HUGETIME at eRisk 2026: Bucket-wise Evidence Pruning for Contextualized Early Depression Detection

Working Notes for the eRisk Task 2 at CLEF 2026

Ilho Shin[1], Hyunchul Kim[1] and Chanyoung Park[1,*]

[1]Data Science and Artificial Intelligence Laboratory (DSAIL), KAIST, Daejeon, Republic of Korea

Abstract

This paper describes the HUGETIME system submitted to Task 2 of the CLEF 2026 eRisk lab, Contextualized Early Detection of Depression. The task requires early depression-risk prediction from progressively released conversational threads, where target-user writings are mixed with surrounding context. Our approach combines target-centered candidate construction, writing-count-aware bucket processing, and bucket-specific evidence pruning before classification. The pruning stage uses two complementary strategies: a fixed Risk Centroid branch based on depression-risk prototype sentences and a learned Prototype Query Bank branch trained from subject-level supervision. The selected evidence is classified by bucket-specific local models and validation-selected ensemble decisions. In the official decision-based evaluation, HUGETIME ranked first by F_1, $ERDE_{50}$, and $F_{latency}$, achieving $P = 0.80$, $R = 0.87$, and $F_1 = 0.83$.

Keywords

Contextualized Early Depression Detection, Evidence Pruning, Prototype Query Bank, Bucket-wise Modeling

1. Introduction

Early risk prediction on the Internet aims to identify users who may be experiencing social or health-related risks before those risks become explicit or severe. The eRisk lab provides a benchmark setting for this problem through reusable test collections, shared evaluation protocols, and early-decision-oriented tasks [1, 2]. Depression detection has been a central eRisk scenario since the early depression and language-use collections [3]. This paper describes the HUGETIME system submitted to CLEF 2026 eRisk Task 2, Contextualized Early Detection of Depression. The task is the second edition of the contextualized setting introduced in eRisk at CLEF 2025 [4]; unlike earlier target-only settings, it provides full conversational threads, including discussion titles, target-user messages, other users' comments, and chronological interaction structure.

The task requires systems to classify each target user as either depressed or control from conversational evidence observed over time. During testing, evidence is released round by round, and the system must output a risk score $s_u^{(t)} \in [0, 1]$ and a binary decision $\hat{y}_u^{(t)} \in \{0, 1\}$ for target user u at round t. Evaluation considers both classification quality and timeliness, using Precision, Recall, and F_1, as well as delay-aware measures such as ERDE and latency-aware alternatives such as $F_{latency}$ [3, 5].

The contextualized setting is more realistic than isolated-post classification, but it also introduces substantial non-target text, repetitive context, and weakly relevant interaction noise. We therefore treat the task as both user-level depression classification and target-relevant evidence selection from noisy thread-level context. HUGETIME first constructs candidate evidence units centered on target-authored writings, assigns each sample to a writing-count bucket, and applies bucket-specific Top-K evidence pruning before classification. The system combines Risk Centroid Pruning, which uses a fixed risk direction for explicit lexical risk signals, with Prototype Query Bank Pruning, which learns multiple prototype directions for diverse risk-relevant expressions. Final predictions are produced by bucket-specific local models and validation-selected ensemble decisions.

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

*Corresponding author.

ilho0926@kaist.ac.kr (I. Shin); khchul@kaist.ac.kr (H. Kim); cy.park@kaist.ac.kr (C. Park)

2. Related Work

Our system builds on two lines of work: lexical SVM-based text classification and representation-based evidence selection. Linear SVMs over high-dimensional lexical features have long been effective for text categorization [6]. More directly, in the eRisk 2025 contextualized depression detection task, the ELiRF-UPV team used lightweight SVM-based models with TF-IDF character n-gram features, including character 4–5 gram representations [7]. Following this direction, our RiskSVM and ProtoSVM branches use the same TF-IDF character 4–5 gram LinearSVC backbone, while differing in the evidence selection strategy applied before classification.

The main difference from these lexical baselines is that our system explicitly separates evidence selection from downstream classification. Since the 2026 task provides full conversational threads, a system must decide which target-centered candidate evidence should be passed to the classifier. We therefore use sentence-encoder embeddings to score candidates before classification. This connects our approach to multiple instance learning, where supervision is available at the bag level rather than for every individual instance [8]. In our setting, each sample is treated as a bag of candidate embeddings, and candidate-level relevance is learned indirectly from subject-level labels.

Our prototype-query branch is inspired by learnable-query architectures. BLIP-2 uses a trainable Q-Former with learnable queries to extract information from frozen encoder representations [9]. We adapt this idea from vision-language representation learning to text evidence pruning: learned queries attend to candidate embeddings and become prototype directions for selecting risk-relevant evidence. Unlike BLIP-2, our goal is not multimodal alignment, but selecting diverse textual evidence from noisy conversational contexts before depression-risk classification.

