Awakened at eRisk 2026: Contextualized Early Detection of Depression using DeBERTa Embeddings and Temporal MLP Classification

Notebook for the eRisk Lab at CLEF 2026

Cosmin-Gabriel Becheru[1], Ciprian-Octavian Truică[1,2,*] and Elena-Simona Apostol[1]

[1]National University of Science and Technology POLITEHNICA Bucharest, Splaiul Independenţei 313, Bucureşti 060042, Romania
[2]Academy of Romanian Scientists, Ilfov 3, Bucharest, 050044, Romania

Abstract

This paper describes how our team, Awakened, took part in Task 2 of the eRisk 2026 Lab at CLEF 2026. The task is about detecting depression early from Reddit conversation threads, taking context into account. What we built is a pipeline where transformer-based sentence embeddings feed into a temporal MLP classifier that works on features constructed incrementally as the conversation unfolds. Each event, whether a submission or a comment, gets encoded with DeBERTa-v3-base. From there, we build feature vectors at every step that reflect how the cumulative, target-user, and contextual embedding statistics change over the course of the conversation. These prefix features then go into a multi-layer perceptron with GELU activations, which outputs a depression risk probability at each step. We tested two variants internally with different MLP sizes and submitted the one that worked better as our single official run. Based on the preliminary evaluation, our submission processed 118 user threads and reached an F1 of 0.45. It also showed perfect precision and latency for immediate early detection, with a P@10 of 1.00 at 1 writing. The remainder of the paper describes the full pipeline, examines how the feature-engineering choices affected performance, and discusses possible improvements for future work.

Keywords

early detection of depression, social media, transformer embeddings, temporal classification, DeBERTa, eRisk, CLEF 2026

1. Introduction

People have used online communities to talk about personal struggles for years, and mental health is no exception. Reddit stands out here because of its pseudonymous setup and how it is organized around communities, which means users end up producing a lot of self-disclosed writing that can hint at their psychological state [1]. Since none of this is prompted by a clinician or collected under controlled conditions, it gives a view of mental health that you simply do not get from structured questionnaires or interviews.

Computational approaches to detecting depression from this kind of data have gone through several stages. Earlier systems used hand-crafted lexical features paired with shallow classifiers [2], while more recent work leans on transformer-based text representations that pick up on richer semantic and contextual information [3, 4]. Still, there is a gap between static text classification, where a model gets a full corpus of user writings and outputs a single label, and the harder problem of *early detection*, where the system has to make decisions from incomplete data that arrives incrementally over time.

The eRisk Lab at CLEF has served as the main benchmark for this early detection setup since 2017 [5, 6, 7]. The 2026 edition of Task 2 [8, 9] builds on what was introduced in 2025 [10]: instead of only looking at the target user's posts, systems now receive full conversational threads, including messages

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany
*Corresponding author.
cosmin.becheru@upb.ro (C. Becheru); ciprian.truica@upb.ro (C. Truică); elena.apostol@upb.ro (E. Apostol)
https://sites.google.com/view/ciprian-octavian-truica (C. Truică); https://sites.google.com/view/elena-simona-apostol (E. Apostol)
0009-0002-7720-8048 (C. Becheru); 0000-0001-7292-4462 (C. Truică); 0000-0001-6397-4951 (E. Apostol)

from other participants. This contextualized setting brings up a question that earlier editions did not really touch on: *can the language and behavior of people around a user, how they respond, what they talk about, the tone they use, help in identifying that user's depression risk?*

Our system, submitted under the team name Awakened, puts this idea into practice through a four-stage pipeline. Each event in a user's timeline, whether a submission or a comment, is encoded into a 768-dimensional vector using DeBERTa-v3-base [11], with role-aware text formatting that separates target-user events from contextual ones. From there, we build temporal prefix features that keep running averages of the target user, context, and combined embedding spaces. These features are then passed to a multi-layer perceptron (MLP) that outputs per-step depression risk probabilities. At the end, a consecutive-threshold decision policy, optimized through grid search, converts the continuous probabilities into binary early-detection decisions.

