SECEMBEDEVAL: A COMPREHENSIVE BENCHMARK FOR EVALUATING TEXT EMBEDDINGS IN CYBERSECURITY

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce SecEmbedEval, a benchmark for evaluating cybersecurity text embeddings. SecEmbedEval contains over 17,000 task instances in Chinese and English, built from public cybersecurity sources through source-aware preprocessing, large language model (LLM)-assisted construction, and blind verification. It covers nine tasks in five categories: semantic similarity, retrieval, classification, clustering, and efficiency. To support aggregated evaluation, we propose two metrics. SEE-Score computes a normalized average across six task measures. Transfer Discordance (TD) quantifies the share of model pairs whose ranking order differs between public leaderboards and our in-domain benchmark. We evaluate 12 representative embedding models. General-domain standing does not transfer (TD $= 0.33$). Frozen cybersecurity-pretrained encoders trail contrastively trained general-purpose models even under English-only controls. No evaluated model, including the strongest commercial system, can ground bare CVE identifiers to their advisories, and clustering remains challenging for every model.

1 INTRODUCTION

Security operations run on large volumes of heterogeneous text. Teams continuously ingest vulnerability advisories, vendor bulletins, threat reports, exploit databases, and social-media intelligence. Deduplicating near-identical advisories, resolving aliases of the same malware family or product, grouping reports by campaign, and tagging records with CWE or ATT&CK labels all reduce to similarity, clustering, and retrieval over security text (Reimers & Gurevych, 2019; Wang et al., 2022; Jiang et al., 2024; Zhang et al., 2025b; Günther et al., 2025; Kolodin et al., 2026). Downstream applications depend on embeddings just as heavily. Intelligence-assisted question answering must recall the right advisory to ground an answer, sometimes starting from a bare CVE identifier, and code auditing matches a change or bug report against known vulnerabilities. In each of these workflows the embedding model bounds recall, precision, and analyst time.

Choosing that model remains guesswork. Cybersecurity text is terminology-dense and heterogeneous, and general-domain benchmarks miss most of what decides performance on it. They rarely require grounding between symbolic identifiers (CVE, ATT&CK, malware families) and free text. Near-synonyms differ in ways that matter operationally. Retrieval must cross sources as different as vulnerability databases and social-media intelligence. Existing resources do not close the gap. MTEB-scale benchmarks contain limited cybersecurity content, while security benchmarks target single downstream tasks rather than embedding quality (Muennighoff et al., 2023; Thakur et al., 2021; Wu et al., 2025; Shi et al., 2026; Priyanshu et al., 2026; Zhang et al., 2025a).

We introduce SecEmbedEval, a benchmark for cybersecurity text embeddings, and evaluate twelve representative models on it. Our main contributions are as follows:

- SecEmbedEval, a benchmark dedicated to evaluating text embedding models in the cybersecurity domain, filling the gap left by general-purpose suites with little security content and by security benchmarks that do not evaluate embeddings.
- A cybersecurity dataset with over 17,000 task instances in Chinese and English, built through a hybrid pipeline that combines public-source data collection, source-aware preprocessing, large language model (LLM)-assisted candidate construction, and blind verification.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- A standardized evaluation framework that scores every model under one fixed protocol, so that results are reproducible and directly comparable. Two proposed metrics support aggregation: SEE-Score summarizes in-domain standing over six task measures, and TD measures the share of model pairs that public leaderboards and our benchmark rank differently.

- An empirical study of twelve representative embedding models. Results show security-domain pretrained encoders underperform general-purpose embedding models trained via contrastive learning. Clustering remains a bottleneck across all tested models. We also observe nearly no correlation between a model's deployment cost and its benchmark scores.

2 Related Work

General-purpose embedding benchmarks provide an important foundation for evaluating text representations, but they do not capture the requirements of cybersecurity text. MTEB (Muennighoff et al., 2023; Thakur et al., 2021) is the broadest, covering retrieval, clustering, and semantic textual similarity, and directly informs the protocol adopted here. MMTEB (Enevoldsen et al., 2025) scales it to over 500 tasks across 250+ languages, and language-specific suites such as TR-MTEB (Baysan & Gungor, 2025) and C-Pack (Xiao et al., 2024) replicate it for Turkish and Chinese respectively. No comparable replication exists for cybersecurity text. The adjacent studies of representation geometry, scaling behaviour, semantic alignment, and contrastive or instruction-tuned objectives (Wang et al., 2022; Jiang et al., 2024; Mahajan et al., 2024; Zhou et al., 2025; Brinner & Zarriess, 2025; Romero et al., 2025; Parupudi, 2026) inherit the same general-language scope, as do GLUE (Wang et al., 2018) and the STS Benchmark (Cer et al., 2017). None of these tests alias resolution, identifier grounding, cross-source threat correlation, or lexically similar but operationally different attack descriptions.

Cybersecurity resources demonstrate the importance of domain-specific evaluation, but they are not designed as unified benchmarks for text embedding quality. Task-specific datasets for command-line semantic similarity (Huang et al., 2024), web attack detection (Bakır, 2025), and multi-granularity traffic representations (Wu et al., 2026) each cover a limited slice of security text understanding. Broader benchmarks such as ExCyTIn-Bench (Wu et al., 2025), CyberMetric (Tihanyi et al., 2024), CyberSecEval (Wan et al., 2024), CTIBench (Alam et al., 2024), SEvenLLM (Ji et al., 2024), and related surveys (Ferrag et al., 2024) primarily target reasoning, generation, or knowledge recall, as do studies on CTF-style and CVE-oriented settings (Zhang et al., 2025a). None provides a unified protocol for comparing text embeddings across multiple cybersecurity tasks.

