DeepCare at eRisk 2026: Decoupling Risk Ranking from Latency-Aware Decisions for Contextualized Early Detection of Depression

Notebook for the eRisk Lab at CLEF 2026

Gheorghe Nistor[1,*], Ana-Sabina Uban[1]

[1]Faculty of Mathematics and Computer Science, University of Bucharest, 14 Academiei Street, 010014 Bucharest, Romania

Abstract

This paper describes the participation of team DeepCare in Task 2 of eRisk 2026, Contextualized Early Detection of Depression. Unlike earlier editions that streamed a user's posts one at a time, this edition releases entire Reddit discussion threads round by round, so a system sees the target user alongside the people they reply to. At each round the task asks for two outputs per user, judged by different metrics: a binary alert and a continuous risk score. We submitted five runs on a shared pipeline. One uses a long-context transformer (Mental-Longformer) over the whole thread. Three share a five-fold MentalBERT ensemble and vary only the score aggregator and the alert policy. The fifth is a classic two-channel character n-gram model with a linear SVM and gradient-boosted trees. An instruction-tuned LLM (GPT-4o-mini) tags every writing for topic and distress cues. Across the released results, our best run had the highest NDCG@100 at all four horizons (0.91 at 500 writings), though its bootstrap interval (0.86 to 0.95) overlaps the next team. P@10 and NDCG@10 reached the 1.00 ceiling that several teams shared. The decision results favored precision: our best submitted F1 was 0.71 (Run 4) against leaders at 0.82 to 0.83, with precision 0.95 and only three false positives. A post-hoc oracle threshold on the released labels—a diagnostic, not a submittable result—lifts the same probabilities to F1 0.82.

Keywords

CLEF eRisk, early depression detection, contextualized early risk detection, conversational context, risk ranking, long-context language models, MentalBERT

1. Introduction

Depression often goes undetected for long periods, and those who struggle frequently write about it online before ever speaking to a clinician; this makes social media a useful place to study early signals of mental health problems [1, 2]. The eRisk lab at CLEF has run shared tasks on this topic since 2017, rewarding systems that are accurate and fast, because a correct alert that arrives too late is worth much less [3, 4].

Task 2 of eRisk 2026 is the second edition of Contextualized Early Detection of Depression [5, 6, 7]. A structural change in this edition is that the task no longer presents a single user's posts in isolation. It releases the full Reddit discussion thread, so the target user is seen inside the conversation: the post they replied to, the replies they drew, and the order in which the exchange unfolded. Conversational context can change how a short reply should be read, and the task asks systems to use it.

We took part as team DeepCare, with five runs built on one shared pipeline that differ only in their encoder, score aggregator, or alert policy. Every round, each run must emit the two outputs the task evaluates separately: a binary decision and a continuous risk score. We treated these as separate problems, tuning the decision to be precise and early while leaving the score as a smoothed estimate of the per-writing probability.

Our best run produced the highest released NDCG@100 (0.91 at 500 writings), while our best decision F1 was 0.71 (Run 4): high precision (0.95) but limited recall under deliberately conservative policies. Our main contribution is an analysis of why this strong ranking did not carry over to a top decision

CLEF 2026 Working Notes, 21–24 September 2026, Jena, Germany

*Corresponding author.

gheorghe.nistor@s.unibuc.ro (G. Nistor); auban@fmi.unibuc.ro (A. Uban)

F1: both outputs come from a single per-user probability, and a post-hoc oracle-threshold diagnostic (Section 5.5) traces the gap to the precision-oriented alert policy rather than the underlying scores. We also use lightweight LLM topic-and-distress tags both as a Run 4 feature and as a lens on the system's errors (Section 5.3).

2. Background and Related Work

Research on detecting depression from text grew out of work showing that linguistic markers, such as increased first-person pronoun use and negative affect, track depressive states [1, 8]. The eRisk shared tasks turned this into a reproducible benchmark and added the early-detection angle: systems read a user's history in chronological chunks and must decide when they have seen enough to commit [3, 9, 10]. The standard cost function, the early-risk detection error (ERDE), penalizes wrong decisions and adds a growing penalty for correct positive alerts that arrive late [4]. Because ERDE is hard to compare across systems, later editions added a latency-weighted F measure that multiplies F1 by a speed factor derived from detection delay [11].

Recent eRisk entries split between two model families. One family comprises transformer encoders fine-tuned on mental-health text. MentalBERT and its relatives are BERT models [12] pretrained further on Reddit mental health communities, and they give a strong starting point for depression classification [13]. A known limitation is the 512-token window, which is short for a long thread. Long-context models such as Longformer extend the window to several thousand tokens with sparse attention [14], and domain-adapted long-context variants have been released for mental-health text [15]. The other comprises classic bag-of-words models. Character n-grams with TF-IDF (term frequency–inverse document frequency) weighting [16] and a linear support vector machine (SVM) remain competitive on eRisk depression data and are cheap to run, which makes them a useful baseline [10]. The first edition of this task, in 2025, drew on both families: the winning entry paired a psychiatric-scale-guided learnable screener with MentalBERT and LLM-based augmentation of the context-free training data [17], while others combined long-context encoders, time-aware losses, and LLM summarization [18] or scraped conversational data to match the test setting [19]. We use both families and add gradient-boosted trees [20] on top of hand-built features, with the classic components built on scikit-learn [21]. Our decision threshold relies on calibrated probabilities, for which we use Platt scaling [22] and isotonic calibration [23]; we evaluate the ranking output with NDCG [24], which depends only on the score order.

