COTECMAR-UTB at eRisk 2025: Semantic-Centroid Symptom Ranking and Early Depression Detection using Adaptive Decision Rule*

Notebook for the eRisk Lab at CLEF 2025

Luis Mendoza[1,2,*], Joan Suarez[1,2,*], Edwin Puertas[1], Juan Martinez[1] and Jairo Serrano[1]

[1]Universidad Tecnológica de Bolívar (UTB), Bolívar, Cartagena D.T. y C., Colombia
[2]Corporación de Ciencia y Tecnología para el Desarrollo de la Industria Naval, Maritima y Fluvial (COTECMAR), Bolívar, Cartagena D.T. y C., Colombia

Abstract

Depression remains a major global health concern, with millions of people affected in various demographics. However, the timely detection of depression symptoms remains a challenge due to biases and limitations in traditional diagnostic methods. Social networks have become valuable sources for identifying early signs of depression, as they provide real-time user interactions that reflect emotional states. This paper explores the eRisk 2025 challenge, focusing on two primary tasks for early detection of depression in online conversations. Task 1 involves ranking sentences according to their relevance to depression symptoms, while task 2 addresses the analysis of emotional progression in real-time conversations. We apply Natural Language Processing (NLP) models, including Transformer architectures such as BERT, to capture semantic nuances in text. Furthermore, our methodology incorporates a Classifier with partial information (CPI) and a Decision Moment Classifier (DMC) to track emotional shifts over time, providing a framework for detecting depression risks early in conversational contexts. We present a comprehensive evaluation of our approach, discussing its challenges, successes, and potential for future improvements in early detection systems.

Keywords

Depression detection, early intervention, social media analysis, Natural Language Processing (NLP), Transformer models, BERT, emotional progression, Decision Moment Classifier (DMC), Classifier with Partial Information (CPI),

1. Introduction

Depression remains one of the leading causes of disability worldwide, affecting millions of individuals across various age groups and socio-economic backgrounds. Despite its high prevalence, a significant proportion of individuals suffering from depressive symptoms do not receive adequate treatment, often due to barriers such as social stigma, limited access to mental health services, and underdiagnosis [1]. Traditional diagnostic methods based on clinical assessments and self-reporting are often subject to biases and resource constraints, making it increasingly difficult to detect depression at an early stage. Consequently, the need for innovative approaches to mental health diagnostics has become more pressing.

In recent years, the rise of social media platforms, such as Reddit, Quora, Facebook, and so on, has opened new avenues for detecting early signs of depression. These platforms, where individuals share their thoughts and emotional states in real time, provide rich sources of data that can be analyzed to identify psychological distress [2]. The application of Natural Language Processing (NLP) and Machine Learning (ML) techniques to these data has led to the development of automated systems for detecting depression and other mental health issues, potentially offering timely intervention [3, 4].

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain
luimendoza@utb.edu.co (L. Mendoza); jloaiza@utb.edu.co (J. Suarez); epuertas@utb.edu.co (E. Puertas); jcmartinezs@utb.edu.co (J. Martinez); jserrano@utb.edu.co (J. Serrano)
https://edwinpuertas.github.io/ (E. Puertas)
0009-0001-4660-4892 (L. Mendoza); 0009-0007-7874-8261 (J. Suarez); 0000-0002-0758-1851 (E. Puertas); 0000-0003-2755-0718 (J. Martinez); 0000-0001-8165-7343 (J. Serrano)

This paper focuses on the eRisk 2025 challenge, which aims to advance early risk detection systems for mental health through NLP and AI techniques. The challenge consists of two main tasks that address different aspects of depression detection in online conversations: *Task 1*, which involves ranking sentences based on their relevance to specific depression symptoms as outlined in the Beck Depression Inventory-II (BDI-II) questionnaire, and *Task 2*, which focuses on tracking the emotional progression of individuals in real-time, identifying subtle shifts in emotional states that may indicate worsening depression [5, 6].

The complexity of detecting depression from online texts arises from the nuanced and varied nature of depressive symptoms, which can manifest differently between individuals. To address this challenge, this work integrates state-of-the-art NLP models, including Transformer-based architectures such as BERT and its variants, to better capture the contextual and semantic meaning of text.

In addition, the task of identifying depression symptoms in sequential conversational data, as in Task 2, introduces additional complexity. Unlike static data, conversational posts often involve incremental changes in emotional state, requiring dynamic models capable of understanding emotional progression over time.

The contributions of this paper include the following.

- A detailed exploration of the eRisk 2025 challenge, highlighting the significance of early detection of depression through social media analysis.
- A robust evaluation of our methodology for both Task 1 and Task 2, demonstrating the challenges and successes of applying NLP techniques to noisy data in the real world.
- A decision-making component that tracks emotional shifts over time, providing a framework for early and continuous risk detection in conversational contexts.

2. Related Works

Task 1 and Task 2 of the eRisk 2025 challenge have driven the development of numerous methodologies for the detection of depression on social networks, using cutting-edge natural language processing (NLP) and machine learning (ML) techniques. These tasks have significantly influenced the state-of-the-art in assessing depression symptoms and early detection in conversational interactions.

Task 1 focuses on ranking sentences from social media posts according to their relevance to the 21 symptoms of depression outlined in the Beck Depression Inventory-II (BDI-II) [7]. Transformer-based models, such as BERT and its variants, such as RoBERTa and Distilbert, have become the dominant approach for this task. These models generate sentence embeddings that capture contextualized information, which can be used to rank sentences based on their relevance to specific depression symptoms, such as sadness, pessimism, and agitation. The ability of Transformer models to encode semantic relationships between words has significantly improved performance in depression symptom detection, making them a powerful tool in the task of ranking relevant sentences [8, 9].

Another sort of strategy used in Task 1 was the usage of synthetic data generated by Large Language Models (LLMs) like GPT-O3-mini and ChatGPT 4o. These models augment existing datasets by generating diverse and semantically rich examples of depression-related sentences for each symptom in the BDI-II. This helps to enhance the training data, increasing its diversity and enabling models to learn more nuanced patterns of depression expression in online text [10]. Furthermore, this was also implemented with another sort of approach based on using early maladaptive schema (EMS), which was adopted to achieve an enrichment of the ranking dataset [6].

Task 2 addresses the early detection of depression in conversational interactions, with the goal of predicting the likelihood of depression from a user's posts in social media threads. This task typically involves both binary classification and regression approaches. Models that combine various types of data, such as lexical features, phonetic embeddings, and syntactic patterns, have shown great promise. SVM (Support Vector Machines) models have been used to detect depression by incorporating these multimodal inputs, which capture emotional expression, self-referential language, and interpersonal dynamics in user interactions. Similarly, LSTM (Long Short-Term Memory) networks [11], which are

a type of recurrent neural network, have been applied to this task as well. These models learn from sequential data, capturing temporal dependencies that are crucial for identifying patterns indicative of depression in conversations.

In the context of early risk detection (ERD), the problem is addressed using the CPI (Classification with Partial Information) and DMC (Decision Making Component) framework [12]. The CPI model classifies users as at risk for depression based on the partial information available in their posts. The DMC component, on the other hand, is responsible for deciding when to raise an alarm, determining the optimal moment for classification by weighing the risk of false negatives against the need for timely intervention. These components work together to identify and alert users at risk as early as possible, balancing precision and recall in a real-world scenario.

3. Task 1: Search for Symptoms of Depression

In Task 1 of the eRisk 2025 challenge, the goal is to classify and rank sentences from a dataset of social media posts based on their relevance to the 21 symptoms of depression in the Beck Depression Inventory-II (BDI-II) [13, 14]. The aim is to develop models to identify and rank sentences that best represent each depression symptom. The rankings are evaluated using metrics such as Mean Average Precision (MAP), Recall Precision (R-Prec), Precision at 10 (P@10), and Normalized Discounted Cumulative Gain (NDCG), which assess how accurately the models prioritize relevant content.