2. Related Work

2.1. Early Depression Detection from Social Media

Using social media data to assess mental health goes back to the work of De Choudhury et al. [2], who showed that linguistic markers and behavioral patterns on Twitter could predict depressive episodes at a population scale. The field has grown substantially since then. Garg [1] surveys the computational methods used for mental health analysis on social media and traces how the models evolved from bag-of-words to transformer-based architectures. Zhang et al. [3] cover a similar arc in a narrative review centered on NLP approaches to mental illness detection, going from rule-based systems to deep learning. Within the eRisk Lab, the early detection setup has been adjusted over several editions. The 2022 to 2024 rounds [5, 12, 6] dealt with single-user writing histories, and the 2025 edition [10] brought in the contextualized variant that Task 2 of eRisk 2026 carries forward. The foundational test collection behind the eRisk depression detection tasks was put together by Losada and Crestani [7].

2.2. Transformer Embeddings for Mental Health NLP

Most teams in recent eRisk editions have moved to transformer-based models for text representation. One example is MentalBERT [4], which Ji et al. pre-trained specifically on mental health content from Reddit. It outperformed general-purpose models on benchmarks for depression, anxiety, and stress classification. We chose DeBERTa-v3 [11] for our pipeline because of its disentangled attention mechanism and ELECTRA-style pre-training, both of which gave it an edge over older BERT variants on standard NLU tasks. In the eRisk 2024 edition, Fabregat et al. [13] proposed a pipeline combining sentence encoders with approximate nearest neighbors for the early detection of anorexia, achieving competitive results through efficient retrieval-based methods. Riewe-Perła and Filipowska [14] explored hybrid architectures combining recommender systems with language models for the same task. Beyond the eRisk context, Truică et al. [15] demonstrated that combining textual content representations with social context features through a deep neural network ensemble can substantially improve classification performance, even under limited training data — a finding that motivates our own separation of target-user and contextual embeddings.

2.3. Temporal and Sequential Classification

Our work relates to the broader literature on temporal text classification, where the goal is to make predictions from incrementally observed sequences of documents. Rather than fine-tuning a transformer end-to-end for classification, we use the transformer as a fixed feature extractor and build temporal prefix features that capture evolving conversation dynamics. This design allows efficient GPU-accelerated training of a lightweight classifier while preserving the rich semantic representations from the pre-trained model.

3. Task Definition and Data

Task 2 of eRisk 2026 focuses on the contextualized early detection of depression [8, 9]. As specified by the organizers, the training phase reuses the contextualized 2025 test collection (909 Reddit users) [10], divided into two categories: individuals suffering from depression (positive) and control users (negative). The 2026 test phase is conducted interactively over a separate set of 500 user threads released by the eRisk server. For each user, the collection contains a chronological sequence of writings, including both the target user's posts and the surrounding conversational context.

There are two kinds of events in the data. A submission is a post that a user creates, and it includes a title, body text, who wrote it, and when. A comment is a reply, either to a submission or to another comment, and the parent references between them create a tree-like conversational structure. Labels are stored separately. Each user gets a binary label, 0 for control and 1 for positive. Out of the 909 users in the training set we worked with, 807 are controls and 102 are positive, so the class imbalance sits around 8 to 1. We had to handle this in the loss function. The test phase runs through the eRisk server in an interactive fashion. Threads for each target user come in one after another, and after each one the system has to output a binary call (positive or negative) along with a confidence score. The idea is to mimic what continuous monitoring would look like in practice. Two evaluation paradigms are used. One looks at the decisions directly through Precision, Recall, F1, ERDE [7], latency, and latency-weighted F1. The other is ranking-based and reports Precision@K and NDCG@K across several writing thresholds.

4. Methodology

Our pipeline consists of four sequential stages: (1) data parsing and event extraction, (2) transformer-based event embedding, (3) temporal prefix feature construction, and (4) MLP classification with decision policy optimization.