3. Methodology

Figure 1 summarizes our pipeline. At each round, the system updates each target user's observed state, constructs target-centered candidates, assigns a writing-count bucket, applies evidence pruning, and produces a calibrated risk score with bucket-specific local models.

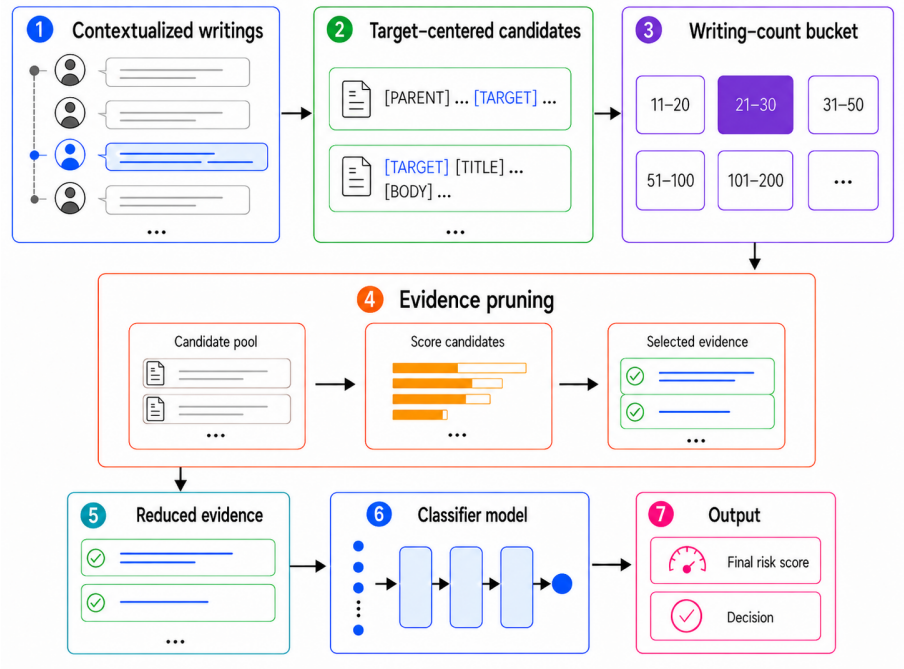

Figure 1: Overall pipeline of the system. The system constructs target-centered candidates from contextualized writings, assigns each sample to a writing-count bucket, applies bucket-specific evidence pruning, and produces final predictions using local models and bucket-wise ensemble decisions.

3.1. Candidate Construction

The contextualized task provides full discussion threads rather than isolated target-user writings. We therefore convert thread-level writings into text-level candidate evidence units centered on target-authored writings. This keeps the target user as the anchor of each candidate while retaining local conversational context when it is available.

Each candidate is represented as a marker-structured text sequence. We use explicit textual markers to indicate both the role and the content type of each segment. The role markers [PARENT] and [TARGET] indicate whether the segment comes from the surrounding context or from the target user, while [TITLE] and [BODY] indicate the title and body fields of a post when they are available.

For reply candidates, we concatenate the parent writing and the target-authored reply. Each segment is first marked by its role, and post-type segments additionally include title and body fields when available:

[ROLE] [TITLE] title [BODY] body,

where [ROLE] is either [PARENT] or [TARGET]. If the writing has no title, we use only its body:

[ROLE] [BODY] body.

For example, a reply candidate can be represented as:

[PARENT] [TITLE] title [BODY] parent_body [TARGET] [BODY] target_reply.

A target-authored post without a parent is represented as:

[TARGET] [TITLE] title [BODY] body.

The markers are not learned parameters; they are textual cues inserted to expose segment roles in the flattened candidate text. This distinction is important because a risk-related phrase in the parent context should not be treated in the same way as the same phrase written by the target user.

The marker-structured candidates are used by both the pruning and classification components. During pruning, candidate texts are encoded by the sentence encoder and scored by either Risk Centroid Pruning or Prototype Query Bank Pruning. During SVM-based classification, selected candidate texts are represented with TF-IDF character 4–5 gram features, so the markers become part of the character n-gram vocabulary. In contrast, the ProtoTransformer receives sentence-encoder embeddings of selected candidates; thus, marker information is reflected only through the candidate embeddings.

3.2. Writing-count-aware Bucket Processing

The amount of observed target-authored writing varies substantially across users and across online rounds. We therefore assign each sample to a writing-count bucket according to the cumulative number of target-authored writings observed at the current round. During online inference, this bucket assignment is updated dynamically as new writings are released, so the same target user may move from a shorter-writing bucket to a longer-writing bucket over time.