3. Task and Data

3.1. Task setup

In Task 2 the server releases the test cohort one round at a time. At round r each user has contributed up to r writings, embedded in their threads. After each round a system must return, for every user, a decision (0 or 1) and a risk score in $[0, 1]$. Once a system raises an alert for a user, that decision stands. The two outputs are scored on different objectives: the `decision` (binary, 0/1) by precision, recall, F1, ERDE_5, ERDE_{50}, $\mathrm{latency}_{TP}$, and F_{latency}; the `score` (continuous risk in $[0, 1]$) by the ranking metrics P@10, NDCG@10, and NDCG@100 at 1, 100, 250, and 500 writings.

The decision metrics combine accuracy and speed: ERDE_o penalizes both false positives and false negatives, and adds a latency cost to correct but late positive alerts. That cost grows with the number of writings k read before the alert, with its inflection set by o (5 or 50). The latency-weighted F measure is

$$F_{\mathrm{latency}} = F_1 \cdot \big(1 - \mathrm{median}\{\, \rho(k_u) : u \in TP \,\}\big), \qquad \rho(k) = -1 + \frac{2}{1 + e^{-\gamma(k-1)}}, \tag{1}$$

where TP is the set of true-positive users (those with decision and gold label both 1), k_u is the number of writings read before the alert for user u, and $\gamma = 0.0078$ (the organizers' p) [11]. A correct alert on the first writing gives no penalty; a very late one approaches a full penalty. The ranking metrics ignore

Table 1
Composition of the training (2025) and test (2026) collections from the released labels. During the live run our system processed 107,054 writings for the test cohort across 500 rounds.

Quantity	Train (2025)	Test (2026)
Subjects	909	523
positive (depression)	102 (11.2%)	91 (17.4%)
control	807 (88.8%)	432 (82.6%)
Median writings per subject (mean $\pm$ SD)	211 (307 $\pm$ 281)	181 (205 $\pm$ 160)

the binary decision and look only at the ordering produced by the score. We report NDCG@100, P@10, and NDCG@10 below.

3.2. Data

The collection consists of Reddit discussion threads. Each thread has an opening submission and a tree of comments. The target user can be the author of the opening post or one of the commenters, and the rest of the thread is the conversational context around them. The task releases the target's contributions one at a time; we call each release a *writing* (a target-user post together with its thread), and the evaluation horizons count writings. We used the data under the eRisk user agreement, for research only. For training we used the 2025 contextualized collection (909 subjects); for analysis after the competition we used the released golden truth for the 2026 test set (523 subjects); both collections are summarized in Table 1. The two come from different eRisk editions with disjoint subjects, so we train on 2025 and evaluate on 2026. During the live task the system processed 500 rounds over the 523 test subjects, each round adding one further writing per still-active subject. Because the 2025 release already carries the full conversational threads, our models train on the same contextualized format they meet at test time, unlike the first-edition systems that trained on context-free writings and had to synthesize context [17, 18]. The classes are imbalanced, so accuracy is a poor measure. The positive rate rises from 11.2% in training to 17.4% in the test set, so a threshold picked on the 2025 mix is likely too cautious for the 2026 mix, an effect we return to in Section 5.2.

4. System

4.1. Overview

All five runs share one pipeline, shown in Figure 1. Each round the client fetches the new batch of threads, normalizes the raw JSON, builds four payload views (a short context for the LLM tagger and one for each of the three encoders), and tags each writing. Payloads and tags are written to a local database so the run can resume after a crash. The encoders score the batch; five handlers (one per run) aggregate the scores per user, apply a per-run alert policy, and emit a decision and a score, which are posted back to the server. Because every input and output was persisted, we could later reconstruct exactly what the system saw and recompute all statistics offline.

4.2. Deriving two outputs from one probability

We decouple the two output objectives rather than the underlying signal: both outputs derive from the same per-user probability p, and what differs is how each is tuned, not the model behind it. The decision compares p against a threshold and fires only after a latency-aware alert policy is satisfied (the warm-up and consecutive-round knobs of Table 2). The score is an exponential moving average (EMA) of p that we clamp so it cannot go down:

$$s_t = \max\big(s_{t-1},\ (1-\alpha)\, s_{t-1} + \alpha\, p_t\big), \qquad \alpha \in \{0.15, 0.20, 0.30\}. \tag{2}$$

