unsupervised learning of debiased representations with pseudo-attributes,

in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 16742–16751.

[31] S. Merity, N. S. Keskar, R. Socher, Regularizing and optimizing lstm language models, 2017. URL: https://arxiv.org/abs/1708.02182. arXiv:1708.02182.

[32] A. Mao, M. Mohri, Y. Zhong, Cross-entropy loss functions: Theoretical analysis and applications, 2023. URL: https://arxiv.org/abs/2304.07288. arXiv:2304.07288.

[33] P. Álvarez-Ojeda, M. V. Cantero-Romero, A. Semikozova, A. Montejo-Ráez, The PRECOM-SM Corpus: Gambling in Spanish Social Media, in: Proceedings of the 31st International Conference on Computational Linguistics, 2025, pp. 17–28.

[34] J. Á. González-Barba, L. Chiruzzo, S. M. Jiménez-Zafra, Overview of IberLEF 2025: Natural Language Processing Challenges for Spanish and other Iberian Languages, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2025), co-located with the 41st Conference of the Spanish Society for Natural Language Processing (SEPLN 2025), CEUR-WS. org, 2025.

[35] A. M. Mármol-Romero, P. Álvarez-Ojeda, 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 2025: Early Detection of Mental Disorders Risk in Spanish, Procesamiento del Lenguaje Natural 75 (2025).

A. Prompt used for data augmentation

```
Act as a data augmentation expert. Create a different variation of the following text while:

1. Preserving the original meaning and key information
2. Using different sentence structures and synonyms
3. You can change the formality of the text and the tone while maintaining the same meaning
4. Keeping the same emotional content and sentiment
5. Ensuring the text remains natural and fluent
6. Messages are in Spanish, take this into account
7. You can use emojis to make the text more friendly and natural
8. Avoid using the same words or phrases as the original text
9. Avoid using the same sentence structure as the original text
10. Avoid using the same punctuation as the original text
11. You can make orthographical mistakes as this would be written by a human
12. You can use slang to make the text more friendly and natural
13. You can make grammatical mistakes as this would be written by a human
14. Remember the text is written by a human, so it can have mistakes
15. Do not always commit the same mistakes, try to make it different each time
16. Your main goal is doing data augmentation REMEMBER THIS

This text is classified as {classification} (where 1 indicates HIGH risk content and 0
    indicates LOW risk).
For mental health-related text, ensure that clinical significance and risk indicators are
    preserved.

Format your response as a JSON array with the same structure as your input.

The original text that you shall modify:
```