This design reflects the assumption that different writing-count ranges correspond to different evidence regimes. When only a small number of target-authored writings has been observed, depression-risk evidence may be sparse, and overly aggressive pruning can remove the few available risk-bearing candidates. When many target-authored writings have been observed, the number of candidates increases, but irrelevant or noisy thread context also accumulates. Therefore, each bucket uses a bucket-specific Top-K value during evidence pruning: smaller buckets retain a relatively larger fraction of evidence to avoid losing sparse signals, whereas larger buckets apply stricter selection to reduce noise and keep a smaller set of higher-quality candidates. Table 1 summarizes the bucket configuration.

The bucket assignment is used consistently throughout the later stages of the pipeline. In particular, the current bucket determines the Top-K pruning budget, the set of local model candidates, the validation-selected ensemble configuration, and the decision threshold used for the current sample.

Table 1
Writing-count buckets and Top-K values used for evidence pruning.

Bucket	Target-authored writings	Top-K
warmup	0–10	—
b010_020	11–20	10
b020_030	21–30	20
b030_050	31–50	30
b050_100	51–100	40
b100_200	101–200	60
b200_300	201–300	80
b300_500	301–500	100
b500_800	501–800	120
b800_inf	801+	200

3.3. Evidence Pruning

After candidate construction, the system applies evidence pruning to select target-relevant candidates before downstream classification. Since full conversational threads can produce many noisy or weakly relevant candidates, using all candidates directly may dilute the depression-risk signal. We therefore retain only the bucket-specific Top-K candidates according to pruning scores. We use two complementary branches: Risk Centroid Pruning, which relies on a fixed risk direction derived from predefined symptom prototypes, and Prototype Query Bank Pruning, which learns multiple prototype directions for more diverse risk-relevant evidence.

Let $\mathrm{Enc}(\cdot)$ denote the sentence encoder and let c_i be the marker-structured text of candidate i. For a sample with T candidates, we compute normalized candidate embeddings

$$e_i = \frac{\mathrm{Enc}(c_i)}{\|\mathrm{Enc}(c_i)\|_2}, \quad i = 1, \ldots, T,$$

where $e_i \in \mathbb{R}^d$. Both pruning branches operate on the same embedding set $E = \{e_i\}_{i=1}^{T}$ and compute a candidate-level relevance score for each e_i. The output of this stage is not a final depression prediction, but a reduced evidence set that is passed to the local classification models described in Section 3.5.

3.3.1. Risk Centroid Pruning

Risk Centroid Pruning uses a single fixed risk direction to score candidate embeddings. For this branch, we construct a risky sentence lexicon from seed categories that represent major clinical and behavioral symptoms of depression. Starting from these symptom-level seeds, we use an LLM to draft diverse self-report style risk sentences, and then manually review and lightly post-process the generated sentences. The resulting lexicon is used as a fixed set of sentence-level risk prototypes.

Let ℓ_j be the j-th lexicon prototype sentence. Using the same sentence encoder as above, we compute normalized lexicon embeddings

$$z_j = \frac{\mathrm{Enc}(\ell_j)}{\|\mathrm{Enc}(\ell_j)\|_2}, \quad j = 1, \ldots, J.$$

The risk centroid r is then defined as the normalized mean of the lexicon prototype embeddings:

$$r = \frac{\frac{1}{J}\sum_{j=1}^{J} z_j}{\left\|\frac{1}{J}\sum_{j=1}^{J} z_j\right\|_2}.$$

For each candidate embedding e_i, the risk score is computed by dot product:

$$s_i^{risk} = e_i^{\top} r.$$

Since both e_i and r are normalized, this dot product corresponds to cosine similarity in the sentence-embedding space. Candidates with the highest s_i^{risk} values are retained according to the bucket-specific Top-K budget.

This branch is designed as a high-precision selector for explicit depression-risk signals. However, because it relies on a single fixed risk direction, it may miss indirect, implicit, or lexically diverse expressions of depression risk that do not align well with the centroid. This limitation motivates the second pruning branch, Prototype Query Bank Pruning.

3.3.2. Prototype Query Bank Pruning

Prototype Query Bank Pruning replaces the single fixed centroid with multiple learned prototype directions. Instead of scoring candidates against one predefined risk direction, this branch uses a bank of learnable query vectors that can specialize to different types of risk-relevant evidence. The goal is to complement Risk Centroid Pruning by selecting candidates that may express depression risk in more diverse or less explicit ways.

Let $Q_{b,i}^{\star} = \{q_{i,m}^{\star}\}_{m=1}^{M}$ denote the refined query bank produced for sample i in bucket b by the trained Query Transformer. Each query vector $q_{i,m}^{\star} \in \mathbb{R}^d$ is the output of the final Query Transformer layer and represents a prototype direction conditioned on the candidate embedding bag of the current sample. For candidate embedding $e_{i,t}$, we compute the scaled candidate-query matching score as

$$\ell_{i,t,m} = \frac{e_{i,t}^{\top} q_{i,m}^{\star}}{\sqrt{d}}.$$

The prototype score of candidate t in sample i is defined as the maximum matching score over the query bank:

$$s_{i,t}^{proto} = \max_{m=1,\ldots,M} \ell_{i,t,m}.$$

This max operation allows a candidate to receive a high score if it strongly matches any learned prototype direction.