3.1. Pipeline Explanation

The initial challenge lay in designing an appropriate structure for integrating the available data. This involved harmonising the `.trec` formatted files with their corresponding classification files, which were annotated under two distinct evaluation schemes: *unanimity* and *majority vote*.

Considering the data structure, a cleaning and filtering process was done on both datasets corresponding to the 2023 and 2024 sets. This stage was imperative to ensure consistency in data quality, format, and labelling prior to model development.

After organising and cleaning the dataset, we proceeded to train various models using different learning approaches. First, we employed classic machine learning techniques with the `PyCaret` framework; then, we explored deep learning methods using `PyTorch` and Transformer-based architectures.

In the next stage, attention was turned to the processing of the 2025 evaluation set, which resembles the structural design of the 2024 data set. At this point, the focus was on filtering the data in preparation for symptom-level classification.

In this context, filtering refers to the application of the VADER algorithm -*Valence Aware Dictionary and sEntiment Reasoner* - as a preprocessing step to identify texts that exhibit negative sentiment. This selection criterion allowed us to isolate samples that were considered most likely to reflect psychological distress, providing a more relevant subset for classification and scoring, an issue that was subsequently addressed by two alternative modeling strategies.

An overview of the workflow is presented in Figure 1, which illustrates the core stages of data integration, such as cleaning, filtering, and running models in the machine learning and deep learning paradigms.

3.2. Dataset overview

The initial step involved reading and examining the accumulated text data stored in .TREC format. This preliminary phase allowed us to understand the data structure we would be working with throughout the challenge.

3.2.1. Understanding the Document Structure

The structure of the files provided for 2023 will be explained below using a sample.

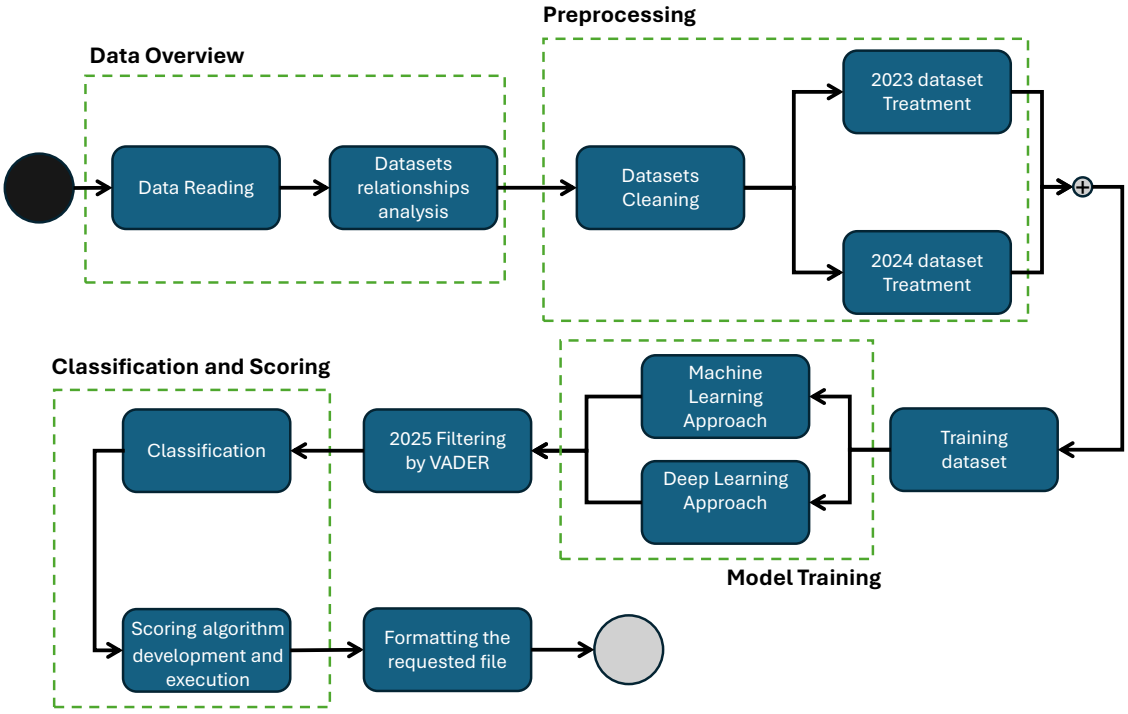


Figure 1: Diagram of the processing pipeline, from raw data ingestion through filtering, modelling, and evaluation.

```
<DOC>
    <DOCNO>s_480_0_0</DOCNO>
    <TEXT>It came out in 1990 lol.</TEXT>
</DOC>
<DOC>
    <DOCNO>s_480_1_0</DOCNO>
    <TEXT>I didn't say he wasn't charismatic.</TEXT>
</DOC>
```

In the data set for 2024 was also included a thread-like structure comprising a central message and reply sequences similar to those found on platforms such as Quora, Reddit, or Facebook. An example is shown below:

```
<DOC>
    <DOCNO>99001_0_0</DOCNO>
    <PRE></PRE>
    <TEXT>My high school math teacher's rule was that we had to
    show all the steps...</TEXT>
    <POST>The exception to this rule was steps could be skipped if
    she was able...</POST>
</DOC>
<DOC>
    <DOCNO>99001_0_1</DOCNO>
```

```
    <PRE></PRE>
    <TEXT>The exception to this rule was steps could be skipped
    if she was able...</TEXT>
    <POST>I got a zero on a homework assignment because I just
    wrote all the answers...</POST>
</DOC>
```

Understanding the predecessor–response structure (PRE and POST) was key to extracting and organising text entries during preprocessing. This same structure is present in the 2025 dataset, hence, making an efficient, non-redundant parsing method, was regarded primary goal.

3.2.2. Structure of Classification Files

Alongside the text corpora, classification files were provided for each dataset, formatted as follows:

```
query   q0   docid              rel
1       0    s_405_1279_1       1
```

This structure was interpreted as:

- **query:** Identifier referring to one of the 21 BDI-II symptom attributes.
- **q0:** A fixed value (0), not used further in processing.
- **docid:** Document identifier following the pattern `s_X_Y_Z`, where:
  - `s_X` denotes a conversation or discussion thread.
  - `Y` refers to a subgroup within that thread.
  - `Z` indicates the specific phrase within the subgroup.
- **rel:** Relevance label; 1 for relevant, 0 for non-relevant.

Two versions of the classification data were provided, based on different annotation philosophies: one called *majority vote*, and the other *unanimous agreement*. Upon review [5], it was found that although the documents retained the same symptom label under both schemes, their relevance annotations sometimes differed.

This discrepancy typically arose from two categories of text:

1. Texts describing events or states associated with someone other than the author. For example: *"I believe my aunt has been experiencing similar stress due to her previous job."*
2. Texts expressing opinions or reflections on events that are emotionally neutral or not personally impactful. For instance:
   "I found it sad that the film ended with the protagonists separated."

In light of these observations, the decision was to prioritise the use of texts labelled as relevant only under the unanimous agreement criterion. This ensured a more contextually grounded training dataset, favoring examples where the user's emotional involvement was clearer and more likely to indicate depressive symptoms.