A separate line of work adapts language models to cybersecurity data. Social media has been used for real-time threat intelligence, technique knowledge graphs have been constructed from threat reports, and automatic CTI extraction from unstructured reports has been studied (Ech-Chammakhy et al., 2026; Li et al., 2026; Zhang et al., 2026). SecureBERT (Aghaei et al., 2023), CTI-BERT (Park & You, 2023), and CySecBERT (Bayer et al., 2024) were introduced to capture cybersecurity semantics better than general-purpose encoders. Domain-adapted resources such as CTIBench (Alam et al., 2024) likewise define task-level protocols rather than comparing the representations that underpin them, and Rajapaksha et al. (2024) showed the central role of embedding quality in retrieval-based attack investigation. These studies motivate domain-specific evaluation but leave open the need for a multi-task benchmark dedicated to cybersecurity text embeddings.

3 Dataset Construction

This section describes the SecEmbedEval construction pipeline. Figure 1 summarizes how the real-world sources flow into the five task categories, and Figure 2 details the shared preprocessing and per-task construction stages. The public sources fall into five families: CVE/CNNVD entries, vendor advisories, threat reports and blogs, security news and social media, and exploit repositories.

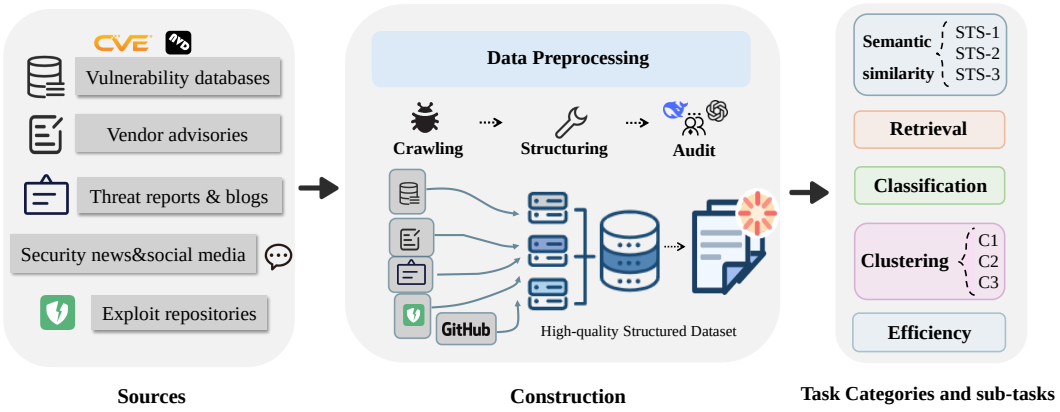

Figure 1: Overview of SecEmbedEval.

3.1 Data Sources and Composition

The dataset spans semantic similarity, retrieval, classification, clustering, and efficiency evaluation. Data are collected from public sources, with LLMs supplying supplementary support for sparse negatives, semantic edge cases, and class-imbalance issues.

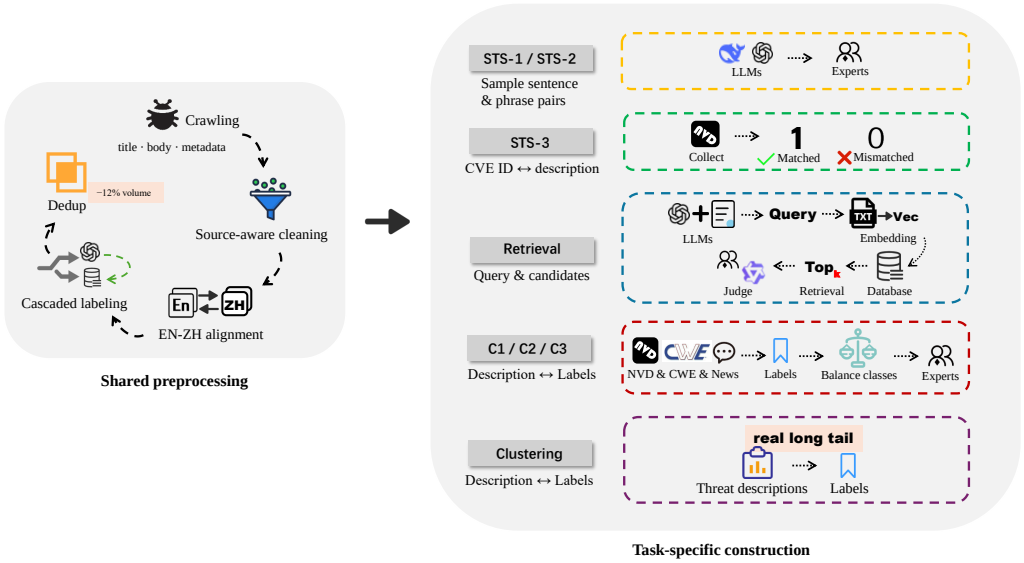

Figure 2: Construction of SecEmbedEval.