Candidates are ranked by $s_{i,t}^{proto}$, and the highest-scoring candidates are selected according to the bucket-specific Top-K budget. As with Risk Centroid Pruning, this branch is an evidence selector rather than a final classifier; its output is later consumed by downstream local models.

3.4. Prototype Query Bank Training

Prototype Query Bank Training aims to learn prototype directions for evidence pruning when candidate-level labels are unavailable. The query bank is not trained as a final depression classifier; instead, it is optimized as a candidate-level evidence selector using subject-level supervision, so that candidates from positive samples receive higher prototype relevance scores than candidates from negative samples. Following the bucket-wise design in Section 3.2, we train a separate query bank for each non-warmup writing-count bucket. Figure 2 illustrates the overall training flow, from subject-level candidate embedding bags to query refinement and ranking-based query learning.

3.4.1. Candidate Bags and Bag Scores

For a given bucket, each training sample is represented as a bag of candidate embeddings. Let

$$X_i = \{e_{i,t}\}_{t=1}^{T_i}, \quad e_{i,t} \in \mathbb{R}^d$$

denote the candidate embedding bag for sample i, where T_i is the number of candidates in the sample. The supervision is available only at the subject level, with label $y_i \in \{0, 1\}$, and no candidate-level relevance labels are provided. This setting naturally corresponds to a multiple-instance learning view.

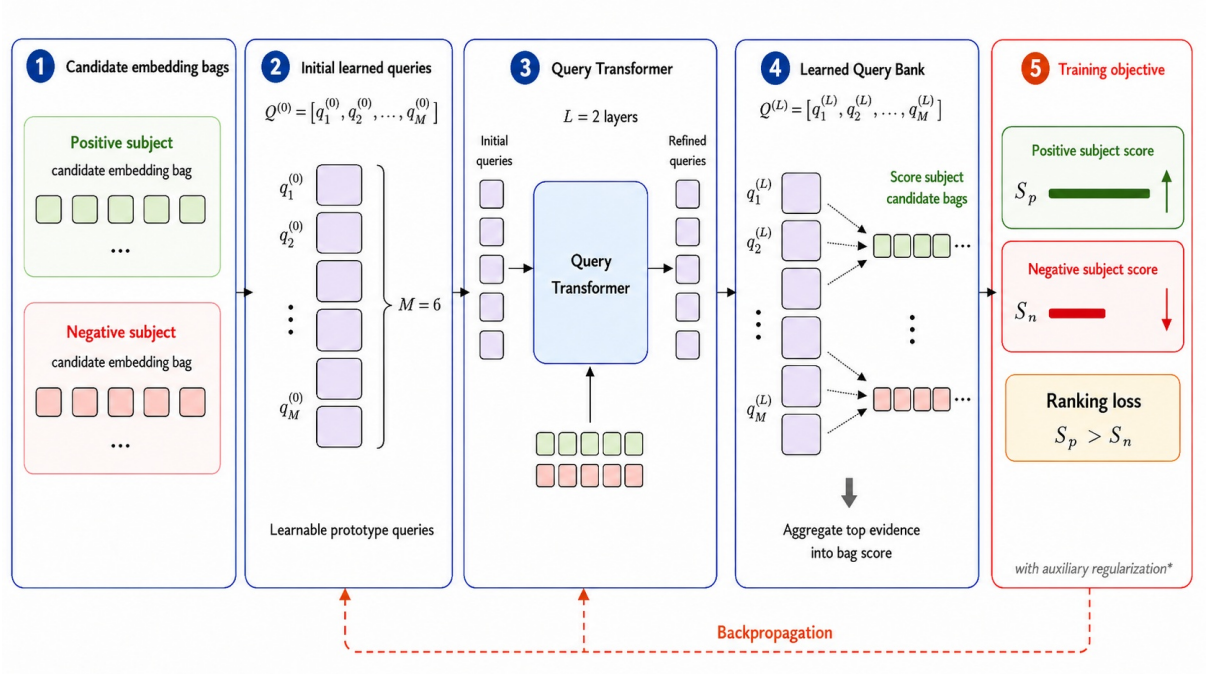

Figure 2: Prototype Query Bank training. Each subject is represented as a candidate embedding bag, and learnable prototype queries are refined by the Query Transformer. The learned query bank assigns bag-level scores through candidate-query matching and top-evidence aggregation. The ranking objective encourages positive-subject scores S_p to exceed negative-subject scores S_n, while auxiliary regularization reduces query collapse.