3.2.3. Relevance Discrepancy Statistics

An audit of the datasets revealed notable numbers of disagreements between majority and unanimous labels:

- For 2023, a total of **2,348** texts showed disagreement in relevance. Examples include:
  `DocID: s_975_61_2  Symptom: 1  Majority: 1  Consensus: 0`
  `DocID: s_993_582_1  Symptom: 14  Majority: 1  Consensus: 0`

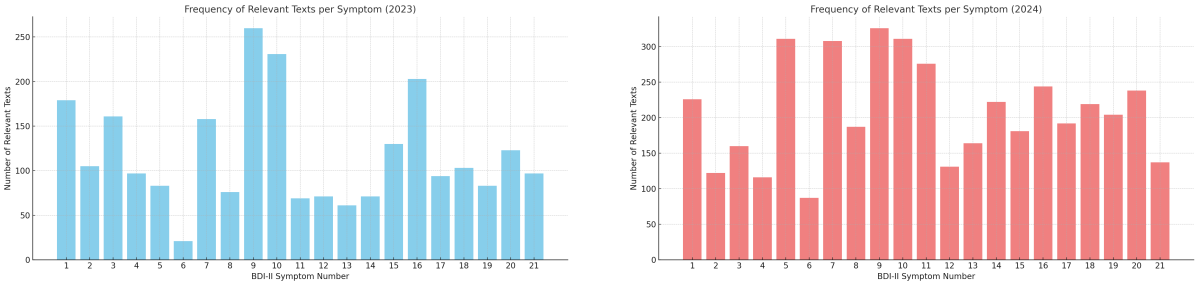
- For 2024, a total of **2,531** texts exhibited similar discrepancies. Examples include:
  `DocID: 49519_0_1  Symptom: 1  Majority: 1  Consensus: 0`
  `DocID: 414454_0_17  Symptom: 1  Majority: 0  Consensus: 1`

These discrepancies highlighted the importance of filtering and focusing on a single annotation paradigm — in this case, the one based on unanimous consensus. Furthermore, as shown below, a statistical analysis of each dataset was carried out to assess whether the usage of balancing and compensation techniques was necessary or worth considering.

3.3. Class Distribution and Data Balancing Methodology

Prior to the implementation of any regularisation strategies, a statistical analysis was conducted to evaluate the distribution of symptom classes within the datasets for the years 2023 and 2024.

This assessment exclusively considered texts deemed relevant under the unanimous annotation criterion, as previously discussed. A summary of the number of symptom-annotated texts for each year is illustrated in Figure 2.



(a) Frequency of relevant texts per symptom (2023)

(b) Frequency of relevant texts per symptom (2024)

Figure 2: Number of texts classified by BDI-II symptom under the unanimous annotation scheme for the 2023 and 2024 datasets.

The analysis revealed a pronounced imbalance across the symptom classes. In particular, considering 2023 data set, symptoms 6, 11, and 13 were markedly underrepresented, whereas symptoms 9 and 10 appeared in disproportionately high quantities.

Descriptive statistics indicated a mean of 117.9 texts per symptom, with a standard deviation of 60.43. The interquartile range was defined by the first quartile (*Q1*) at 76 and the third quartile (*Q3*) at 158. Given this distribution, the 75th percentile (*Q3*) was chosen as the reference threshold for balancing class frequencies.

3.3.1. Balancing Strategy for the 2023 Dataset

To mitigate the imbalance within the 2023 dataset, a two-stage augmentation strategy was adopted for minority classes. Initially, the Easy Data Augmentation (EDA) technique was applied to generate approximately 30% of the additional samples required. This included operations such as synonym replacement and adverb insertion, following the methodology introduced in [15].

The remaining 70% of the augmented data was generated using prompt-based synthesis with the `o3-mini` language model. Prompts were constructed following an EMS + BDI-II schema, enabling the model to produce semantically coherent and symptom-aligned synthetic sentences. This methodology draws inspiration from [6], which employed Early Maladaptive Schemas for sentence generation.

3.3.2. Balancing Strategy for the 2024 Dataset

For the 2024 dataset, where the degree of class imbalance was less extreme, the Synthetic Minority Oversampling Technique (SMOTE) was employed to interpolate new samples for underrepresented classes.

Simultaneously, classes exceeding the *Q3* threshold were reduced through a targeted undersampling procedure. Rather than applying random removal, a semantic preservation strategy was adopted to retain the most representative samples. Specifically, TF-IDF vectorisation was used to compute centroid vectors for each class. The cosine similarity of each text to its class centroid was then calculated and only those samples with the highest similarity values, indicating maximum semantic representativeness, were retained.

This approach sought to ensure that core linguistic features for each symptom category were preserved within the reduced subsets, avoiding the random elimination of important samples.

3.3.3. Final Training Corpus

Following completion of the augmentation and reduction procedures, the balanced datasets for 2023 and 2024 were merged to form a unified training corpus. This final dataset contained approximately 450 samples per symptom, organised in a tabular format with two primary columns: one for the text content and another specifying the corresponding BDI-II symptom label.

3.4. Machine Learning Modelling and Performance Evaluation

Following the text preprocessing phase—including the removal of stopwords and non-informative tokens—the modelling stage was initiated with the aim of developing a classifier capable of recognising depressive symptoms and assigning texts to their corresponding BDI-II category.

Two distinct modelling paradigms were explored. The first focused on traditional machine learning techniques, implemented using the `PyCaret` and `Scikit-learn` libraries; the second centred on deep learning approaches employing `PyTorch` and Transformer-based architectures. For the classical pipeline (machine learning approach), a compact yet informative textual representation was used, namely Term Frequency–Inverse Document Frequency (TF-IDF), limited to 5,000 features and configured with bi-grams (`max_features=5000`, `ngram_range=(1,2)`), as suggested in [8].

The `PyCaret` framework enabled rapid experimentation across a broad suite of supervised learning algorithms using standardised pipelines and evaluation metrics. A total of eight classifiers were assessed: Ridge Classifier, Support Vector Machine (linear kernel), Logistic Regression, Random Forest, Decision Tree, *k*-Nearest Neighbours, Naïve Bayes, and AdaBoost. Each model was trained using five-fold cross-validation, and performance was primarily evaluated using the F1-score, as summarised in Table 1.

Table 1
Model comparison based on 5-fold cross-validation results.

Model	Accuracy	Precision	Recall	F1-score
Ridge Classifier	0.9221	0.9237	0.9221	**0.9220**
SVM (Linear Kernel)	0.9195	0.9216	0.9195	0.9194
Logistic Regression	0.9130	0.9150	0.9130	0.9130
Random Forest	0.8987	0.9004	0.8987	0.8985
Decision Tree	0.8561	0.8604	0.8561	0.8566

Among the models tested, the Ridge Classifier yielded the highest performance, achieving an F1-score of 0.9220, along with high precision and recall.

Subsequently, the Ridge Classifier was further refined through hyperparameter optimisation using `GridSearchCV`, a module within Scikit-learn that performs an exhaustive search across a predefined parameter grid, employing cross-validation to ensure generalisability. This optimisation procedure led to an enhanced final F1-score of 0.9283.

The learning dynamics of the final model, including its convergence trajectory, were visualised as shown in Figure 3, the learning curve demonstrated steady performance improvements and no

indications of overfitting.

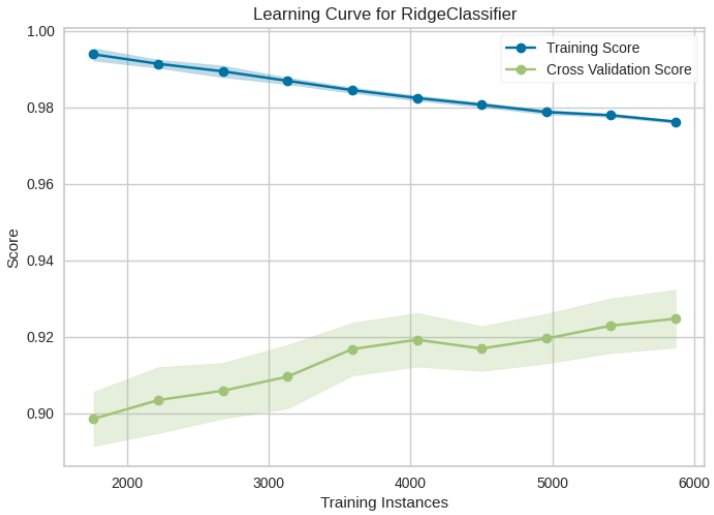


Figure 3: Learning curve of the final Ridge Classifier model visualised using PyCaret, illustrating stable convergence.