SecEmbedEval is bilingual, but the language mix follows from where each task draws its supervision. Tasks built from official structured feeds are entirely English: STS-3, C1, and C2 derive from National Vulnerability Database (NVD) descriptions and the MITRE CWE catalogue. Tasks built from the crawled reporting stream are predominantly Chinese. Table 11 reports the full breakdown.

The three cybersecurity-pretrained encoders we evaluate are English-only while the crawled reporting tasks are predominantly Chinese, so language and domain specialization are entangled on those tasks. We therefore build English parallel subsets of STS-1 and STS-2. This turns language into a controlled factor on the two subtasks where it matters most.

3.2 Preprocessing and Task Construction

Raw crawled data pass through the shared preprocessing shown in Figure 2 before task construction. Source-specific cleaners remove boilerplate while extracting titles, body text, and metadata, and both the English original and the Chinese repost of the same event are preserved. Source-level topic labels come from a cascade in which a lightweight classifier labels the full corpus and only low-confidence cases are escalated to a stronger model. Two-stage deduplication removes exact duplicates by SimHash and near-duplicates by cosine similarity, retaining the longer version, for about 12% of the volume. Sensitive surface forms (emails, IPs, phone numbers, identifiers, and domains) are anonymized while the technical content is preserved.

The essentials of task construction are as follows.

Semantic similarity. We assess model capability with three granularity-varying subtasks: sentence-level semantic equivalence (STS-1), term-wise relatedness (STS-2), and CVE identifier grounding (STS-3). STS-1 uses 589 domain-stratified anchors across twelve security domains, pairing each with one positive and one generated negative to preserve matched topic distributions between classes. Using 138 seed terms, STS-2 creates minimal-contrast term pairs whose negatives derive from each anchor's own term family. For STS-3, 4,000 CVE identifiers are matched against NVD descriptions. Negatives are filtered to discard trivial string-search hits and false negatives from sequential CVE numbering. All subtasks use binary labels and are evaluated with threshold-free ranking metrics. The measurements behind that choice are in Appendix A.1, and the English parallel subsets (STS-1-en, STS-2-en) built to control language are in Appendix A.6.

Retrieval. A corpus of about 6,000 documents across ten source categories is paired with 800 queries. Relevance labels come from TREC-style pooling over the twelve evaluated systems at depth 20, judged blind to provenance on a three-grade scale.

Classification. Three subtasks take their labels from structured resources. C1 maps vulnerability descriptions to 37 CWE classes at a fixed 40 real plus 10 synthetic instances per class, a uniform augmentation ratio that decorrelates provenance from the label. C2 maps CWE entries onto the ten top-level Pillars of the official View-1000 hierarchy. C3 assigns six threat categories, balanced simultaneously across category and disclosure year so that category is not confounded with reporting era.

Clustering. The threat-description space behind C3 supplies 3,600 instances over the same six categories. Their natural label skew and their genuine multi-label structure are retained, with 41.9% of instances carrying more than one category, so the task is scored with extended BCubed precision and recall (§A.9.3).

Table 17 in Appendix B summarizes the resulting task-level statistics.

4 Evaluation Task Framework and Metrics

SecEmbedEval evaluates each model on five task families under one protocol. Appendix A.9 specifies each in full.

The three similarity subtasks carry binary labels and are scored by AUC and average precision. Retrieval is scored against pooled judgments with Recall@K, NDCG@K, and MRR. For classification, a lightweight L_2-regularized logistic-regression probe is fitted to the frozen embeddings of each subtask and the reported score is macro-F1, which weights the long-tailed classes equally. Clustering is scored by extended BCubed precision and recall against the full multi-label ground truth. A bare BCubed F1 is not interpretable on its own, so every score is read between two reference points, a trivial-solution ceiling of 0.524 from placing every instance in one cluster and a random-embedding floor of 0.228. Efficiency reports throughput, latency, and peak memory under a standardized setting after warm-up.

Beyond the per-family metrics, two aggregate measures summarize overall standing. SEE-Score is the mean of six task metrics (classification macro-F1, KMeans BCubed F1, the AUCs of the three similarity subtasks, and NDCG@10), each normalized across all twelve evaluated models before

averaging, since the metrics do not share a scale:

$$\text{SEE} = \frac{1}{6} \sum_{m \in \{\text{CLS, CLU, STS}_{1,2,3}\text{, RET}\}} \tilde{v}_m. \tag{1}$$

Five of the components are min–max normalized over the twelve models. STS-3 is placed on the fixed $[0.5, 1]$ AUC interval running from chance to perfect instead, because min–max over its 0.5045–0.5487 chance band would stretch that band across the full range and hand it full metric weight. TD quantifies the share of model pairs whose ranking order differs between a public leaderboard and this benchmark. It is defined on two rankings r_1, r_2 of the same n models:

$$\text{TD}(r_1, r_2) = \frac{D}{\binom{n}{2}}, \tag{2}$$

where D is the number of discordant pairs, that is, pairs of models that the two rankings order differently. TD is 0 when the two rankings agree on every pair and grows toward 1 as they disagree. Per-model SEE-Score values and the six component metrics are reported for all twelve models in Table 19 (Appendix D.1).