Using the query-bank scoring rule for bucket b, each candidate embedding $e_{i,t}$ receives a prototype score $s_{i,t}^{proto}$, as defined in Section 3.3.2. We aggregate candidate-level scores into a bag-level score by averaging the highest prototype scores using a fixed training-time pooling width K_{bag}:

$$S_i = \text{MeanTopK}_{K_{\text{bag}}} \left(\{ s_{i,t}^{proto} \}_{t=1}^{T_i} \right).$$

Here, $\text{MeanTopK}_{K_{\text{bag}}}(\cdot)$ denotes the average of the largest $\min(K_{\text{bag}}, T_i)$ values. This K_{bag} value is distinct from the bucket-specific pruning budget k_b: K_{bag} is used only to compute the training bag score for the ranking objective, whereas k_b determines how many candidates are retained during the actual pruning stage. The bag score S_i summarizes the strength of high-scoring risk-relevant evidence in sample i and is used by the ranking term described in Section 3.4.3.

3.4.2. Query Transformer

The Query Transformer refines learnable prototype queries using candidate embedding bags. It is trained as part of the query-bank learning module and later used with fixed parameters to produce refined prototype directions for Prototype Query Bank Pruning. It is not a final depression classifier.

$$Q_b^{(0)} = \{ q_m^{(0)} \}_{m=1}^{M}, \quad q_m^{(0)} \in \mathbb{R}^d,$$

where M is the number of prototype queries. During training, these queries are refined by attending to the candidate embedding bag of each sample, and the resulting objective updates the query parameters.

Because samples in the same bucket can still contain different numbers of candidates, mini-batch training uses padding and attention masks. For a batch, candidate bags are padded to the maximum candidate length in the batch, and a binary attention mask indicates valid candidate positions. The mask is used so that padded candidates do not contribute to query-to-candidate cross-attention, bag-score pooling, or query-usage regularization.

Each transformer layer applies query self-attention, query-to-candidate cross-attention, and a feed-forward network with residual connections and layer normalization. Query self-attention allows the prototype queries to interact with one another, while cross-attention lets the queries extract information from valid candidate embeddings:

$$Q_{b,i}^{(\ell)} = \mathrm{TransformerLayer}\left(Q_{b,i}^{(\ell-1)}, X_i, A_i\right), \quad \ell = 1, \ldots, L,$$

where A_i is the attention mask for sample i. The output of the final Query Transformer layer, $Q_{b,i}^{(L)}$, is used to compute candidate-query matching scores for the bag-level objective. After training, the learned query parameters and the trained Query Transformer are fixed. During Prototype Query Bank Pruning, we run the fixed Query Transformer with the candidate bag and use its final-layer output as the refined query bank $Q_{b,i}^{\star} = Q_{b,i}^{(L)} = \{q_{i,m}^{\star}\}_{m=1}^{M}$. Thus, after training, both the query parameters and the Query Transformer are fixed; they are used only to produce refined prototype directions for pruning, not to make the final depression prediction.

This design follows the learnable-query idea of Q-Former, but uses it for textual evidence pruning rather than vision-language alignment.

3.4.3. Training Objective

The query bank is trained with a weighted objective that combines a bag-level ranking term and query regularization terms:

$$\mathcal{L} = \lambda_{rank}\mathcal{L}_{rank} + \lambda_{div}\mathcal{L}_{div} + \lambda_{cov}\mathcal{L}_{cov} + \lambda_{dom}\mathcal{L}_{dom}.$$

The ranking term encourages positive bags to obtain higher scores than negative bags, while the remaining regularization terms control the geometry and usage of the learned queries.

For the ranking term, we use a margin-based objective over positive and negative bags:

$$\mathcal{L}_{rank} = \frac{1}{|\mathcal{P}|} \sum_{(p,n)\in\mathcal{P}} \max(0, \gamma - S_p + S_n),$$

where S_p and S_n are the bag scores of positive and negative samples, γ is a margin, and $\mathcal{P}$ is the set of positive-negative bag pairs in a batch. This term provides the main learning signal for selecting risk-relevant evidence: positive bags are encouraged to rank above negative bags.

The diversity term discourages different final-layer query vectors from collapsing to similar directions. Let $\tilde{q}_{i,m}^{(L)} = q_{i,m}^{(L)} / \|q_{i,m}^{(L)}\|_2$ denote the normalized query vector. We penalize off-diagonal query similarities:

$$\mathcal{L}_{div} = \frac{1}{|\mathcal{B}|M(M-1)} \sum_{i\in\mathcal{B}} \sum_{m\neq m'} \left((\tilde{q}_{i,m}^{(L)})^{\top} \tilde{q}_{i,m'}^{(L)}\right)^2.$$

This term encourages the query bank to maintain distinct prototype directions.