4.1. Data Parsing

Each user's JSON file is parsed to extract a flat chronological sequence of events. For each event, we record: user ID, target subject, timestamp, round number, event type (submission/comment), author, parent reference, and whether the author is the target subject (`is_target_author`). The event text is constructed by concatenating the title and body for submissions, or using the body alone for comments. Events are sorted chronologically and assigned sequential step indices.

4.2. Event Embedding with DeBERTa-v3-base

Each event is encoded using `microsoft/deberta-v3-base` [11], a transformer model with a 768-dimensional hidden state. We construct a structured text representation for each event that incorporates role and type information:

```
[ROLE] TARGET/OTHER [TYPE] submission/comment
[TITLE] ... [TEXT] ... [PARENT] ...
```

The [ROLE] prefix distinguishes whether the event was authored by the target user (TARGET) or another participant (OTHER), allowing the model to produce role-aware representations without fine-tuning. Events are tokenized with a maximum length of 256 tokens and encoded using mean pooling over the last hidden state. Embeddings are stored as float16 in a memory-mapped array for efficient disk-backed access.

To handle the large dataset (909 users and potentially millions of events), we introduced several optimizations. Batch sizes are set dynamically based on estimated token counts so the GPU stays busy without running into out-of-memory errors. Within each chunk, events are sorted by estimated token length to cut down on padding. If a CUDA or MPS out-of-memory error still happens, the system

falls back automatically by shrinking the batch size with a configurable backoff factor and, if needed, encoding events one at a time.

4.3. Temporal Prefix Feature Construction

At every time step in a user's chronological event sequence, we build a feature vector that summarizes everything in the conversation up to that point. This prefix-based design means the classifier can produce a prediction at any step, which is what the early detection setup requires. Each prefix feature vector has 2,309 dimensions (5 meta features plus 3×768 embedding dimensions). The meta features (5 dimensions) capture basic counts and flags: how many events have been processed so far (`count_all`), how many of those belong to the target user (`count_target`), how many come from other participants (`count_context`), whether the current event was written by the target user (`is_target_author`), and whether it is a comment rather than a submission (`is_comment`). The embedding aggregates ($3 \times 768 = 2{,}304$ dimensions) are running averages computed over the event embeddings. `mean_all` is the cumulative mean across all events up to the current step, `mean_target` averages only the target user's events, and `mean_context` averages the events from everyone else. We keep the target and context embeddings separate because we wanted to test whether the conversational context, things like how others respond to the user or what topics come up between them, carries signals that the target user's own language alone would not show.

4.4. MLP Classifier

The classifier is a feed-forward neural network with three layers, GELU activations [16], and dropout regularization:

```
Input (D) -> Linear(D, H) -> GELU -> Dropout(p) ->
Linear(H, H/2) -> GELU -> Dropout(p) ->
Linear(H/2, 1) -> Sigmoid
```

For the loss function, we used binary cross-entropy with logits (`BCEWithLogitsLoss`), with a `pos_weight` derived from the training label distribution to deal with the class imbalance [17]. The optimizer is AdamW with gradient clipping set to `max_norm=1.0`. We also enabled mixed precision through `torch.amp.autocast` to speed up GPU computation. All input features are z-score normalized based on statistics from the training set. Because the feature matrices can reach millions of rows, we could not just load everything into memory at once. Instead, both training and validation rely on streaming Parquet readers that go through the data in configurable batch sizes. Training stops early if the validation loss does not improve for 3 consecutive epochs. The train/validation split is done at the user level with an 80/20 ratio using `GroupShuffleSplit`, so that all events belonging to the same user always end up in the same partition.

4.5. Decision Policy

Each step in a user's timeline gets a probability from the MLP. The way we decide whether someone is positive or not is based on a simple rule: the probability has to cross a threshold τ and remain above it for c consecutive steps, but the system waits until at least m steps have passed before it even starts checking. If none of that ever happens for a given user, we default to comparing their last predicted probability against 0.5.

The parameters (τ, m, c) are optimized via exhaustive grid search on the validation set:

- $\tau \in [0.30, 0.94]$ with step 0.01
- $m \in \{1, 2, 3\}$
- $c \in \{1, 2, 3\}$

The ranking criterion balances F1 (primary), average decision fraction (earliness), and accuracy.