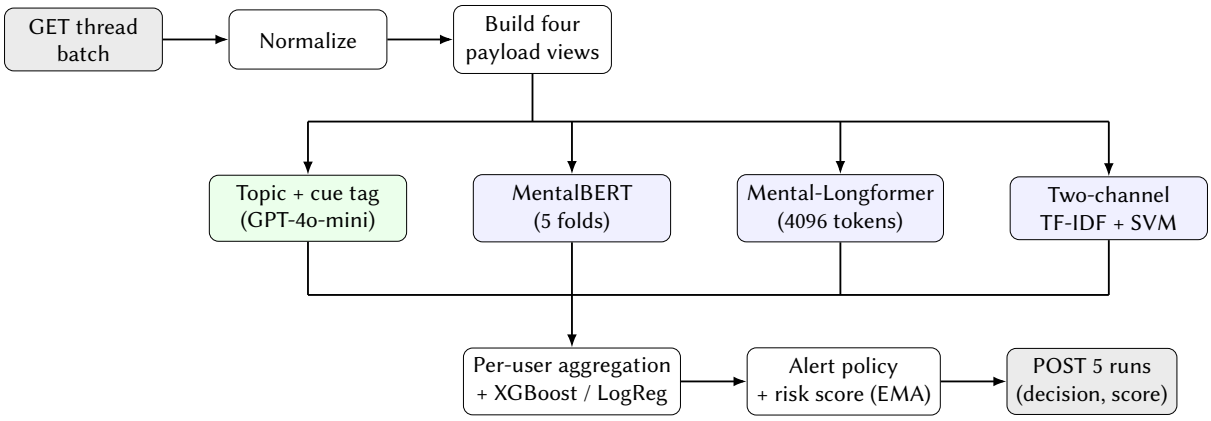

Figure 1: The shared pipeline. Data preparation runs along the top; the payload views feed the LLM tagger and the three encoder families in the middle; per-run aggregation, alert policy, and posting run along the bottom. Each round emits a 0/1 decision and a smoothed risk score for every user. The classic TF-IDF channel (Run 4) runs on CPU, independently of the transformer service.

We added the clamp for stability, so a user's rank would not swing between rounds; we report the clamp's effect on the ranking in Section 5.1.

4.3. Payload views and topic tagging

Before any model runs, each writing is tagged by the LLM tagger, a commercial instruction-tuned model (GPT-4o-mini, from the GPT-4o family [25]), with a coarse topic and a boolean "distress-cue" flag (the `has_mh_cues` field in our schema). The flag is set only for genuine expressions of psychological distress or mental illness, not for casual use like "this game is depressing". These tags are inexpensive features and also act as gates in two of the runs. Over the live run the model tagged all 107,054 writings. Most are not about mental health: 48% were tagged `other`, the largest single class, and only about 5% fell into the four mental-health topics combined. The distress flag fired on 8.1% of them, a coarse signal that differs between the classes on average (Section 5.3).

The encoder payloads differ by model. The MentalBERT payload is branch aware: it walks the reply chain from the target user up to the root and marks the target's own turns, the surrounding context, and whether the target is the original poster. The Longformer payload encodes the writing's whole thread as a single sequence of up to 4096 tokens, which keeps the full context in one pass instead of splitting it. The classic payload splits the text into two channels, the target user's own words and everyone else's, letting the model weight a word according to its author. The fourth view is a short, word-budgeted slice of the thread used only by the topic tagger. Every payload is assembled under per-section character budgets sized to the tokenizer window, so the target user's own text is never truncated.

All of these payloads preserve conversational structure rather than flattening it: each keeps the target's own words distinct from the surrounding turns. Two of Run 4's features go further and are purely contextual: the rates of concern and of hostility a user *receives*, computed only from what other people write to the target, a signal no target-only system can see. We included them in the expectation that a user's distress can draw visible concern from those who reply.

4.4. The five runs

Table 2 lists the runs. Run 0 is a single Mental-Longformer fine-tuned on one fold; it produces whole-thread scores mapped to a probability by logistic regression over the running mean of a user's per-round scores, and it was tuned for ERDE_5. Runs 1 to 3 share a five-fold MentalBERT ensemble. We pool the per-branch scores for a user by averaging over branches and folds, then a second-stage model maps the per-round history to a probability. Runs 1 and 2 share the same second stage (gradient-boosted trees over the mean of the three highest scores so far) and therefore emit an identical risk score; they differ

Table 2
The five submitted runs. "thr" is the alert threshold (Runs 0 and 1 decay from the first value to the second at 0.02 per round; Runs 2–4 use a fixed threshold), "cons" the consecutive rounds above threshold required, "warm" the warm-up rounds before any alert. Gates require the mean mental-health topic share to exceed a value.

Run	Encoder	Aggregator	Policy (thr / cons / warm / gate)	Tuned for
0	Mental-Longformer (4096)	LogReg, running mean	0.60→0.55 / 1 / 1 / none	ERDE_5
1	MentalBERT, 5 folds	XGBoost, top-3 mean	0.65→0.55 / 2 / 3 / none	F1
2	MentalBERT, 5 folds	XGBoost, top-3 mean	0.75 / 3 / 2 / mh>0.30	precision
3	MentalBERT, 5 folds	XGBoost, window 15	0.65 / 3 / 5 / mh>0.05	F_{latency}
4	char TF-IDF + SVM	XGBoost blend, 13 feats	0.70 / 1 / 0 / mh>0	F1 / F_{latency}

only in alert policy, Run 1 with an F1-oriented policy and Run 2 with a precision-oriented one behind a topic gate. Run 3 uses boosted trees over a sliding window of the last 15 scores and was tuned for the latency-weighted F. Run 4 is a non-transformer baseline: two character n-gram TF-IDF channels feeding a calibrated linear SVM, plus boosted trees over a 13-feature user vector, combined by a fixed equal-weight blend. It runs on CPU only and does not depend on the transformer service.