However, geometric diversity alone does not guarantee balanced query usage. Even if the query vectors point in different directions, most candidates may still be assigned to only one or two queries. Therefore, we add coverage and dominance regularization terms to control the actual candidate-query assignment behavior. For query-usage regularization, let $\ell_{i,t,m}$ denote the matching score between candidate $e_{i,t}$ and query m, computed from the training-time query representation used in this objective. To measure query usage, we compute soft assignment mass from these candidate-query scores:

$$a_{i,t,m} = \frac{\exp(\ell_{i,t,m}/\tau_{\text{assign}})}{\sum_{m'=1}^{M} \exp(\ell_{i,t,m'}/\tau_{\text{assign}})}.$$

where τ_{assign} is the query-assignment temperature. Let $v_{i,t} \in \{0, 1\}$ indicate whether position t is a real candidate. The batch-level usage of query m is computed over valid positions as

$$u_m = \frac{\sum_{i\in\mathcal{B}} \sum_{t=1}^{T_{\max}} v_{i,t} a_{i,t,m}}{\sum_{i\in\mathcal{B}} \sum_{t=1}^{T_{\max}} v_{i,t}}.$$

The coverage term softly counts queries whose usage exceeds a threshold:

$$g_m = \sigma\left(\frac{u_m - \theta_{cov}}{\tau_{cov}}\right), \quad C = \sum_{m=1}^{M} g_m.$$

We penalize insufficient coverage when the active query count is below a target value:

$$\mathcal{L}_{cov} = \max(0, C_{target} - C).$$

The dominance regularization term prevents a single query from monopolizing the assignment mass. Let

$$\rho_m = \frac{u_m}{\sum_{m'=1}^{M} u_{m'}}, \quad \rho_{\max}^{obs} = \max_m \rho_m.$$

We penalize the largest observed query share when it exceeds a predefined limit:

$$\mathcal{L}_{dom} = \max(0, \rho_{\max}^{obs} - \rho_{limit}).$$

Together, the coverage and dominance terms encourage the query bank to use multiple prototype directions rather than relying on a single dominant query. After training, the learned query parameters and the Query Transformer are fixed and used in Prototype Query Bank Pruning to produce refined prototype directions for scoring and selecting candidates.

3.5. Local Models and Bucket-wise Ensemble Decision

After evidence pruning, the reduced candidate set is passed to local classification models that produce depression-risk scores. For each writing-count bucket, we train two SVM-based lexical models with the same classifier backbone but different evidence sources. The first model, RiskSVM, uses the evidence texts produced after Risk Centroid Pruning. The second model, ProtoSVM, uses the evidence texts produced after Prototype Query Bank Pruning. In both cases, the selected Top-K candidate texts are concatenated into a single sample-level evidence text before SVM training. The SVM therefore does not process candidates as separate instances; instead, each sample is represented as one evidence text. This text is then encoded with TF-IDF character word-boundary 4–5 gram features. We train a class-balanced LinearSVC model on these lexical features and obtain probability scores through sigmoid calibration.

In addition to the SVM-based models, we train representation-based ProtoTransformer models on the candidate embeddings selected by Prototype Query Bank Pruning. For each sample, the selected candidates form an embedding sequence

$$X = [e_1, e_2, \ldots, e_{k_b}], \quad e_i \in \mathbb{R}^d,$$

where k_b is the Top-K budget for the current writing-count bucket. The ProtoTransformer applies a TransformerEncoder to this selected candidate sequence and maps the resulting sequence representation to a depression-risk score. We use two local model variants: ProtoTransformer-L2, with two TransformerEncoder layers, and ProtoTransformer-L3, with three TransformerEncoder layers.

The final decision is made separately for each writing-count bucket. For every bucket, we evaluate the available local models on validation data and select either a single model or a pair ensemble. When a single model is selected, its risk score is used as the final risk score. When a pair ensemble is selected, the final score is computed as a weighted average of the two selected model scores:

$$s_b(x) = \alpha_b s_{b,1}(x) + (1 - \alpha_b) s_{b,2}(x),$$

where $s_{b,1}(x)$ and $s_{b,2}(x)$ are the two model scores selected for bucket b, and α_b is the bucket-specific ensemble weight chosen on internal validation data. The resulting score $s_b(x)$ is then compared with a bucket-specific decision threshold τ_b:

$$\hat{y}(x) = \begin{cases} 1, & s_b(x) \geq \tau_b, \\ 0, & s_b(x) < \tau_b. \end{cases}$$

4. Experiment

4.1. Setup

All splits were performed at the subject level. The full subject set contained 909 subjects, divided into 727 training subjects and 182 held-out final validation subjects. The final validation subjects were not used for Query Bank training, local model training, ensemble selection, or threshold tuning. Within the training partition, Prototype Query Bank training used 581 training subjects and 146 validation subjects. For embedding-based components, we used `Alibaba-NLP/gte-base-en-v1.5`, which produces $d = 768$-dimensional sentence embeddings [10, 11].