3.5. Deep Learning Approach: Fine-Tuning Transformer-Based Models

The second modelling strategy explored a deep learning approach grounded in pre-trained Transformer architectures, with the aim of harnessing contextual embeddings to improve classification performance. This methodology is based on contemporary advancements in transfer learning, whereby models trained on large-scale language corpora are adapted to specific downstream tasks through fine-tuning [16].

The selected backbone was `distilbert-base-uncased`, a distilled variant of BERT designed to retain much of its representational power while significantly reducing model size and computational demand [17]. This model was integrated via the Hugging Face `transformers` library. Input encoding was handled using the `DistilBertTokenizerFast`, offering efficient subword tokenisation aligned with the base model's vocabulary.

To tailor the architecture for multi-class classification, the model's output head was replaced with a fully connected linear layer whose output dimension corresponded to the number of BDI-II symptom classes.

Training was conducted using the cross-entropy loss function, paired with the Adam optimiser. The optimiser dynamically adjusted learning rates for each parameter based on estimates of the first and second moments of gradients. During training, the randomly initialised output layer weights and selected intermediate Transformer layers were updated via backpropagation.

The dataset was partitioned as follows:

- **Training set:** 6,515 texts
- **Validation set:** 1,400 texts
- **Test set:** 1,407 texts

The fine-tuning process spanned three epochs, with training and validation metrics monitored on a per-batch basis. Initial results during the first epoch showed a training accuracy of 95.35% and validation accuracy of 89.43%. By the final epoch, performance increased to 97.51% accuracy on the training set, 91.21% on the validation set, and 91.90% on the held-out test set.

While these results were consistent with those obtained using classical machine learning methods, it is noteworthy that the deep learning model required more computational resources and longer training times. Interestingly, the optimised Ridge Classifier from the traditional pipeline achieved comparable—and in some instances superior— F1-scores, while maintaining greater computational

efficiency and faster inference speed. This may be partially attributed to the relatively short length of the texts in the dataset.

Given the balance between performance gain and resource constraints, training over three epochs was considered adequate for this iteration. Nonetheless, Transformer-based architectures remain promising and merit further exploration in future studies.

3.6. Semantic Scoring Methodologies for BDI-II Symptoms

To assess the degree of semantic similarity between a given input phrase and a predefined set of symptom-related expressions, two complementary methodologies were employed. These approaches are grounded in contextual embeddings and make usage of established transformer-based models for sentence representation.

3.6.1. Contextual Similarity via Centroid Embeddings

The first method relies on comparing an individual phrase against a class-level semantic centroid, which acts as a condensed representation of prototypical expressions for a given symptom class. These centroids were computed directly from the real training dataset, which consists of the merged datasets from 2023 and 2024.

The objective was to assign a continuous score to each input phrase x, scaled in the range $[0, 10]$, which quantifies its semantic resemblance to a particular BDI-II symptom, as inferred from training data.

Formal Notation Let:

- $\mathcal{S} = \{1, 2, \ldots, 21\}$ be the set of BDI-II symptom classes.
- $D = \{(x_i, y_i)\}_{i=1}^{N}$ the training dataset, where $x_i \in \mathbb{R}^d$ represents a sentence embedding ($d = 384$ as defined by 'all-MiniLM-L6-v2'), and $y_i \in \mathcal{S}$ the corresponding symptom class.
- $m(\cdot)$ the sentence embedding function provided by 'sentence-transformers'.
- $\hat{s}$ the predicted symptom class for a new input x, according to the trained classifier $f(x)$.

Semantic Centroid Definition For each class $s \in \mathcal{S}$, its semantic centroid was defined as the mean embedding of all training samples belonging to that class:

$$\mu_s = \frac{1}{|D_s|} \sum_{(x_i, y_i) \in D_s} m(x_i) \tag{1}$$

where $D_s = \{x_i \mid y_i = s\}$ denotes the set of samples associated with class s.

Score Computation for New Inputs Given a new input phrase x, its predicted class $\hat{s}$ is obtained via the trained classifier (both models defined formerly). The final score was then computed as the cosine similarity between its embedding and the centroid of the predicted class:

$$\text{Score}(x, \hat{s}) = 10 \cdot \cos\left(m(x), \mu_{\hat{s}}\right) \tag{2}$$

with cosine similarity defined as:

$$\cos(u, v) = \frac{u \cdot v}{\|u\| \cdot \|v\|} \tag{3}$$

The resulting score, scaled to $[0, 10]$, served as an interpretable indicator of symptomatic relevance. The pseudocode presented below illustrates the logic implemented to rank texts based on this criterion.

```
Input:
- Dataset D = {(x_i, y_i)}
- New input phrase x
- Trained classifier f(x)
- Embedding model m(.)

Step 1: Group phrases by class s in S
Step 2: For each class s:
            Compute mu_s = mean of m(x_i) where y_i = s
Step 3: Predict class: s_hat = f(x)
Step 4: Get embedding of x: v_x = m(x)
Step 5: Score = 10 * cosine_similarity(v_x, mu_s_hat)
Output: Score in [0, 10]
```

3.6.2. Semantic Score via Symptom-Based Exemplars

In contrast to the centroid-based method, the second approach employs a hybrid set of real and synthetic exemplars to estimate semantic similarity. The synthetic exemplars were generated using the OpenAI model O3-mini, this time leveraging a technique known as Generated Knowledge Prompting, as introduced by Liu et al. in [18].

The Semantic Score via Exemplars method assigns to each input phrase x a continuous score in $[0, 10]$, measuring its semantic closeness to a curated set of first-person exemplar sentences for each BDI-II symptom.

Formal Notation Let:

- $\mathcal{S} = \{1, 2, \ldots, 21\}$ denote the BDI-II symptom classes.
- $\mathcal{X}_s = \{x_s^{(1)}, x_s^{(2)}, \ldots, x_s^{(k)}\}$ the set of handcrafted or generated exemplars for class $s \in \mathcal{S}$.
- $m(\cdot)$ the embedding function for natural language input.
- $\hat{s}$ the class prediction for the input phrase x as determined by the classifier $f(x)$.

Score Computation Using Exemplars The score is computed as the average scaled cosine similarity between the embedding of the input phrase and the embeddings of the k exemplars in the predicted class $\hat{s}$:

$$\text{Score}(x, \hat{s}) = \frac{10}{|\mathcal{X}_{\hat{s}}|} \sum_{j=1}^{|\mathcal{X}_{\hat{s}}|} \left(\frac{1 + \cos\left(m(x), m(x_{\hat{s}}^{(j)})\right)}{2} \right) \tag{4}$$

where $\cos(\cdot, \cdot)$ is defined as before. The following pseudocode illustrates the logic applied to rank the texts according to the equation presented above.

```
Input:
- New input phrase x
- Trained classifier f(x)
- Embedding model m(.)
- Dictionary of exemplars: exemplars_dict[symptom_id] = [phrases]

Step 1: Predict class: s_hat = f(x)
Step 2: Get embedding of x: v_x = m(x)
Step 3: For each exemplar x_j in exemplars_dict[s_hat]:
```

```
            Compute v_j = m(x_j)
            Compute sim_j = cosine_similarity(v_x, v_j)
            Scale sim_j to [0, 10]: sim_scaled_j = 10 * (1 + sim_j)
                / 2
Step 4: Score = average(sim_scaled_j)
Output: Score in [0, 10]
```

3.7. Final Dataset Processing and Evaluation Strategies

After examining the 2025 test dataset, it was determined that it comprised approximately 17,384,603 texts. Given the computational power required to classify such a scale of data, a filtering mechanism was introduced to reduce the volume of data processed. Specifically, we employed a pre-filtering step designed to identify only those texts that exhibited a generally negative sentiment. To this end, the VADER sentiment analysis tool (Valence Aware Dictionary and sEntiment Reasoner) was applied [19], resulting in a reduced subset of 5,430,903 texts. This approach is consistent with prior methodologies adopted in eRisk 2023 [9], where sentiment filtering effectively reduced the search space while preserving meaningful signal.