Run 4 uses 13 deliberately simple, recency-aware features: cumulative mental-health and drug-term lexicon hits in the user's own text and the mental-health term density; concern and hostility terms received from others; a token-weighted ratio of first-person (self-referential) language; the share of writings where the user is the thread's original poster; the topic and distress-cue rates from the tagger; the number of writings seen, the posting cadence, and a nocturnal-posting ratio (a proxy for circadian disruption [1]); and the mental-health hits over the last ten writings, to separate recent onset from long-standing patterns. At each round these values form one per-user vector for the gradient-boosted trees. On five-fold, user-stratified cross-validation this run reached a mean F1 of 0.83. Although we meant it as a lightweight baseline, Run 4 went on to give our strongest decision results on the 2026 test set (Section 5.2).

4.5. Alert policy

Each run combines the four policy knobs in Table 2. All require sustained evidence before an alert to avoid early false alarms on the large pool of control users, which gave high precision and limited recall; Section 5.5 quantifies the trade-off on the released labels.

4.6. Implementation details

The encoders are `mental-bert-base-uncased` (512-token window) and `mental-longformer-base-4096` (4096-token window), both fine-tuned for binary depression classification. MentalBERT is trained as a five-fold ensemble and the Longformer as a single fold. Both used a learning rate of 2×10^{-5}, an effective batch size of 32, and a weighted binary cross-entropy loss (positive-class weight 3.0) for the class imbalance, training up to four epochs per fold at the encoder's native maximum sequence length with early stopping on validation F1. Each fold's output is mapped to a probability by a Platt calibrator fitted on held-out data, and for MentalBERT the probabilities from the five folds are averaged. All cross-validation is user-stratified, so no subject appears in both train and validation folds. The thread tagger is `gpt-4o-mini` called at temperature 0 with JSON output, over a fixed taxonomy of twelve topics, four of which are mental-health related (depression, mental health, suicide watch, self harm). Run 4 calibrates the SVM with isotonic regression and uses gradient-boosted trees (depth 8, learning rate 0.03, about 600 trees) over the 13 features, blended with equal weights. Its lexicons are five small custom keyword lists (mental-health terms, drug terms, expressions of concern and of hostility, and first-person pronouns; Appendix A), not LIWC [26] or Empath [27]. The EMA coefficient α is 0.15 for Run 0, 0.20 for Runs 1 to 3, and 0.30 for Run 4. The nocturnal-posting feature uses UTC hours, since per-user time zones are unknown. Splits and shuffles use a fixed random seed (42 for fold construction;

Table 3
Ranking results (NDCG@100) by number of writings: the four next teams (top of the official table) and all five DeepCare runs. Our Run 0 is first at every horizon. P@10 and NDCG@10 were 1.00 from 100 writings on for all DeepCare runs (every horizon for Runs 0 and 4). Runs 1 and 2 emit the same ranking score (they differ only in alert policy) and share a row. The complete official results table appears in the extended overview [6].

Team	Run	1 w	100 w	250 w	500 w
DeepCare	**0**	**0.74**	**0.88**	**0.90**	**0.91**
MindAILab	0	0.69	0.86	0.89	0.90
NoirByte	0	0.72	0.86	0.88	0.89
INSA-Lyon	0	0.64	0.84	0.87	0.89
LCDAA	0	0.61	0.84	0.87	0.88
DeepCare	1, 2	0.65	0.79	0.83	0.85
DeepCare	3	0.67	0.79	0.84	0.85
DeepCare	4	0.65	0.82	0.83	0.85

XGBoost is trained with seed 0). The full tagger prompt and taxonomy are given in Appendix A.

5. Results and Analysis

The numbers below come from three sources. *Official* numbers are the organizers' preliminary results (Tables 3 and 5); *recomputed* numbers come from our reimplementation of the eRisk metric code applied to the per-round probabilities we stored during the live run; *post-hoc* numbers are sweeps over those probabilities against the released labels (Section 5.5). Our reimplementation reproduces the official precision, recall, and F1 exactly for all five runs, and matches the official NDCG@100 from 250 writings onward; at the first writing the two differ (0.66 recomputed versus 0.74 official) because near-identical scores make NDCG tie-sensitive. We therefore use the recomputation only for internally consistent ablations.