For RiskSVM and ProtoSVM, we trained bucket-specific SVM models using TF-IDF character word-boundary 4–5 gram features. The vectorizer used lowercase normalization, `min_df` $= 3$, `max_features` $= 300000$, and sublinear term frequency scaling. The classifier was a class-balanced LinearSVC with $C = 1.0$, squared-hinge loss, `tol` $= 10^{-4}$, and `max_iter` $= 20000$. Since LinearSVC does not directly output probabilities, we used 3-fold sigmoid calibration to convert decision scores into probability-like risk scores.

For Prototype Query Bank training, we trained one query bank per non-warmup bucket. We used $M = 6$ learnable queries, $d = 768$-dimensional query vectors, two Query Transformer layers, four attention heads, feed-forward multiplier 4, dropout 0.1, bag-pooling width $K_{\text{bag}} = 35$, margin $\gamma = 0.2$, and query-assignment temperature 0.05. The objective weights were $\lambda_{rank} = 1.0$, $\lambda_{div} = 0.20$, $\lambda_{cov} = 0.10$, and $\lambda_{dom} = 0.10$. Query-usage regularization used $C_{target} = 5$, $\theta_{cov} = 0.20$, $\tau_{cov} = 0.01$, and $\rho_{limit} = 0.50$. Models were optimized with AdamW using learning rate 10^{-4}, weight decay 10^{-4}, batch size 16, four positive samples per batch, and maximum 100 epochs. Early stopping used patience 15. The default patience metric was validation AUC; for the `b800_inf` bucket, validation loss was used with a minimum of 20 epochs.

For ProtoTransformer local models, we trained bucket-specific models using the candidate embedding sequences selected by Prototype Query Bank Pruning. ProtoTransformer-L2 and ProtoTransformer-L3 used the same training configuration except for the number of TransformerEncoder layers, with two and three layers, respectively. We used $d = 768$-dimensional input embeddings, four attention heads, dropout 0.3, batch size 16, maximum 150 epochs, early stopping patience 20, learning rate 10^{-5}, and weight decay 0.01. Final model selection was performed separately for each writing-count bucket using only internal validation data, and decision thresholds were selected by maximizing F_1 on the same internal validation split.

4.2. Results

Table 2 reports the official decision-based evaluation result of our submitted run. The decision-based evaluation considers both classification quality and timeliness, including F_1, $ERDE_5$, $ERDE_{50}$, and the latency-aware $F_{latency}$ score [1]. ERDE is an early-detection error measure that penalizes delayed true positive alerts, with the penalty increasing as more user writings are processed before the alert is issued [3]. Thus, the decision-based setting rewards systems that not only make correct predictions, but also issue positive alerts early enough to be useful in an online monitoring scenario. HUGETIME achieved the highest F_1 score among the submitted runs, with $P = 0.80$, $R = 0.87$, and $F_1 = 0.83$. The system also achieved $ERDE_{50} = 0.05$ and $F_{latency} = 0.81$, indicating that the final binary decisions were effective under standard classification criteria and selected latency-aware measures.

Table 2
Official Task 2 decision-based evaluation result for HUGETIME. Ranks in parentheses are shown for selected official metrics. Lower is better for ERDE, and higher is better for F_1 and $F_{latency}$.

Team	Run	P	R	F_1	$ERDE_5$	$ERDE_{50}$	$latency_{TP}$	speed	$F_{latency}$
HUGETIME	0	0.80	0.87	**0.83 (1)**	0.15 (30)	**0.05 (1)**	9.00	0.97	**0.81 (1)**

Table 3 shows the official ranking-based evaluation result. In this evaluation, systems are assessed by sorting users according to the submitted risk scores at different evidence release points and computing ranking metrics such as P@10 and NDCG [1]. Our ranking-based performance was lower than the decision-based performance, especially at the 1-writing point. This was partly caused by an implementation error in the warmup-bucket handling in the final submission. Specifically, for five subjects whose total observed target-authored writings remained within the warmup bucket, i.e., 10 writings or fewer even at the final round, the submitted score and label were set unconditionally to 0.999 and positive, although only one of them was actually positive. This submission issue also introduced false positive decisions in the decision-based evaluation, but its effect was especially harmful for ranking-based evaluation because these low-evidence subjects were placed near the top of the risk ranking.

Table 3
Official Task 2 ranking-based evaluation result for HUGETIME.

		1 writing			100 writings			250 writings			500 writings		
Team	Run	P@10	NDCG@10	NDCG@100	P@10	NDCG@10	NDCG@100	P@10	NDCG@10	NDCG@100	P@10	NDCG@10	NDCG@100
HUGETIME	0	0.70	0.81	0.32	0.60	0.44	0.73	0.60	0.44	0.79	0.60	0.44	0.79

Despite this warmup-bucket issue, the ranking results after more evidence became available suggest that the submitted scores still captured part of the intended risk ordering. In particular, NDCG@100 increased from 0.32 at 1 writing to 0.73 at 100 writings and 0.79 at both 250 and 500 writings. This pattern suggests that the score ranking became more reliable once the system moved beyond the extremely sparse evidence regime. Overall, the official results suggest that the proposed bucket-wise evidence pruning and local decision procedure were effective for binary early-risk decisions, while also highlighting that score calibration and warmup-stage handling are critical for ranking-based evaluation.