4.6. Model Configurations

We tried two MLP configurations, which are listed in Table 1. The one that did better on our internal evaluation was the one we went with for the official submission.

Table 1
Model configurations. Both variants use the same DeBERTa-v3-base embeddings and prefix feature set; they differ in MLP capacity and regularization strength.

Parameter	Variant 1	Variant 2
Hidden dimension	512	1024
Dropout	0.25	0.20
Weight decay	1e-4	5e-4
Input dimension	2,309	2,309
Learning rate	5e-5	5e-5
Epochs	16	16
Train batch size	1,024	1,024
Decision threshold (τ)	0.95	0.95

5. Results and Discussion

5.1. Internal Validation Results

Table 2 presents the validation results for both variants at the best threshold (τ=0.95), using a consecutive policy with `min_steps=2` and `consecutive=2`.

Table 2
Validation metrics for both variants. Variant 2 achieves higher F1 and precision, while Variant 1 shows a slightly higher AUC.

Metric	Variant 1	Variant 2
Accuracy	0.824	0.846
Precision	0.425	0.474
Recall	0.654	0.692
F1	0.515	0.563
AUC	0.775	0.758
Avg. Decision Fraction	0.796	0.818

5.2. Decision Policy Optimization

We ran a grid search over the decision policy parameters using the validation probabilities from Variant 2. The search went through 585 combinations (65 thresholds $\times$ 3 `min_steps` $\times$ 3 `consecutive` values). The configuration that worked best reached an F1 of 0.600 with τ = 0.49, m = 3, and c = 2 (Precision = 0.477, Recall = 0.808, AUC = 0.842, Avg. Decision Fraction = 0.771). Going from τ=0.95 (F1=0.563) down to τ=0.49 (F1=0.600) shows that the original threshold was too conservative and was causing the system to miss true positives. Table 3 shows how Variant 2 performs through different threshold values, along with the trade-off between precision and recall.

5.3. Official Evaluation Results

According to the preliminary evaluation released by the eRisk 2026 organizers [8, 9], our official submission processed 118 of the 500 user threads released during the test phase. On the decision-based evaluation, the system obtained Precision = 0.33, Recall = 0.68, F1 = 0.45, ERDE_5 = 0.11, ERDE_{50} =

Table 3
Variant 2 metrics across thresholds (`consecutive=2, min_steps=2`). Higher thresholds improve precision at the cost of recall.

Threshold	Precision	Recall	F1	Avg. Dec. Frac.
0.50	0.396	0.731	0.514	0.759
0.65	0.422	0.731	0.535	0.777
0.75	0.422	0.731	0.535	0.778
0.85	0.439	0.692	0.537	0.803
0.95	0.474	0.692	0.563	0.818

0.10, median latency on true positives = 1.0, speed = 1.0, and latency-weighted F1 = 0.45 [8, 9]. On the ranking-based evaluation, the system achieved P@10 = 1.00 and NDCG@10 = 1.00 at every reported cut-off (1, 100, 250, and 500 writings), with NDCG@100 of 0.54, 0.63, 0.63, and 0.51 respectively [8, 9]. The combination of perfect P@10 at 1 writing with a minimum latency of 1 confirms that the earliest positive decisions issued by our system were highly precise.

5.4. Discussion

Looking at the internal validation results, the variant with `hidden_dim=1024` gets a higher F1 than the one with `hidden_dim=512`, 0.563 compared to 0.515. Precision follows the same pattern, 0.474 versus 0.425. The bigger hidden layer likely helps the network get more out of the prefix features. Variant 1 with `hidden_dim=512` does edge it out slightly on AUC (0.775 vs 0.758), suggesting that it ranks positive examples above negative ones slightly better, despite having lower F1 and precision.

Each prefix feature captures a different slice of the discussion history. `mean_all` averages all messages up to a given point, `mean_target` averages only what the target user has written, and `mean_context` averages contributions from everyone else. Keeping target and context separate is motivated by research showing such conversational dynamics around a user, for example how others respond or what subjects come up, provide relevant cues for depression detection [10].