Having established the final evaluation corpus, the subsequent stage involved the application of previously developed classification models and semantic scoring methodologies. Three distinct configurations were implemented to generate the final results:

1. A **Ridge-classifier**, utilising the *semantic centroid similarity* strategy to assign scores to each input phrase.
2. A second **Ridge classifier**, this time employing the *exemplar-based semantic similarity* method. Notably, this configuration demonstrated more optimistic scoring behaviour and thus was deemed promising for further experimentation.
3. A **deep learning classifier**, specifically fine-tuned on relevant training data, combined with the *exemplar-based scoring* methodology.

3.8. Reflection on Evaluation Results

The participation of COTECMAR-UTB in Task 1 of eRisk 2025 yielded an intermediate performance outcome when benchmarked against the top-performing systems in the challenge. The submitted run, titled *ranked updated*, was the best among the three submissions; it employed a Ridge classifier in conjunction with an exemplar-based semantic similarity scoring mechanism. This run achieved 10th place overall in the final ranking of participating systems.

According to the official evaluation results, the COTECMAR-UTB system attained the following scores under the *unanimity*-based assessment framework: AP = 0.042, R-Precision = 0.108, P@10 = 0.181, and NDCG = 0.243, as shown in Table 2. These figures positioned the team within the middle tier of the ranked systems, showing potential yet leaving notable room for improvement. For comparison, the same model evaluated under the *majority vote* scheme achieved AP = 0.077, R-Precision = 0.165, P@10 = 0.414, and NDCG = 0.290.

The rationale for emphasising the results under the unanimity scheme lies in its heightened sensitivity to semantic and contextual precision. As discussed in preceding sections, the exemplar-based scoring strategy is designed to capture fine-grained textual alignment with prototypical symptom expressions—an alignment that is more rigorously evaluated under the unanimity setting, where all expert assessors must agree on a document's relevance.

A comparative inspection reveals that leading teams—such as INESC-ID and UET-Psyche-Warriors—achieved substantially higher metrics (e.g., INESC-ID with AP = 0.269, P@10 = 0.509, and NDCG = 0.561), demonstrating stronger alignment between sentence relevance scores and expert annotations.

Table 2
Comparison of evaluation metrics under the unanimity annotation scheme.

Team	AP	R-Prec	P@10	NDCG
INESC-ID	0.269	0.383	0.509	0.561
UET-Psyche-Warriors	0.248	0.330	0.476	0.577
SonUIT	0.223	0.303	0.462	0.545
PJs-team	0.188	0.311	0.452	0.446
BGU-Data-Science	0.171	0.272	0.419	0.489
COTECMAR-UTB	**0.042**	**0.108**	**0.181**	**0.243**

While the exemplar-based approach adopted proved to be promising—particularly due to its interpretability and synthetically enriched exemplars—several factors likely contributed to the observed performance gap:

- **Sparse representation coverage:** The use of a limited set of exemplars per symptom, even when generated synthetically, may have failed to sufficiently encapsulate the broad diversity of linguistic expressions associated with depressive symptoms.
- **Context neglect:** Although the system operated on sentence-level embeddings, the task allowed for adjacent sentence context.
- **Model depth and learning capacity:** While Ridge classifiers offer efficiency, they may be outperformed by deeper models (e.g., fine-tuned Transformers) in capturing subtle nuances of language. Incorporating deep contextual encoders with attention mechanisms could enhance the precision of symptom detection.
- **Score calibration:** The cosine similarity score, although scaled, might not have been fully aligned with the thresholds for relevance set by human assessors.

4. Task 2: Contextualized Early Detection of Depression

The eRisk 2025 Task 2 introduces a new approach for early detection of depression symptoms by analyzing user interactions on social media platforms [14]. This task differs from previous ones in that it focuses on understanding the contextual nature of conversations. Rather than simply evaluating isolated messages, the objective is to detect patterns of behavior across entire conversations, accounting for both the temporal and emotional evolution of the user's posts and comments. This makes the task particularly challenging, as it requires capturing the subtle emotional shifts that may signal the onset of depression, even when these shifts occur gradually over time.

Building on previous editions of the eRisk Challenge, where early detection systems were applied to detect risks such as pathological gambling [20], we chose to implement a Contextualized Early Detection System using a combination of two models: CPI (Classifier with Partial Information) and DMC (Decision Moment Classifier). This architecture has been applied in past challenges, where it proved effective in detecting early signs of depression and other behavioral risks based on users' interactions [12].

The CPI model for our solution is built using a long-short-term memory (LSTM) network. LSTMs are well suited for sequential data, as they can capture long-range dependencies in time-series data. In our case, the model processes the messages exchanged by a user, considering the partial information available at each moment. This allows the CPI model to classify whether a user's message is indicative of depression, using only the information available up to that point in the conversation.

On the other hand, the DMC model adopts a fixed policy model approach, designed to assess the emotional progression of the user over time. This model does not classify individual messages, but instead tracks the trend of emotional content across multiple messages. By evaluating the overall

emotional trajectory of the conversation, the DMC determines if the user's emotional state is improving or worsening. This allows the system to make a decision about whether the user needs additional intervention or support.

The combination of the CPI and DMC models helps us create a system that not only classifies individual messages, but also takes into account the contextual evolution of the conversation, leading to a more accurate early detection of depression symptoms. This system can identify emotional changes over time, even when individual messages do not explicitly indicate a risk of depression.

The Figure 4 illustrates the system pipeline used in our solution. It details the sequential flow of data and the various stages of processing, from data reading and text cleaning to the creation of batches, re-labeling with VADER, vectorization with Word2Vec, and model training. Each stage plays a crucial role in ensuring that the system can detect depression symptoms accurately and efficiently. The pipeline will be explained in detail in the next section.

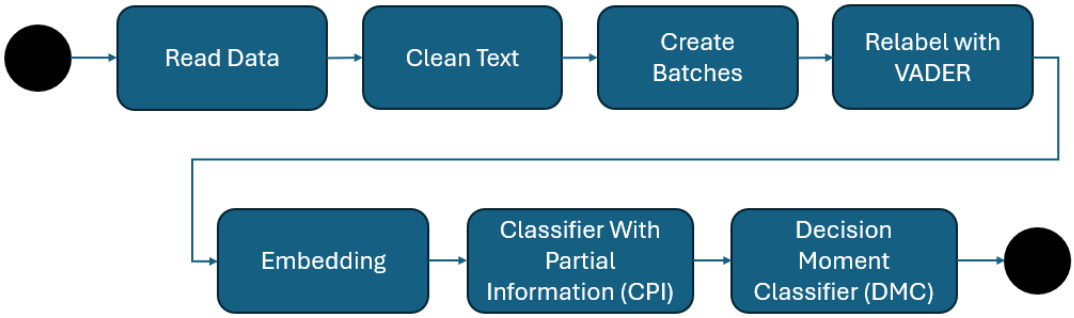


Figure 4: System pipeline

4.1. Architecture explanation

The architecture for the early detection of depression involves several key stages, from reading and preprocessing the data to training the model and integrating them with the competition server.