5.1. Ranking

Run 0 produced the strongest ranking among all submitted runs, leading the released NDCG@100 table at every horizon (Table 3, Figure 2): 0.74 at the first writing (the best in the field, though tie-sensitive; the closest team, NoirByte, reaches 0.72), then 0.88, 0.90, and 0.91 at 100, 250, and 500 writings, so most of the climb is already in place by 100 writings. A user-level bootstrap puts the 500-writing figure at 0.91 (95% CI 0.86 to 0.95), an interval that contains MindAILab's second-placed 0.90, so we do not read the lead as statistically clear. MindAILab and NoirByte also submitted for only part of the run (302 and 289 of the 500 rounds), which makes their scores less directly comparable; among the teams that submitted all 500 rounds, DeepCare leads at every horizon, with INSA-Lyon next (0.89 at 500 writings).

The top-heavy metrics tell a flatter story. P@10 and NDCG@10 reached 1.00 from 100 writings on for all five runs, and at every horizon for Runs 0 and 4. At the first writing the three MentalBERT runs scored 0.90 on P@10 and 0.93 on NDCG@10, and several other teams reached the same 1.00 ceiling. With 91 positives, far more than ten, the top of the ranking can be filled entirely with true positives, so these measures saturate; NDCG@100, which reaches deeper into the ranking, is the more informative one here.

Smoothing helped only the noisier runs. The clamp ablation in Table 4 makes this concrete: for Run 0 the raw per-round probability ranks at least as well as the clamped score we submitted (0.92 versus 0.91 at 500 writings), and a plain (non-clamped) EMA lands between or matches the two at every horizon, so neither smoothing step helps it. The MentalBERT runs behave differently—Run 3's raw NDCG@100 at 500 writings is 0.77 but 0.85 after the same smoothing, which does help when the per-round probabilities are unstable. On the raw probabilities the long-context run still orders users best (0.92, versus 0.84 for Run 1 and 0.77 for Run 3), though it also differs in model family, folds, aggregator, and calibration, so

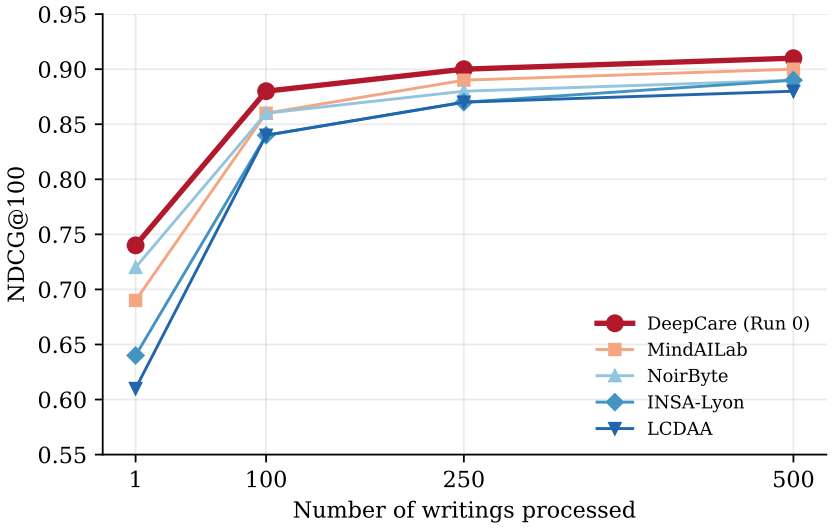

Figure 2: NDCG@100 against the number of writings processed, for our best run and the next four teams. Run 0 orders users well from the first round. The y-axis is truncated at 0.55. The leading teams' lines run close together, and the bootstrap interval in Section 5.1 indicates that our lead over the next team is not statistically clear.

Table 4
Clamp ablation on Run 0, NDCG@100 by writings, recomputed on the released labels. The clamped score is what we submitted; the raw probability and a plain (non-clamped) EMA are alternatives. For Run 0 the clamp does not improve ranking. Absolute values differ slightly from Table 3 at the early horizons (1 and 100 writings) because of tie-handling.

Ranking score	1 w	100 w	250 w	500 w
Clamped score (submitted)	0.66	0.87	0.90	0.91
Raw probability	0.66	0.89	0.92	0.92
Plain EMA (no clamp)	0.66	0.88	0.92	0.91

this is an association, not isolated proof that context length is the cause.

5.2. Decisions

The decision results split sharply by run (Table 5). Our alert policy was built for precision, and four of the five runs delivered it: Runs 0, 2, 3, and 4 reached precision of 0.92 or higher (Run 2 at 0.96), with recall between 0.25 and 0.57. The trade-off was a lower F1: our best was 0.71 (Run 4), against F1 leaders at 0.82 to 0.83 (HUGETIME 0.83; UNED-GELP, NoirByte, and MindAILab 0.82) [6]. Run 1, built to anchor F1, instead produced the opposite profile (recall 0.89, precision 0.54, F1 0.67): it raised 150 alerts, 69 of them false, where the precision-oriented runs raised 24 to 55 each with at most four false positives. The low recall of those precision-oriented runs follows from their deliberately conservative alert policies, compounded for the fixed-threshold runs by a base-rate shift: with the positive rate rising from 11.2% to 17.4% between 2025 and 2026, a threshold tuned on the 2025 mix sat too high for the test set and under-alerted. The same shift explains Run 4's fall from a mean F1 of 0.83 on 2025 cross-validation to 0.71 live. Run 1 failed the other way—its F1-oriented policy was simply too permissive on the 2026 data, over-alerting rather than suffering from a too-cautious threshold. Speed barely entered into it: Run 0 fired its true positives on the first writing (speed factor 1.00), and because every run's true positives arrived within about ten writings, the speed factor stayed near 1, so F_{latency} tracked F1 closely and the latency weighting did little to reorder the runs; Run 3, though tuned for F_{latency}, was edged out on that metric by Run 4 (0.70 versus 0.65).