The grid search over the decision policy brought F1 from 0.563 up to 0.600, which amounts to a 6.5% gain. What this shows is that how you go from raw probabilities to binary labels actually matters and should not be treated as a trivial post-processing choice. Setting the consecutive-step threshold at c=2 worked best in practice because it discards isolated probability spikes that would otherwise count as false positives.

A few limitations should be noted. The training set only contains messages from the target user, so conversational context is absent during training but present at test time, creating a mismatch. We did not fine-tune DeBERTa on any mental health corpus, and a model like MentalBERT [4] might be more sensitive toward domain-specific language. The MLP treats every time step independently without modeling their order, so something with recurrence or attention across time steps could do better. The class imbalance (`pos_weight` ≈ 4.56) is another concern. We use a weighted loss to counteract it, but precision stays low across all configurations, which suggests that the weighting alone does not solve the problem.

6. Conclusions and Future Work

We presented a pipeline for contextualized early detection of depression that combines DeBERTa-v3-base embeddings with temporal prefix features and an MLP classifier. The system processes each users conversation timeline incrementally and builds feature vectors that represent cumulative, target-user, and contextual embedding statistics as they develop over time. The final binary classification is obtained through a consecutive-threshold decision policy.

The results show that temporal prefix features can summarize evolving conversations for early detection without relying on sequential models. Increasing the MLP capacity from 512 to 1024 hidden

units led to better classification performance in this setting. The decision policy also had a clear effect on the results, with the best F1 score increasing from 0.563 to 0.600 (a relative gain of $\approx 6.5\%$), after tuning the threshold and the number of consecutive steps. The official P@10 score of 1.00 at 1 writing indicates that the earliest positive decisions made by the system were reliable.

Future work will consider domain-adapted encoders, such as fine-tuning DeBERTa or using MentalBERT [4] on the training data before feature extraction. Another direction is to replace the per-step MLP with sequential classifiers, including LSTM, GRU, or Transformer-based models, in order to model temporal dependencies more directly. Additional features could also be added, such as first-person pronoun frequency [18] and sentiment scores, alongside embedding-based representations. The decision policy could be learned jointly with the classifier instead of being optimized after training. Ensemble methods that combine different embedding models and classifier architectures may also be examined to improve prediction stability.

Acknowledgments

The research presented in this paper is supported in part by The Academy of Romanian Scientists, through the funding of the project "NetGuardAI: Intelligent system for harmful content detection and immunization on social networks" (AOȘR-TEAMS-IV).

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly for grammar and spelling checks, paraphrasing, and writing-style improvements, and ChatGPT for light language-editing support and formatting assistance. These tools were used solely to improve clarity, readability, and presentation. After using these tools, the authors carefully reviewed, edited, and verified the content as needed and take full responsibility for the publication's content.

References

[1] M. Garg, Mental health analysis in social media posts: A survey, Archives of Computational Methods in Engineering 30 (2023) 1819–1842. doi:10.1007/s11831-022-09863-z.

[2] M. De Choudhury, S. Counts, E. Horvitz, Social media as a measurement tool of depression in populations, in: Proceedings of the 5th Annual ACM Web Science Conference, WebSci '13, ACM, 2013, pp. 47–56. doi:10.1145/2464464.2464480.

[3] T. Zhang, A. M. Schoene, S. Ji, S. Ananiadou, Natural language processing applied to mental illness detection: A narrative review, npj Digital Medicine 5 (2022) 46. doi:10.1038/s41746-022-00589-7.

[4] S. Ji, T. Zhang, L. Ansari, J. Fu, P. Tiwari, E. Cambria, MentalBERT: Publicly available pretrained language models for mental healthcare, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, LREC 2022, European Language Resources Association, 2022, pp. 7184–7190. URL: https://aclanthology.org/2022.lrec-1.778.