4.1.1. Data Reading

The first step of the pipeline is to read the data, which is stored in XML files. These files contain textual interactions of users, which can be posts or comments, ordered chronologically. Each file corresponds to a single user and contains a sequence of writings, allowing the analysis of the user's interaction evolution over time. The dataset is labeled so that each user has a binary label: "1" (positive) if the user is identified as at risk of depression, or "0" (negative) if the user belongs to the control group.

The XML files are structured as follows:

```
<INDIVIDUAL>
  <ID> ... </ID>
  <WRITING>
    <TITLE> ... </TITLE>
    <DATE> ... </DATE>
    <INFO> ... </INFO>
    <TEXT> ... </TEXT>
  </WRITING>
  ...
</INDIVIDUAL>
```

Where each XML document includes:

- **ID**: An anonymous user identifier. This ID is unique for each user and ensures the privacy of the data.
- **TITLE**: The title of the post, if available. If the writing is a comment, the TITLE field will be empty.
- **INFO**: Additional information about the origin of the post or comment (e.g., whether it comes from a forum or a blog).
- **TEXT**: The body of the post or comment written by the user.

Regarding the **label** of the users:

- **Depressed Users (labeled as "1")**: These are users identified with signs of depression, either because they explicitly mentioned their diagnosis or due to the analysis of their interactions over time.
- **Control Users (labeled as "0")**: These users do not show signs of depression based on their posts and comments.

The dataset used in eRisk 2025 contains a considerable number of writings, this dataset is divided into several editions, each with a specific number of control and positive users (those at risk of depression). The total dataset size for training is as follows:

- **T1 2017 Dataset**: 752 control users and 135 positive users (with depression).
- **T2 2018 Dataset**: 741 control users and 79 positive users (with depression).
- **T2 2022 Dataset**: 1,302 control users and 98 positive users (with depression).

The users are represented by their writings, which are processed sequentially to identify patterns indicating depression risk. Each dataset contains thousands of writings, enabling the models to train on a large number of examples to detect signs of depression from the user's texts.

4.1.2. Cleaning text

After reading the dataset, the next step is to clean up the text in user posts, an essential process for optimizing the data before it is processed by the model. This process includes:

- Conversion to lowercase: The text is normalized to lowercase to avoid inconsistencies due to capitalization.
- Replacement of URLs and mentions: URLs and mentions of users or subreddits are replaced with generic tokens, eliminating irrelevant information.
- Normalization of entities and special characters: Unicode entities and HTML characters are converted to their readable form, eliminating unnecessary encoding.
- Noise removal: Irrelevant fragments such as URL parameters, external links, and empty words that do not add value are removed based on Exploratory Data Analysis (EDA), where extensive repetitions of words such as "Vik Vik ..." appeared.
- Replacement of numbers and removal of punctuation: Numbers are replaced with a generic token and unnecessary punctuation is removed to avoid interference in the analysis.
- Filtering of short posts: Posts with fewer than five words are discarded, as they do not provide sufficient context.

These steps ensure that the data is consistent and relevant, improving the accuracy of the model in the early detection of depression.

4.1.3. Batch Creation (Batches)

The **batch creation** is a crucial experimental step in our pipeline, aimed at organizing user messages in a way that maximizes computational efficiency and facilitates a more precise analysis of emotional signals. This process involves several key aspects that optimize both the performance and accuracy of the model.

4.1.4. Grouping into Batches

To develop the solution, we decided to group each user's messages into batches of 10. This choice is not arbitrary and is based on two main considerations. First, by dividing the messages into smaller fragments, we optimize computational resources under Colab constraints. Processing large amounts of data caused memory overload and increased processing time. Second, grouping messages allows us to analyze the user's overall emotional context, as a single message alone may not adequately reflect their emotional state. By grouping messages, we can observe patterns of interaction that would be difficult to detect in an isolated message.

4.1.5. Progress Marker j/n

To ensure proper evaluation of each batch, we use a **j/n** marker, where **n** represents the total number of messages in a user's conversation and **j** is the specific message number within the batch. This marker provides a way to track the progress of processing and ensures that the temporal order of messages within each batch is maintained. This approach also facilitates the integration of the full context of each message within the batch, which is essential for precise emotional analysis.

4.1.6. Reclassification within the Batch

One of the most important goals of batch creation is the **reclassification of messages**. Not all messages from a user clearly indicate signs of depression, so classification should be done considering the **general emotional tone** of the batch as a whole. This approach allows the model to classify the messages correctly by taking into account the flow and context of the interaction. Thus, a batch composed of seemingly neutral messages could reveal significant emotional content when analyzed together. This type of contextual analysis has proven essential for accurate user classification.

4.2. Relabeling with VADER

Once the message batches have been created, the VADER (Valence Aware Dictionary and sEntiment Reasoner) sentiment analysis tool is applied to relabel the emotional tone of the messages within each batch. VADER is a particularly effective sentiment analysis model for informal texts, such as social media messages, due to its ability to identify and classify emotions in complex and subjectively charged contexts. It is an approach based on a lexical dictionary of emotions, together with syntactic rules that allow the polarity (positive, negative, or neutral) and intensity of feelings expressed in the texts to be analyzed [19].

The analysis performed by VADER assigns a polarity score to each message, indicating whether the overall tone is positive or negative. In addition, VADER calculates an intensity score for each emotional category, allowing for a more nuanced analysis of the emotionality of messages. This feature is especially useful for detecting subtle feelings or mixed emotions, which are common in user messages and can be difficult to identify using simpler methods or methods based solely on keywords.

In the context of detecting symptoms of depression, relabeling with VADER allows for refining the classification of messages by adjusting labels based on emotional cues found in the texts. This

reassessment of the emotional charge of messages improves the accuracy of the model, especially in situations where messages are not overtly negative, but contain emotional nuances that may indicate depression.

The use of VADER for this relabeling is a strategy proven in previous research, as the tool has been shown to improve the ability of machine learning models to detect complex emotional states, such as depression, by incorporating the analysis of polarity and emotional intensity. According to Hutto and Gilbert [19], VADER is particularly effective at capturing the subtleties of informal texts on social media platforms, making it ideal for tasks such as ours, where users often do not express emotions directly or clearly.

By reevaluating the emotional content of messages using VADER, we ensure that the model's training data reflects the emotional tone that is indicative of depression.

4.3. Embedding

For this stage, Word2Vec was used with specific parameter settings that optimize both the quality of semantic representations and computational efficiency. The model was trained with a `vector_size` of 100, which provides an adequate balance between the ability to represent semantic relationships and computational cost. The `window` size was set to 10, allowing for adequate capture of the context of words within a sufficiently wide range. The parameter `min_count=2` was used to filter out low-frequency words, ensuring that only the most relevant ones influenced the representation. Training was performed using 2 processing cores (`workers=2`), which reduced training time without compromising model quality. In addition, the model was trained for 20 epochs, allowing adequate convergence without the risk of overfitting. Finally, the random seed (`seed=42`) was set to ensure the reproducibility of the results. These parameters were selected after evaluating different configurations and seeking a balance between accuracy and computational efficiency.

4.4. CPI Training with LSTM

The **CPI** model was trained using an **LSTM** (Long Short-Term Memory) to classify message batches. The LSTM processes the vector representations of the messages, which have been obtained using **Word2Vec** and labels adjusted by **VADER**. This type of neural network is particularly well suited for time series tasks, such as ours, as it can capture interaction patterns over time. The ability of LSTMs to remember relevant information throughout sequences allows them to identify emotional risks even when signs of depression manifest gradually. The **CPI** model predicts the risk of depression based on the partial information available in each message in the batch, learning to classify sequentially.