5 Experimental Results

5.1 Experimental Setup

Similarity and retrieval are scored by cosine similarity. Classification attaches a logistic-regression head to frozen embeddings over five random seeds, clustering is scored with extended BCubed against the multi-label ground truth, and efficiency is measured after warm-up. The full protocol, covering hardware, checkpoint-native pooling, instruction prefixes and hyperparameter grids are consolidated in Appendix A.10.

We evaluated twelve models covering complementary model families. The selection rationale is given in Appendix D.

5.2 Do Public Benchmarks Predict Cybersecurity Performance?

Seven of the twelve evaluated models appear on the MTEB(Multilingual, v2) leaderboard. Their public scores, aggregated from the official release under the leaderboard's own Mean rule, are plotted against SEE-Score in Figure 3.

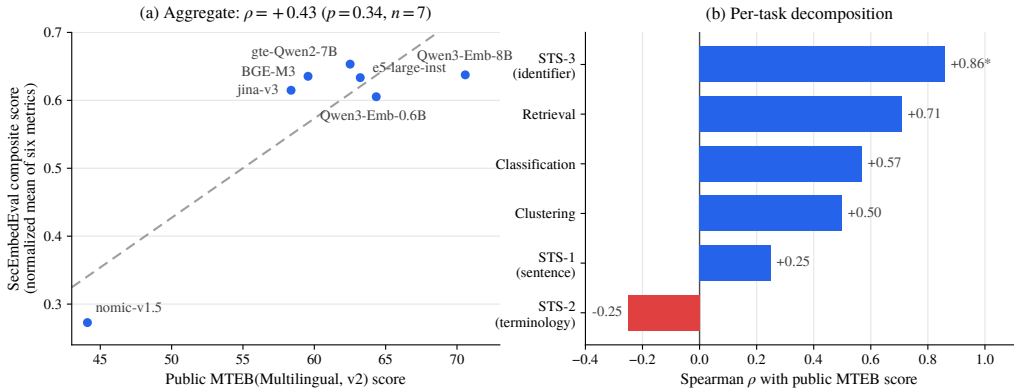

Figure 3: Public leaderboard score against SecEmbedEval performance, computed separately for each task family.

At the aggregate level the two measures barely agree: Spearman $\rho = +0.43$ ($p = 0.34$), and TD $= 7/21 = 0.33$, seven of the twenty-one model pairs that the public leaderboard settles coming out the other way under SEE-Score. Sensitivity checks at $n = 7$ move the coefficient between -0.64 and

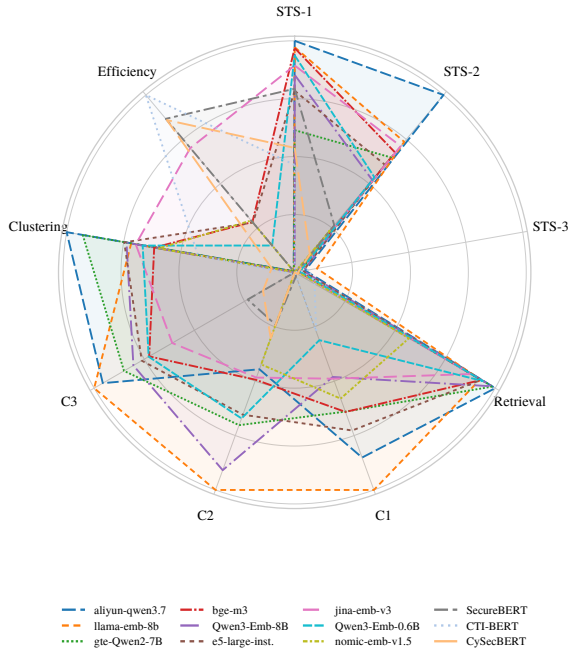

Figure 4: Per-subtask profiles of all twelve models on the nine subtasks.

$+0.64$ without any specification reaching significance, so a public score is not a usable predictor of in-domain standing.

Per-task decomposition in Figure 3 pinpoints where cross-domain transfer breaks down. For all task families covered by the general-domain benchmark, correlations against public MTEB scores remain positive yet statistically insignificant. Retrieval yields $\rho = +0.71$ ($p = 0.07$), classification $+0.57$, and clustering $+0.50$, with no task family reaching $p < 0.05$ at $n = 7$. By contrast, neither similarity subtask shows meaningful correlation. This pattern holds even for English-only subsets, implying the disconnect stems from task properties instead of language differences. TD metrics reproduce this divergence. Pair-wise discordance grows monotonically, from $4/21$ for retrieval to $15/21$ for terminology-level similarity, as decomposed by task family in Table 20.

5.3 Results by Task Family

Figure 4 shows the per-subtask profile of all twelve models across the nine subtasks. Eight of the axes are normalized over the twelve models. STS-3 keeps a fixed 0.5–1.0 AUC scale from chance to perfect instead, because normalizing its chance band would stretch that band over the full radius. The outlines therefore compare standing within a subtask rather than raw metric magnitude, and the raw ranges are those of the result tables.

Sentence-level equivalence is close to solved at the top, where the leader reaches 0.9852 AUC, as shown in Table 1. Term-level relatedness spreads the field far wider, and only the commercial system clears 0.65. Ranking well on full sentences says little about ranking terms, and the two orderings disagree. CVE identifier grounding is the flattest result in the benchmark. All twelve models sit inside a narrow chance band, so no evaluated system, commercial or open, connects a bare identifier to its advisory.