[5] J. Parapar, P. Martín-Rodilla, D. E. Losada, F. Crestani, Overview of eRisk 2022: Early risk prediction on the internet, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction - 13th International Conference of the CLEF Association, CLEF 2022, Bologna, Italy, September 5–8, 2022, Proceedings, Lecture Notes in Computer Science, Springer, 2022, pp. 233–256. doi:10.1007/978-3-031-13643-6_18.

[6] J. Parapar, P. Martín-Rodilla, D. E. Losada, F. Crestani, Overview of eRisk 2024: Early risk prediction on the internet, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction - 15th International Conference of the CLEF Association, CLEF 2024, Grenoble, France, September 9–12, 2024, Proceedings, Lecture Notes in Computer Science, Springer, 2024, pp. 73–92. doi:10.1007/978-3-031-71908-0_4.

[7] D. E. Losada, F. Crestani, A test collection for research on depression and language use, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction - 7th International Conference of the CLEF Association, CLEF 2016, Évora, Portugal, September 5–8, 2016, Proceedings, Lecture Notes in Computer Science, Springer, 2016, pp. 28–39. doi:10.1007/978-3-319-44564-9_3.

[8] A. Perez, J. Parapar, X. Wang, F. Crestani, Overview of eRisk 2026 Early Risk Prediction on the Internet: Symptom Ranking and Conversational Approaches for Depression and ADHD (Extended Overview), in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2026), Jena, Germany, 21–24 September, 2026, volume To be published of *CEUR Workshop Proceedings*, CEUR-WS.org, 2026.

[9] A. Perez, J. Parapar, X. Wang, F. Crestani, Overview of eRisk 2026 Early Risk Prediction on the Internet: Symptom Ranking and Conversational Approaches for Depression and ADHD, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction - 17th International Conference of the CLEF Association, CLEF 2026, Jena, Germany, September 21–24, 2026, Proceedings, volume To be published of *Lecture Notes in Computer Science*, Springer, 2026.

[10] 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, Lecture Notes in Computer Science, Springer, 2025, pp. 242–265. doi:10.1007/978-3-032-04354-2_15.

[11] P. He, J. Gao, W. Chen, DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing, in: International Conference on Learning Representations, 2023. URL: https://openreview.net/forum?id=sE7-XhLxHA.

[12] J. Parapar, P. Martín-Rodilla, D. E. Losada, F. Crestani, Overview of eRisk 2023: Early risk prediction on the internet, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction - 14th International Conference of the CLEF Association, CLEF 2023, Thessaloniki, Greece, September 18–21, 2023, Proceedings, Lecture Notes in Computer Science, Springer, 2023, pp. 294–315. doi:10.1007/978-3-031-42448-9_22.

[13] H. Fabregat, D. Deniz, A. Duque, L. Araujo, J. Martínez-Romo, NLP-UNED at eRisk 2024: Approximate nearest neighbors with encoding refinement for early detecting signs of anorexia, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024), CEUR Workshop Proceedings, CEUR-WS.org, 2024, pp. 813–824.

[14] O. Riewe-Perła, A. Filipowska, Combining recommender systems and language models in early detection of signs of anorexia, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024), CEUR Workshop Proceedings, CEUR-WS.org, 2024.

[15] C.-O. Truică, E.-S. Apostol, P. Karras, DANES: Deep neural network ensemble architecture for social and textual context-aware fake news detection, Knowledge-Based Systems 294 (2024) 111715. doi:10.1016/j.knosys.2024.111715.

[16] D. Hendrycks, K. Gimpel, Gaussian error linear units (GELUs), arXiv preprint arXiv:1606.08415 (2016).

[17] C.-O. Truică, C. A. Leordeanu, Classification of an imbalanced data set using decision tree algorithms, University Politehnica of Bucharest Scientific Bulletin - Series C Electrical Engineering and Computer Science 79 (2017) 69–84.

[18] T. Edwards, N. S. Holtzman, A meta-analysis of correlations between depression and first person singular pronoun use, Journal of Research in Personality 68 (2017) 63–68. doi:10.1016/j.jrp.2017.02.005.