4.5. DMC Modeling with Fixed Policy

The DMC was modeled using a fixed decision policy, designed to evaluate the emotional trend of messages over time. Unlike the CPI model, which classifies individual messages, the DMC does not rely on a trained model but instead applies a set of rules to adjust decisions based on the evolution of messages in the conversation.

In this system, the CPI provides the grouped predictions of the user's messages. These predictions, which reflect the emotional tone of the messages, are passed to the DMC. The DMC then analyzes the emotional trend by observing how the sentiment evolves across the messages. This allows the DMC to make decisions regarding whether the user is at risk of depression (classification as 1) or not at risk (classification as 0).

The DMC measures the emotional trend by evaluating several key parameters:

- **Median**: The median of the predictions from previous batches is calculated to identify the general trend.
- **Trend**: The difference between the first and last values in the sequence of predictions, which indicates whether the user's emotional state is improving or worsening.
- **Standard deviation**: Measures the dispersion of predictions, ensuring that decisions are not made at times of high emotional variability.
- **Maximum score**: Verifies that the prediction score is above a minimum threshold to ensure that emotional changes are significant.

The $adaptive\_rule$ function adjusts the intervention decision based on these parameters, using a trend threshold and a maximum standard deviation to ensure that decisions are robust and that no misclassifications are made due to minor emotional fluctuations. The fixed policy ensures that the system acts consistently and in line with the user's overall behavior, without the need for additional training.

The parameters selected for the DMC were determined after parameter search, evaluating different combinations of threshold, slope weight, standard deviation limit, and minimum delay. The following parameters were found to be optimal for this task: threshold = 0.6, slope weight = 0.4, std max = 0.2, and min delay = 5. These values were selected based on the results of multiple tests, based on metrics such as ERDE, precision, recall, and F1-score to evaluate model performance. The optimal parameter combination was the one that yielded the best performance in terms of both detection accuracy and decision latency.

The DMC's goal is to determine when a user shows clear signs of emotional deterioration or improvement. If the model detects that the emotional trend indicates deterioration and that the dispersion is low (i.e., that the emotional behavior is consistent), the system can make an intervention decision.

This fixed policy-based approach is especially useful in scenarios such as ours, where detecting emotional changes over time is the key to accurately classifying the risk of depression, without the need for deep model training. The use of these adaptive rules improves the efficiency of the system, while maintaining consistent and reliable classification based on the user's emotional behavior.

4.6. Results

The results obtained during the evaluation phase of Task 2 are presented in Table 3. Key performance metrics such as Precision (P@10), Recall (R), F_1-score (F1), ERDE@5, ERDE@50, Latency, Speed, and $F_{latency}$ are shown. Below, each of these metrics is briefly explained:

- Precision (P@10): Measures the proportion of relevant messages among the first 10 messages recommended or classified by the model.
- Early Risk Detection Error: Measures the cost associated with early risk detection, evaluating both the precision and the latency of the intervention depending on the number of messages indicating by the @.
- Latency: Measures the number of writings processed before the true positive detection is made. It indicates how quickly the system can detect a risk case.
- Speed: Evaluates how fast the system processes messages and makes intervention decisions. It is calculated based on the median penalty of detected true positives, with a higher penalty indicating slower detection.
- $F_{latency}$: A combined metric that takes into account both precision and latency, providing an overall performance evaluation of the system in terms of its ability to act both precisely and promptly.

Table 3
Decision-based evaluation for Task 2

Team	Run	P	R	F_1	$ERDE_5$	$ERDE_{50}$	$Latency_{TP}$	$Speed$	$F_{Latency}$
HIT–SCIR	0	0.72	0.96	0.82	0.06	0.03	4.00	0.99	0.81
	1	0.72	0.95	0.82	0.06	0.03	4.00	0.99	0.81
	2	0.74	0.94	0.83	0.06	0.03	4.00	0.99	0.82
	3	0..73	0.94	0.82	0.08	0.03	7.00	0.98	0.80
	4	0.77	0.94	0.85	0.09	0.03	8.00	0.97	0.82
ELiRF-UPV	0	0.78	0.81	0.79	0.08	0.04	7.00	0.98	0.78
	1	0.37	0.62	0.46	0.07	0.06	1.00	1.00	0.46
	2	0.83	0.47	0.60	0.10	0.07	8.00	0.97	0.58
	3	0.68	0.47	0.67	0.09	0.05	7.00	0.98	0.66
	4	0.68	0.47	0.67	0.09	0.05	7.00	0.98	0.66
HU	0	0.61	0.77	0.68	0.09	0.05	10.00	0.96	0.66
	1	0.72	0.77	0.75	0.10	0.05	11.00	0.96	0.72
	2	0.14	0.94	0.25	0.15	0.09	6.0	0.98	0.24
	3	0.11	1.00	0.20	0.11	0.10	1.00	1.00	0.20
	4	0.27	0.88	0.41	0.10	0.07	11.00	0.96	0.40
COTECMAR-UTB	0	0.29	0.65	0.40	0.12	69.00	0.74	0.74	0.29
	1	0.25	0.01	0.02	0.11	1.00	1.00	1.00	0.02

Table 4
Ranking-based evaluation for Task 2

Team	Run	1 Writing			100 Writing			500 Writing			1000 Writing		
		P@10	NDCG@10	NDCG@100	P@10	NDCG@10	NDCG@100	P@10	NDCG@10	NDCG@100	P@10	NDCG@10	NDCG@100
HIT–SCIR	0	1.00	1.00	0.58	1.00	1.00	0.84	1.00	1.00	0.89	1.00	1.00	0.90
	1	1.00	1.00	0.58	1.00	1.00	0.84	1.00	1.00	0.89	1.00	1.00	0.90
	2	1.00	1.00	0.58	1.00	1.00	0.84	1.00	1.00	0.89	1.00	1.00	0.90
	3	1.00	1.00	0.58	1.00	1.00	0.84	1.00	1.00	0.89	1.00	1.00	0.90
	4	1.00	1.00	0.58	1.00	1.00	0.84	1.00	1.00	0.89	1.00	1.00	0.90
ELiRF-UPV	0	0.90	0.88	0.36	1.00	1.00	0.69	0.90	0.94	0.74	0.90	0.81	0.74
	1	0.30	0.25	0.32	1.00	1.00	0.45	1.00	1.00	0.44	1.00	1.00	0.46
	2	0.20	0.31	0.14	1.00	1.00	0.45	1.00	1.00	0.44	1.00	1.00	0.46
	3	0.90	0.94	0.35	1.00	1.00	0.68	0.60	0.46	0.60	0.70	0.63	0.63
	4	0.60	0.75	0.27	1.00	1.00	0.68	0.60	0.46	0.60	0.70	0.63	0.63
HU	0	0.90	0.81	0.53	0.80	0.87	0.49	0.70	0.68	0.48	0.70	0.66	0.49
	1	1.00	1.00	0.62	0.90	0.88	0.57	0.60	0.71	0.35	0.40	0.60	0.26
	2	0.30	0.21	0.11	0.20	0.16	0.12	0.00	0.00	0.11	0.40	0.60	0.26
	3	0.30	0.21	0.11	0.20	0.16	0.12	0.00	0.00	0.11	0.40	0.60	0.26
	4	0.60	0.53	0.33	0.4	0.58	0.36	0.30	0.37	0.24	0.40	0.60	0.26
COTECMAR-UTB	0	0.30	0.23	0.23	0.00	0.00	0.22	0.20	0.15	0.18	0.20	0.13	0.17
	1	0.00	0.00	0.12	0.00	0.00	0.12	0.00	0.00	0.12	0.00	0.00	0.12

4.7. Conclusions on the Results

The results obtained by our team in run 0 reflect a moderate level of performance. First, the model achieved an F1 score of 0.40, which points to a reasonable balance between precision and recall; this suggests that, although the system is capable of retrieving a reasonable number of relevant cases, its overall effectiveness is limited by imbalances in these two key metrics. In particular, the recall value

of 0.65 indicates that the model managed to capture a significant proportion of the actual positive cases, which is an encouraging sign for early detection. However, the low accuracy of 0.29 reveals that the system stumbled upon a significant challenge in distinguishing the most relevant or informative messages, resulting in a higher false positive rate. This imbalance reveals the need to further refine the model's decision-making processes, especially with regard to improving its capability to better accurately filter and focus on high-risk content.