Retrieval separates the models more sharply than any other task, as Table 2 shows. The three leading systems form one statistically inseparable tier, and the interpretable structure is the boundary below it. nomic-embed-text-v1.5 drops to roughly 0.42 NDCG@10, and the three cybersecurity-pretrained encoders retrieve almost nothing, in part because their English-only vocabularies.

llama-embed-nemotron-8b leads the classification average, with the commercial system and gte-Qwen2-7B-instruct close behind, as reported in Table 3. The three subtasks differ in what they

Table 1: Semantic similarity results over all 12 models, ordered by Task 1 AUC. Brackets give bootstrap 95% confidence intervals over 1,000 resamples.

Model	Task 1 (1,178 pairs) AUC	Task 1 AP	Task 2 (1,281 pairs) AUC	Task 2 AP	Task 2-zh AUC	Task 3 AUC
aliyun_qwen3.7-text-embedding	**0.9852** [0.979, 0.990]	**0.9863**	**0.7697** [0.744, 0.793]	**0.7957**	**0.8018**	0.5230
llama-embed-nemotron-8b	0.9731 [0.965, 0.980]	0.9742	0.6487 [0.616, 0.679]	0.6947	0.7854	**0.5487**
bge-m3	0.9715 [0.963, 0.979]	0.9752	0.6209 [0.590, 0.654]	0.6719	0.7309	0.5085
Qwen3-embedding-0.6B	0.9576 [0.947, 0.967]	0.9621	0.5571 [0.525, 0.587]	0.5721	0.6589	0.5109
jina-embeddings-v3	0.9385 [0.926, 0.951]	0.9457	0.6377 [0.606, 0.668]	0.6766	0.7288	0.5085
Qwen3-embedding-8B	0.9207 [0.905, 0.935]	0.9331	0.5502 [0.516, 0.581]	0.5776	0.6810	0.5186
SecureBERT	0.8945 [0.876, 0.913]	0.8741	0.4333 [0.402, 0.464]	0.4901	0.6160	0.5045
multilingual-e5-large-instruct	0.8907 [0.873, 0.907]	0.8936	0.5894 [0.557, 0.622]	0.6227	0.7224	0.5183
gte-Qwen2-7B-instruct	0.8161 [0.791, 0.840]	0.8286	0.6083 [0.578, 0.639]	0.6593	0.7430	0.5117
CySecBERT	0.7825 [0.754, 0.806]	0.7990	0.3581 [0.328, 0.387]	0.4448	0.5020	0.5084
CTI-BERT	0.7503 [0.722, 0.776]	0.7841	0.3114 [0.283, 0.338]	0.4646	0.4341	**0.5315**
nomic-embed-text-v1.5	0.5469 [0.514, 0.581]	0.5678	0.4219 [0.391, 0.453]	0.4467	0.5468	0.5062

Table 2: Retrieval results over all 12 models, ordered by NDCG@10. All models are scored against the same pooled relevance judgments.

Model	Recall@10	Recall@100	NDCG@10	NDCG@100	Hit@10	MRR
aliyun_qwen3.7-text-embedding	**0.5311**	0.8461	**0.7439**	**0.7816**	**0.9812**	**0.9292**
gte-Qwen2-7B-instruct	0.5228	0.8417	0.7338	0.7749	0.9675	0.9133
Qwen3-embedding-8B	0.5219	**0.8534**	0.7321	0.7770	0.9688	0.9166
Qwen3-embedding-0.6B	0.4957	0.7978	0.6983	0.7361	0.9575	0.8976
bge-m3	0.4820	0.7644	0.6920	0.7165	0.9475	0.8922
multilingual-e5-large-instruct	0.4854	0.7609	0.6914	0.7162	0.9537	0.8932
llama-embed-nemotron-8b	0.4917	0.7800	0.6881	0.7192	0.9513	0.8956
jina-embeddings-v3	0.4633	0.7518	0.6535	0.6880	0.9487	0.8639
nomic-embed-text-v1.5	0.3221	0.4832	0.4249	0.4486	0.7950	0.6458
CTI-BERT	0.0665	0.1955	0.0762	0.1157	0.3125	0.1831
CySecBERT	0.0091	0.0435	0.0141	0.0228	0.1237	0.0532
SecureBERT	0.0019	0.0310	0.0028	0.0134	0.0387	0.0208

Table 3: Classification results over all 12 models, ordered by the three-task average. Values are mean test-set macro-F1 $\pm$ std over 5 random seeds (1, 2, 3, 4, 42).