Table 5
Decision results for all five DeepCare runs (official preliminary numbers). Best value per column in bold; lower is better for ERDE_5, ERDE_{50}, and $\mathrm{latency}_{TP}$ (the median number of writings to the first true-positive alert).

Run	P	R	F1	ERDE_5	ERDE_{50}	$\mathrm{latency}_{TP}$	F_{latency}
0	0.92	0.51	0.65	**0.11**	0.09	**1.0**	0.65
1	0.54	**0.89**	0.67	0.15	**0.06**	7.0	0.66
2	**0.96**	0.25	0.40	0.16	0.13	8.0	0.39
3	0.92	0.53	0.67	0.16	0.10	10.0	0.65
4	0.95	0.57	**0.71**	0.13	0.09	5.0	**0.70**

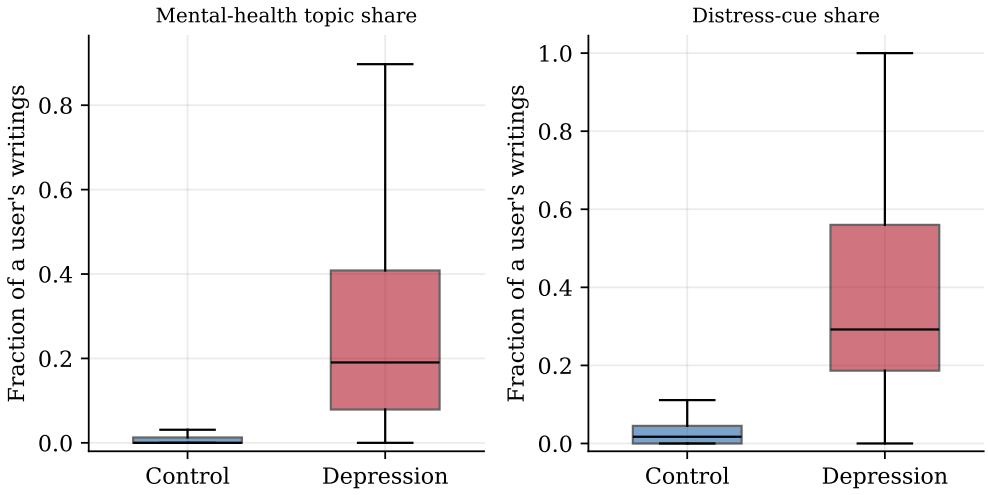

Figure 3: Per-user mental-health topic share (left) and distress-cue share (right), split by golden-truth label (depression $n = 91$, control $n = 432$). Computed from the 107,054 tags stored during the live run. Boxes show the interquartile range and median; whiskers extend to 1.5 times the interquartile range.

5.3. The topic signal

To check whether the lightweight topic tagging helped, we measured, per user, the share of their writings tagged as a mental-health topic and the share flagged with a distress cue, then split by the golden-truth label (Figure 3). Depression-positive users had a mean mental-health topic share of 0.28 against 0.017 for controls, and a mean distress-cue share of 0.38 against 0.042 (the figure shows medians and interquartile ranges, which are lower because the distributions are right-skewed). The topic tags alone thus show a clear class-level difference, and because the tagger reads the whole thread, that difference reflects the surrounding conversation as well as the target's own words. This motivated using the tags as a Run 4 feature and as a gate in Runs 2 and 3. Because the tagger's own accuracy was not separately validated, these tags are best read as a descriptive signal rather than a verified measure of topic or distress. Run 2's gate requires this mental-health share to exceed 0.30; because that share is right-skewed among positive users (median 0.19, mean 0.28), most of them fall below the cut and are screened out, a major reason its recall was only 0.25.

The same tags characterize most of Run 4's errors. Of the 91 positives, Run 4 missed 39. Those missed users had a median mental-health topic share of just 0.08, against 0.41 for the 52 it caught. The system misses depressed users whose writings are not openly about mental health, a setting where content-driven models have little signal to draw on. False positives were rare: only three controls were flagged, each with a raised distress-cue rate.