In terms of latency, the value of 69.00 suggests that the system took a reasonable amount of time to make decisions, but there is still room to optimize this metric to achieve faster interventions. ERDE@5 of 0.12 shows a relatively low cost for early risk detection, which means that the model was able to make decisions without significant delays, although further optimization could reduce this cost even more.

In the ranking-based evaluation as shown in Table 4, the metrics P@10 and NDCG@10 were relatively low, suggesting that the model had difficulty prioritizing the most relevant messages for the detection of depression. This indicates that the system needs adjustments to improve the ranking of critical messages.

4.8. Future Work

To improve the performance of the model in future editions of the challenge, the following strategies are proposed:

- Optimization of the CPI model: Use more advanced architectures, such as **BERT** or **DistilBERT**, which can better handle language complexities and provide better semantic representations of messages.
- Adjusting the Decision Policy in the DMC: Explore approaches based on **adaptive learning** for the fixed decision policy, so that the model can be dynamically adjusted according to the signals in the messages.
- Improve relabeling with VADER: Evaluate additional sentiment analysis techniques or even combine analysis tools to increase the accuracy of message relabeling.

These strategies aim to improve system efficiency, increase prediction accuracy, and ensure that intervention decisions are made at the right time.

Acknowledgments

We would like to express our sincere gratitude to COTECMAR for providing the necessary space and resources to carry out this research. We also extend our thanks to the Universidad Tecnológica de Bolívar (UTB) for offering the facilities and processing services for running the models, which significantly contributed to the success of this work. Additionally, we acknowledge the support received through the Convocatoria 950 of 2024 sponsored by Minciencias, which provided the resources for the scholarship that made this research possible.

Declaration on Generative AI

Throughout the process of crafting this work, the authors used ChatGPT-4, ChatGPT-4o-mini, and o3-mini AI models to assist with the enhancement and reorganisation of the text. These tools were employed to improve pragmatic clarity, optimise coherence, and ensure consistency in scientific language. Additionally, DeepL was utilised on certain occasions for translation purposes. Following the use of

these tools, the authors reviewed and edited the content as necessary and take full responsibility for the publication's content.

References

[1] A. Mansoor, F. Ansari, Early detection of mental health crises through ai-powered social media analysis: A prospective observational study, Journal of Personalized Medicine 14 (2024) 153.

[2] X. Liu, Y. Zhang, H. Wang, Detecting and measuring depression on social media using a machine learning approach: Systematic review, JMIR Mental Health 9 (2022) e32786. doi:10.2196/27244.

[3] 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. doi:10.1109/ACCESS.2019.2909180.

[4] Z. Khan, M. Ali, Unravelling minds in the digital era: Mapping mental health disorders through machine learning using online social media, Social Network Analysis and Mining (2024).

[5] J. Parapar, P. Martín-Rodilla, D. E. Losada, F. Crestani, Overview of erisk at clef 2024: Early risk prediction on the internet (extended overview), CEUR Workshop Proceedings (2024). URL: https://ceur-ws.org/Vol-3740/paper-72.pdf, presented at CLEF 2024, September 09–12, 2024, Grenoble, France.

[6] B. H. Ang, S. D. Gollapalli, S.-K. Ng, Nus-ids@erisk2024: Ranking sentences for depression symptoms using early maladaptive schemas and ensembles, in: G. Faggioli, N. F. 0001, P. Galuscáková, A. G. S. de Herrera (Eds.), Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024), Grenoble, France, 9-12 September, 2024, volume 3740 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 782–793. URL: https://ceur-ws.org/Vol-3740/paper-73.pdf.

[7] A. Beck, R. Steer, G. Brown, Beck Depression Inventory-II (BDI-II): Manual, San Antonio, TX, 1996.

[8] D. Guecha, A. Potdar, A. Miyaguchi, Ds@gt erisk 2024: Sentence transformers for social media risk assessment, 2024. URL: https://arxiv.org/abs/2407.08008. arXiv:2407.08008.

[9] H. Thompson, L. Cagnina, M. Errecalde, Strategies to harness the transformers' potential: Unsl at erisk 2023, 2023. URL: https://arxiv.org/abs/2310.19970. arXiv:2310.19970.

[10] A.-M. Bucur, Utilizing chatgpt generated data to retrieve depression symptoms from social media, 2023. URL: https://arxiv.org/abs/2307.02313. arXiv:2307.02313.

[11] J. M. Loyola, S. G. Burdisso, H. Thompson, L. C. Cagnina, M. L. Errecalde, Unsl at erisk 2021: A comparison of three early alert policies for early risk detection, in: Conference and Labs of the Evaluation Forum, 2021. URL: https://api.semanticscholar.org/CorpusID:237298519.

[12] J. M. Loyola, M. L. Errecalde, H. J. Escalante, M. Montes y Gomez, Learning when to classify for early text classification, in: Proceedings of the Argentine Congress of Computer Science, Springer, 2018, pp. 24–34. URL: https://doi.org/10.1007/978-3-319-75214-3_3. doi:10.1007/978-3-319-75214-3_3.

[13] J. Parapar, A. Perez, X. Wang, F. Crestani, Overview of erisk 2025: Early risk prediction on the internet, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction - 16th International Conference of the CLEF Association, CLEF 2025, Madrid, Spain, September 9-12, 2025, Proceedings, Part II, volume To be published of *Lecture Notes in Computer Science*, Springer, 2025.

[14] J. Parapar, A. Perez, X. Wang, F. Crestani, Overview of erisk 2025: Early risk prediction on the internet (extended overview), in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), Madrid, Spain, 9-12 September, 2025, volume To be published of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025.

[15] J. Wei, K. Zou, Eda: Easy data augmentation techniques for boosting performance on text classification tasks, 2019. URL: https://arxiv.org/abs/1901.11196. arXiv:1901.11196.

[16] S. Raschka, Y. H. Liu, V. Mirjalili, Machine learning with pytorch and scikit-learn: Develop machine learning and deep learning models with python, 2022. URL: https://github.com/rasbt/machine-learning-book, iSBN: 9781801819312.

[17] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2019. URL: https://arxiv.org/abs/1910.01108. arXiv:1910.01108.

[18] J. Liu, A. Liu, X. Lu, S. Welleck, P. West, R. L. Bras, Y. Choi, H. Hajishirzi, Generated knowledge prompting for commonsense reasoning, 2022. URL: https://arxiv.org/abs/2110.08387. arXiv:2110.08387.

[19] C. Hutto, E. Gilbert, Vader: A parsimonious rule-based model for sentiment analysis of social media text, Proceedings of the International AAAI Conference on Web and Social Media 8 (2014) 216–225.

[20] H. Fabregat, A. Duque, L. Araujo, J. Martinez-Romo, Nlp-uned-2 at erisk 2023: Detecting pathological gambling in social media through dataset relabeling and neural networks., in: CLEF (Working Notes), 2023, pp. 672–683. URL: https://ceur-ws.org/Vol-3497/paper-056.pdf.