Model	Task 1 (F1)	Task 2 (F1)	Task 3 (F1)
llama-embed-nemotron-8b	**0.8243** $\pm$ 0.0167	**0.6463** $\pm$ 0.0560	**0.7175** $\pm$ 0.0255
aliyun_qwen3.7-text-embedding	0.7927 $\pm$ 0.0149	0.5621 $\pm$ 0.1024	0.7056 $\pm$ 0.0284
gte-Qwen2-7B-instruct	0.7479 $\pm$ 0.0118	0.6011 $\pm$ 0.0650	0.6764 $\pm$ 0.0328
multilingual-e5-large-instruct	0.7663 $\pm$ 0.0149	0.5938 $\pm$ 0.0412	0.6520 $\pm$ 0.0271
Qwen3-embedding-8B	0.7141 $\pm$ 0.0139	0.6325 $\pm$ 0.0458	0.6635 $\pm$ 0.0253
bge-m3	0.7482 $\pm$ 0.0142	0.5694 $\pm$ 0.0883	0.6418 $\pm$ 0.0254
Qwen3-embedding-0.6B	0.6782 $\pm$ 0.0167	0.5964 $\pm$ 0.0350	0.6425 $\pm$ 0.0268
jina-embeddings-v3	0.7158 $\pm$ 0.0188	0.5679 $\pm$ 0.0386	0.6100 $\pm$ 0.0193
nomic-embed-text-v1.5	0.7352 $\pm$ 0.0228	0.5587 $\pm$ 0.0437	0.4415 $\pm$ 0.0281
SecureBERT	0.6157 $\pm$ 0.0129	0.5310 $\pm$ 0.0641	0.5062 $\pm$ 0.0349
CySecBERT	0.6121 $\pm$ 0.0081	0.5404 $\pm$ 0.0357	0.4851 $\pm$ 0.0202
CTI-BERT	0.6697 $\pm$ 0.0099	0.4945 $\pm$ 0.0738	0.4450 $\pm$ 0.0168

demand. Mapping descriptions to CWE classes is the most tractable despite the largest label space. Fine-to-coarse Pillar mapping turns out hardest and least stable across seeds, while threat-category assignment spreads the models most widely. The per-seed values behind each average are in Appendix A.2. A TF-IDF baseline under the same protocol would rank fourth of thirteen on the first subtask and last on the third, so learned representations matter on two subtasks and merely help on the third (Appendix A.3).

Clustering is the hardest category in the benchmark. Every score is read against a random-embedding floor and a trivial one-cluster ceiling, because a collapsed solution otherwise outscores every genuine clustering. Four of the six algorithm configurations collapse in exactly that way and are excluded by the degeneracy criteria of Appendix A.9.3, audited per configuration in Appendix Table 25. Table 4 reports the two that survive, KMeans and Euclidean agglomerative clustering at $k = 6$, with aliyun-qwen3.7 abbreviating the commercial model. On those two configurations even the best model stays

Table 4: Clustering results over all 12 models, ordered by mean KMeans extended BCubed F1. KMeans values are mean ± std over ten random states (n_init= 10).

Model	KMeans ($k = 6$) BC-F1	BC-P	BC-R	ARI	NMI	Agglomerative ($k = 6$) BC-F1	ARI	NMI
aliyun-qwen3.7	**0.365±0.011**	**0.449±0.021**	0.308±0.014	**0.172±0.050**	**0.246±0.047**	**0.435**	0.064	**0.200**
gte-Qwen2-7B	0.358±0.001	0.417±0.001	**0.313±0.002**	0.147±0.031	0.210±0.028	0.367	0.103	0.164
e5-large-inst.	0.340±0.008	0.407±0.002	0.292±0.011	0.059±0.008	0.138±0.003	0.364	0.046	0.136
Qwen3-Emb-8B	0.339±0.012	0.428±0.009	0.282±0.018	0.154±0.021	0.227±0.020	0.424	0.045	0.176
llama-emb-8b	0.337±0.001	0.419±0.001	0.281±0.002	0.057±0.008	0.156±0.008	0.425	0.047	0.178
jina-emb-v3	0.335±0.000	0.404±0.000	0.286±0.001	0.055±0.001	0.127±0.000	0.338	0.038	0.119
Qwen3-Emb-0.6B	0.332±0.009	0.408±0.001	0.280±0.013	0.110±0.012	0.165±0.009	0.360	**0.113**	0.180
bge-m3	0.327±0.007	0.403±0.006	0.275±0.008	0.076±0.015	0.134±0.010	0.390	0.052	0.131
nomic-emb-v1.5	0.325±0.015	0.384±0.004	0.283±0.020	0.072±0.012	0.141±0.004	0.369	0.044	0.133
CTI-BERT	0.308±0.001	0.397±0.000	0.252±0.001	0.050±0.001	0.124±0.001	0.314	0.111	0.166
CySecBERT	0.276±0.011	0.372±0.000	0.219±0.015	0.017±0.003	0.048±0.004	0.299	0.000	0.035
SecureBERT	0.266±0.000	0.364±0.000	0.210±0.000	0.009±0.000	0.036±0.000	0.258	0.007	0.028

Reference points: random 32-d Gaussian embeddings, KMeans $k = 6$: BC-F1 $= 0.228$. Single-cluster trivial solution: BC-F1 $= 0.524$ (BC-P $= 0.355$, BC-R $= 1.000$), reported as a ceiling on degenerate output, not a competitor.

Table 5: Efficiency results for all 12 models, ordered by throughput at batch size 25. Peak GPU memory is measured at the same batch size. Best value per column in bold, second-best underlined.