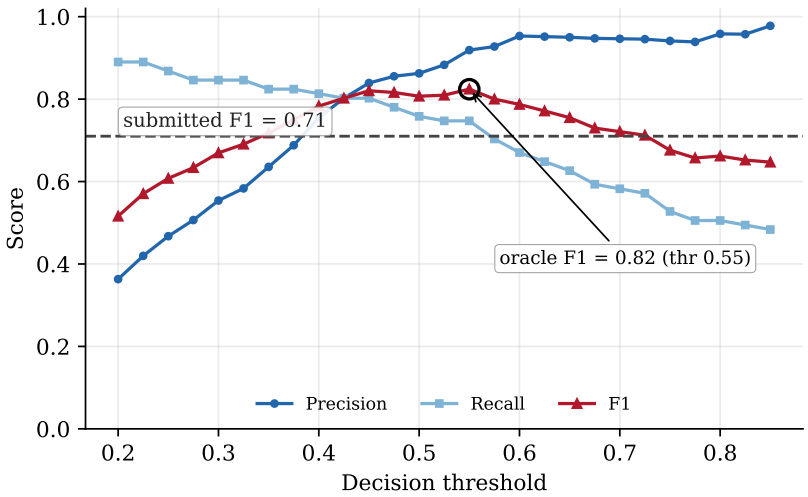

Figure 4: Post-hoc diagnostic. Precision, recall, and F1 for Run 4 as a single decision threshold is swept (from 0.20) over the stored per-round probabilities and scored on the released labels. F1 peaks at 0.82 (threshold 0.55, an oracle choice on the labels); the dashed line marks the 0.71 we submitted with the more cautious policy.

5.4. Efficiency

Orchestration and the CPU-only Run 4 ran on an Apple M3 Pro laptop; the transformers were fine-tuned and served on a Colab A100. The full five-run submission over the cohort finished in about nine hours (the organizers report 9 h 14 m), dominated by transformer encoding, payload preparation, and tagging.

5.5. Threshold sensitivity

The decision threshold has a large effect on the precision-recall balance. We took Run 4's stored per-round probabilities, applied a single fixed threshold (fire on the first crossing), and scored the result against the released labels (Figure 4). F1 peaks at 0.82 at a threshold of 0.55 (precision 0.92, recall 0.75), against the 0.71 we submitted with a more cautious policy; Run 0's raw probabilities reach F1 0.81 under the same sweep, though at a lower threshold. This is an oracle threshold selected on the test labels: an upper bound on what these probabilities can give, not a score the submitted system achieved or one comparable to teams that never saw the labels. It suggests that the stored probabilities supported higher recall under a less cautious threshold. The ranking results are independent of the threshold.

6. Conclusion and Future Work

We described DeepCare's entry in eRisk 2026 Task 2. Our system derives both the binary decision and the risk score from the same per-user probability, thresholding it for the decision and smoothing it for the score, across five runs from a long-context transformer to a classic character n-gram model. Run 0 had the highest released NDCG@100, 0.91 at 500 writings (first among the teams that submitted all 500 rounds), with P@10 and NDCG@10 of 1.00 at every horizon. The decision runs were precise but cautious: our best submitted F1 was 0.71 (Run 4), with precision up to 0.96 and few false positives. A post-hoc oracle threshold lifts Run 4 to F1 0.82 on the released labels, an upper bound rather than a submitted result.

Two limitations qualify these results. We did not run a target-user-only ablation, so the contribution of conversational context remains qualitative; and the headline NDCG@100 comes from a single Longformer fold, whose bootstrap interval (0.86 to 0.95) contains the next team's 0.90, so the lead is not statistically clear.

The data is Reddit only and not representative of clinical settings, where a false positive carries a real cost: this is a research prototype that outputs risk *rankings*, not a diagnostic tool to be used

on individuals without clinical oversight, and its LLM tags are noisy machine signals, not clinical judgments. We processed the data under the eRisk user agreement, for research only, sending thread text to GPT-4o-mini only for tagging. In future editions we would retune the thresholds for recall by calibrating to the expected base rate rather than the training mix, lean on the lightweight components (the classic run gave our best F1 and the tagger a strong signal on its own), and fold the conversation into the decision more directly, for example by weighting a user's turns by the distress of the posts they answer.

Acknowledgments

We thank the eRisk organizers for running the lab and for releasing the data, results, and golden truth that made this analysis possible.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4o-mini as a component of the described system, to tag discussion threads with a coarse topic and a distress-cue flag; this use is part of the method and is reported in Section 4. For the writing of this paper, the authors used Claude (Anthropic) as a writing assistant for grammar and spelling checks and for light wording suggestions. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] M. D. Choudhury, M. Gamon, S. Counts, E. Horvitz, Predicting depression via social media, in: Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media (ICWSM), 2013, pp. 128–137.

[2] S. C. Guntuku, D. B. Yaden, M. L. Kern, L. H. Ungar, J. C. Eichstaedt, Detecting depression and mental illness on social media: An integrative review, Current Opinion in Behavioral Sciences 18 (2017) 43–49.

[3] D. E. Losada, F. Crestani, J. Parapar, eRISK 2017: CLEF lab on early risk prediction on the internet: Experimental foundations, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction - 8th International Conference of the CLEF Association, CLEF 2017, volume 10456 of *Lecture Notes in Computer Science*, Springer, 2017, pp. 346–360.

[4] 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, volume 9822 of *Lecture Notes in Computer Science*, Springer, 2016, pp. 28–39.