5. Conclusion

This paper presented HUGETIME, our system for CLEF 2026 eRisk Task 2 on contextualized early detection of depression. We treat the task not only as user-level classification, but also as evidence selection over noisy conversational threads. The system constructs target-centered candidate units, assigns each sample to a writing-count bucket, and applies bucket-specific evidence pruning before downstream classification.

The proposed pruning strategy combines two complementary branches. Risk Centroid Pruning provides a fixed prior for explicit depression-risk signals using a sentence-level risky lexicon, while Prototype Query Bank Pruning learns multiple prototype directions from subject-level supervision. The learned query bank is trained without candidate-level labels, using a ranking objective that encourages positive-subject candidate bags to receive higher scores than negative-subject bags. The selected evidence is then consumed by bucket-specific local models and validation-selected ensemble decisions.

The official decision-based results demonstrate the strength of this bucket-wise evidence selection and local decision framework for binary early-risk prediction. HUGETIME ranked first by F_1, with $P = 0.80$, $R = 0.87$, and $F_1 = 0.83$, and also achieved first-ranked $ERDE_{50}$ and $F_{latency}$ scores. At the same time, the ranking-based results exposed the sensitivity of score ordering to warmup-stage handling. In particular, the submission logic for subjects that remained in the warmup bucket negatively affected early ranking metrics. Future work should therefore improve score calibration in extremely sparse evidence regimes and develop warmup-stage decision logic that is more robust to limited evidence.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00335098), and by National Research Foundation of Korea (NRF) funded by Ministry of Science and ICT (RS-2022-NR068758).

Declaration on Generative AI

During the preparation of this work, we used generative AI tools for language polishing and drafting support for figures and auxiliary resources. The proposed methods, experiments, analyses, and final manuscript were reviewed and edited by the authors, who take full responsibility for the publication's content.

References

[1] 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.

[2] 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.

[3] D. E. Losada, F. Crestani, A test collection for research on depression and language use, in: N. Fuhr, P. Quaresma, T. Gonçalves, B. Larsen, K. Balog, C. Macdonald, L. Cappellato, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction, volume 9822 of *Lecture Notes in Computer Science*, Springer, Cham, 2016. URL: https://doi.org/10.1007/978-3-319-44564-9_3. doi:10.1007/978-3-319-44564-9_3.

[4] J. Carrillo-de Albornoz, G. Faggioli, N. Ferro, A. García Seco de Herrera, J. Gonzalo, L. Plaza, J. Mothe, F. Piroi, P. Rosso, D. Spina, Report on the 16th conference and labs of the evaluation forum (clef 2025): Experimental ir meets multilinguality, multimodality, and interaction, ACM SIGIR Forum 59 (2025) 1–20. URL: https://doi.org/10.1145/3799914.3799930. doi:10.1145/3799914.3799930.

[5] 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 '18, Association for Computing Machinery, 2018, pp. 495–503.

[6] T. Joachims, Text categorization with support vector machines: Learning with many relevant features, in: Proceedings of the 10th European Conference on Machine Learning, ECML 1998, Springer, 1998, pp. 137–142. doi:10.1007/BFb0026683.

[7] A. Casamayor Segarra, V. Ahuir, A. Molina, L.-F. Hurtado, ELiRF-UPV at eRisk 2025: New approaches to the detection and early detection of symptoms and signs of depression, in: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), volume 4038 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025. URL: https://ceur-ws.org/Vol-4038/paper_122.pdf.

[8] M. Ilse, J. Tomczak, M. Welling, Attention-based deep multiple instance learning, in: Proceedings of the 35th International Conference on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, PMLR, 2018, pp. 2127–2136. URL: https://proceedings.mlr.press/v80/ilse18a.html.

[9] J. Li, D. Li, S. Savarese, S. C. H. Hoi, BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, in: Proceedings of the 40th International Conference on Machine Learning, volume 202 of *Proceedings of Machine Learning Research*, PMLR, 2023, pp. 19730–19742. URL: https://proceedings.mlr.press/v202/li23q.html.

[10] X. Zhang, Y. Zhang, D. Long, W. Xie, Z. Dai, J. Tang, H. Lin, B. Yang, P. Xie, F. Huang, M. Zhang, W. Li, M. Zhang, mGTE: Generalized long-context text representation and reranking models for multilingual text retrieval, 2024. URL: https://arxiv.org/abs/2407.19669. arXiv:2407.19669.

[11] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, M. Zhang, Towards general text embeddings with multi-stage contrastive learning, 2023. URL: https://arxiv.org/abs/2308.03281. arXiv:2308.03281.