Model	Batch 1 Thrpt. (samples/s)	Batch 1 Lat. (ms/sample)	Batch 8 Thrpt. (samples/s)	Batch 8 Lat. (ms/sample)	Batch 25 Thrpt. (samples/s)	Batch 25 Lat. (ms/sample)	Peak GPU (MB)
CTI-BERT	**210.72**	**4.75**	**754.94**	**1.32**	**977.43**	**1.02**	555.81
SecureBERT	206.27	4.85	684.69	1.46	847.91	1.18	574.62
CySecBERT	210.03	4.76	681.83	1.47	840.96	1.19	**521.75**
jina-emb-v3	34.05	29.37	289.89	3.45	689.09	1.45	1147.39
nomic-emb-v1.5	54.53	18.34	233.55	4.28	294.55	3.39	699.83
bge-m3	105.76	9.46	263.78	3.79	286.31	3.49	2295.80
e5-large-inst.	103.25	9.69	264.66	3.78	283.20	3.53	2265.68
Qwen3-Emb-0.6B	46.54	21.49	146.80	6.81	158.39	6.31	2971.25
aliyun-qwen3.7*	1.44	695.20	8.67	115.30	18.81	53.16	—
gte-Qwen2-7B	13.41	74.57	18.86	53.02	18.13	55.17	31645.00
llama-emb-8b	12.28	81.44	17.34	57.67	16.65	60.07	29200.92
Qwen3-Emb-8B	12.38	80.79	12.70	78.77	12.32	81.16	30080.86

* aliyun_qwen3.7-text-embedding is served via a cloud API. Its figures reflect end-to-end network round-trips rather than local compute, and the provider caps a single request at 20 items, so batch 25 is issued as two calls. GPU usage is not applicable.

near the middle of the range, and a TF-IDF control that scores higher does so only by collapsing the corpus into one dominant cluster. Seed-to-seed stability of the KMeans column is analyzed in Appendix Table 15.

Throughput, latency, and peak memory span two orders of magnitude between the 110M security encoders and the 7B–8B models, as shown in Table 5. Cost barely tracks score. CTI-BERT is far faster than the large models while retrieving almost nothing, and mid-sized systems deliver nearly the retrieval score of the 8B tier at a fraction of the memory and throughput cost.

Across families, the training objective dominates everything else. The three masked-language-model security encoders hold the last three places on the aggregate and on retrieval, classification, and clustering alike, despite being pretrained on this literature. The deficit survives on the English parallel subsets and on the entirely English classification subtasks, so it cannot be a language effect. It follows the missing sentence-level contrastive objective rather than the corpus. Within the leading tier, architecture still predicts the profile. gte-Qwen2-7B-instruct ranks near the top in retrieval and clustering yet ninth of twelve on sentence similarity, while llama-embed-nemotron-8b leads classification and trails in retrieval. Organizing a corpus and telling two descriptions apart are separable capabilities, and no evaluated model leads both.

The rankings themselves hold up well. Chinese and English rankings on the parallel similarity subsets correlate strongly, so the conclusions attach to models rather than to one language's data (Appendix A.6). As Table 18 reports, identifier grounding exhibits the weakest pairwise coupling among metrics. Scale draws its own boundary. The overall leader, at SEE-Score 0.804, still sits in the same chance band as every other model on identifier grounding, because tokenizers shatter an identifier's serial digits into unstable fragments and leave no compositional route from identifier to description.

Clustering defeats every model, with recall bounded by genuine multi-label overlap rather than model capacity, and internal cluster-quality metrics invert the label-based ranking outright, SecureBERT being the best model by every internal criterion and the worst by every external one. Deployment cost is almost unrelated to score across the viable band, so model selection is largely an efficiency decision.

6 Conclusion and Future Work

SecEmbedEval measures how text embeddings behave on the work security analysts actually do. Every number comes from one released pipeline over released data, bilingual throughout.

The evaluation returns a clear verdict on model selection. Public standing does not transfer. Training objective matters more than domain exposure. The three masked-language-model security encoders, pretrained on this very literature, hold the last three places on the aggregate, while contrastive general-purpose and instruction-tuned models lead throughout. Rankings are stable across languages and largely consistent across tasks, with identifier grounding the one capability that stands apart. Future work follows from the design. Widening the pool to Chinese-specialized embedding families would test whether the current leaders hold. The failure modes also point at what representations need: identifier-aware tokenization, structured knowledge such as CWE and ATT&CK graphs, and objectives that reward fine semantic distinctions rather than topical organization alone.

Ethics Statement

SecEmbedEval is built exclusively from public sources: the National Vulnerability Database, vendor advisories, public threat reports, security blogs, and public exploit repositories. Its preprocessing pipeline removes emails, IP addresses, phone numbers, account identifiers, and domains, keeping all core technical content intact. Since security-related text can be dual-use, the released generation and review prompts enforce analyst-level descriptions for all cases. Exploitation steps, payloads, and reusable offensive instructions are excluded entirely. Flagged candidates are either discarded or rewritten, as the released templates in Table22 specify. No human-subjects data or personally identifying information is collected.

AI Use Statement

Generative AI was used as a disclosed component of the method and in two supporting roles. In dataset construction, LLMs served as candidate generators. They were never autonomous annotators. Every generation stage was bound by the released prompt templates in Table 22 and gated by a verification pass blind to the generation target. AI tools also assisted with language editing of prose the authors had drafted and with parts of the evaluation and analysis code, all of which was read and verified by the authors before use. We did not use generative AI to produce research ideas, experimental design, or any reported number. Every number comes from running the released code on the released data. We take full responsibility for the content of this work.