[5] A. Perez, J. Parapar, X. Wang, F. Crestani, eRisk 2026: Tasks on symptoms ranking, contextual and conversational approaches for early mental health detection, in: Advances in Information Retrieval - 48th European Conference on Information Retrieval (ECIR 2026), Lecture Notes in Computer Science, Springer, 2026, pp. 233–241.

[6] 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, CEUR Workshop Proceedings, CEUR-WS.org, 2026. To appear.

[7] 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, Lecture Notes in Computer Science, Springer, 2026. To appear.

[8] G. Coppersmith, M. Dredze, C. Harman, Quantifying mental health signals in Twitter, in: Proceedings of the Workshop on Computational Linguistics and Clinical Psychology (CLPsych), 2014, pp. 51–60.

[9] 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, volume 14959 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 73–92.

[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, volume 16089 of *Lecture Notes in Computer Science*, Springer, 2025, pp. 242–265.

[11] F. Sadeque, D. Xu, S. Bethard, Measuring the latency of depression detection in social media, in: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM), ACM, 2018, pp. 495–503.

[12] 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 (NAACL-HLT), 2019, pp. 4171–4186.

[13] 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), European Language Resources Association, 2022, pp. 7184–7190.

[14] I. Beltagy, M. E. Peters, A. Cohan, Longformer: The long-document transformer, arXiv preprint arXiv:2004.05150 (2020).

[15] S. Ji, T. Zhang, K. Yang, S. Ananiadou, E. Cambria, J. Tiedemann, Domain-specific continued pretraining of language models for capturing long context in mental health, arXiv preprint arXiv:2304.10447 (2023).

[16] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Information Processing and Management 24 (1988) 513–523.

[17] Y. Zi, B. Wang, Y. Zhao, B. Qin, HIT-SCIR@erisk2025: Exploring the potential of a learnable screening model and risk post buffer-based framework for contextualized early prediction of depression on social media, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), Madrid, Spain, 9–12 September, 2025, volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025.

[18] M. Saad, A. U. Chaudhry, M. Abbas, F. Alvi, A. Samad, Contextualized early detection of depression – hybrid and time-aware approaches: HU at erisk task 2 2025, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), Madrid, Spain, 9–12 September, 2025, volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025.

[19] A. M. Marmol-Romero, M. Garcia-Vega, M. Á. Garcia-Cumbreras, A. Montejo-Ráez, SINAI at erisk@CLEF 2025: Transformer-based and conversational strategies for depression detection, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), Madrid, Spain, 9–12 September, 2025, volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025.

[20] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2016, pp. 785–794.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al., Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[22] J. C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: Advances in Large Margin Classifiers, MIT Press, 1999, pp. 61–74.

[23] B. Zadrozny, C. Elkan, Transforming classifier scores into accurate multiclass probability estimates, in: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2002, pp. 694–699.

[24] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, ACM Transactions on Information Systems 20 (2002) 422–446.
[25] OpenAI, GPT-4o system card, arXiv preprint arXiv:2410.21276 (2024).
[26] Y. R. Tausczik, J. W. Pennebaker, The psychological meaning of words: LIWC and computerized text analysis methods, Journal of Language and Social Psychology 29 (2010) 24–54.
[27] E. Fast, B. Chen, M. S. Bernstein, Empath: Understanding topic signals in large-scale text, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, ACM, 2016, pp. 4647–4657.

A. Topic tagging prompt, taxonomy, and Run 4 lexicons

The thread tagger calls `gpt-4o-mini` at temperature 0 with the system prompt below; the user turn is the thread payload of Section 4.3.

```
You are an expert moderator classifying a Reddit thread.
Return ONLY a valid JSON object matching this schema:
{
  "topic": "one of: [depression, mental_health, suicide_watch,
           self_harm, relationships, gaming, tech, cooking,
           sports, politics, finance, other]",
  "has_mh_cues": true/false
}
Rules for has_mh_cues:
- Set to true ONLY if there are genuine expressions of
  psychological distress, mental illness, therapy, or crisis.
- Set to false for colloquial uses like 'this game is
  depressing' or 'that idea is crazy'.
```

The taxonomy is the twelve topics above; four of them (depression, mental_health, suicide_watch, self_harm) count as mental-health. Run 4 uses five small case-insensitive regex lexicons matched on word boundaries:

- **mental-health** (14 stems): depress, suicid, anxi, bipolar, therap, antidepress, ssri, ptsd, panic, hopeless, worthless, insomnia, psychiatr, self-harm.
- **drugs** (15): prozac, lamictal, lamotrigine, sertraline, wellbutrin, lexapro, ketamine, xanax, adderall, zoloft, effexor, cymbalta, abilify, fluoxetine, citalopram.
- **concern received** (7 phrases): “are you ok”, “worried about you”, “here for you”, “please reach out”, “hotline”, “talk to a therapist”, “reach out”.
- **hostility received** (6 phrases): “grow up”, “stop whining”, “suck it up”, “loser”, “snowflake”, “crying baby”.
- **first-person pronouns** (5): i, me, my, mine, myself; used in the self-referential-language feature.