References

E. Aghaei, X. Niu, W. Shadid, and E. Al-Shaer. SecureBERT: A domain-specific language model for cybersecurity. In *Security and Privacy in Communication Networks*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 39–56. Springer Nature Switzerland, 2023. doi: 10.1007/978-3-031-25538-0_3. URL https://doi.org/10.1007/978-3-031-25538-0_3.

M. T. Alam, D. Bhusal, L. Nguyen, and N. Rastogi. CTIBench: A benchmark for evaluating LLMs in cyber threat intelligence. *arXiv preprint arXiv:2406.07599*, 2024. URL https://arxiv.org/abs/2406.07599.

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486, 2009.

R. Bakır. UniEmbed: A novel approach to detect XSS and SQL injection attacks leveraging multiple feature fusion with machine learning techniques. *Arabian Journal for Science and Engineering*, 50(19):15591–15604, 2025. doi: 10.1007/s13369-024-09916-4. URL https://doi.org/10.1007/s13369-024-09916-4.

M. Bayer, P. Kuehn, R. Shanehsaz, and C. Reuter. CySecBERT: A domain-adapted language model for the cybersecurity domain. *ACM Transactions on Privacy and Security*, 27(2):1–20, 2024. doi: 10.1145/3652594. URL https://doi.org/10.1145/3652594.

M. S. Baysan and T. Gungor. TR-MTEB: A comprehensive benchmark and embedding model suite for turkish sentence representations. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 8867–8887, 2025. doi: 10.18653/v1/2025.findings-emnlp.471. URL https://aclanthology.org/2025.findings-emnlp.471/.

M. Brinner and S. Zarriess. SemCSE: Semantic contrastive sentence embeddings using LLM-generated summaries for scientific abstracts. *arXiv preprint arXiv:2507.13105*, 2025. URL https://arxiv.org/abs/2507.13105.

D. Cer, M. Diab, E. Agirre, et al. SemEval-2017 Task 1: Semantic textual similarity—multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 1–14, 2017. doi: 10.18653/v1/S17-2001. URL https://aclanthology.org/S17-2001/.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Y. Ech-Chammakhy, O. Azrara, J. Chbili, and A. Motii. STINER: Automated extraction of strategic cyber threat intelligence from x. *arXiv preprint arXiv:2608.14418*, 2026. URL https://arxiv.org/abs/2608.14418.

K. Enevoldsen, I. Chung, I. Kerboua, M. Kardos, A. Mathur, D. Stap, et al. MMTEB: Massive multilingual text embedding benchmark. *arXiv preprint arXiv:2502.13595*, 2025. URL https://arxiv.org/abs/2502.13595.

M. A. Ferrag, N. Tihanyi, M. A. Yazici, M. Debbah, et al. Large language models for cyber security: A systematic literature review. *arXiv preprint arXiv:2405.04760*, 2024. doi: 10.48550/arXiv.2405.04760. URL https://arxiv.org/abs/2405.04760.

M. Günther, S. Sturua, M. K. Akram, I. Mohr, A. Ungureanu, B. Wang, S. Eslami, S. Martens, M. Werk, N. Wang, and H. Xiao. jina-embeddings-v4: Universal embeddings for multimodal multilingual retrieval. *arXiv preprint arXiv:2506.18902*, 2025. URL https://arxiv.org/abs/2506.18902.

Y. Guo, Q. Hu, Q. Tang, and Y. Le Traon. An empirical study of the imbalance issue in software vulnerability detection. In *Computer Security – ESORICS 2023*, volume 14347 of *LNCS*, pp. 371–390. Springer, 2023. doi: 10.1007/978-3-031-51482-1_19. URL https://doi.org/10.1007/978-3-031-51482-1_19.

S.-Y. Huang, C.-L. Yang, C.-Y. Lin, and C.-Y. Huang. CmdCaliper: A semantic-aware command-line embedding model and dataset for security research. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 20188–20206, 2024. doi: 10.18653/v1/2024.emnlp-main.1126. URL https://aclanthology.org/2024.emnlp-main.1126/.

H. Ji, J. Yang, L. Chai, C. Wei, L. Yang, Y. Duan, et al. SEvenLLM: Benchmarking, eliciting, and enhancing abilities of large language models in cyber threat intelligence. *arXiv preprint arXiv:2405.03446*, 2024. URL https://arxiv.org/abs/2405.03446.

T. Jiang, S. Huang, Z. Luan, D. Wang, et al. Scaling sentence embeddings with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3182–3196, 2024. doi: 10.18653/v1/2024.findings-emnlp.181. URL https://aclanthology.org/2024.findings-emnlp.181/.

V. Karpukhin, B. Oguz, S. Min, P. Lewis, et al. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6769–6781, 2020. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

E. Kolodin, E. Krasnoperov, E. Kosarev, and F. Minkin. Giga-Embeddings: Mixture-of-experts encoders for high-throughput text embeddings. *arXiv preprint arXiv:2608.23806*, 2026. URL https://arxiv.org/abs/2608.23806.

C. Li, Y. Cheng, T. C. Finnisa, Q. Cui, W. Ding, and P. Gao. BEACON: Behavior-anchored cross-source knowledge graph construction for cyber threat intelligence. *arXiv preprint arXiv:2608.28394*, 2026. URL https://arxiv.org/abs/2608.28394.

Y. Mahajan, N. Bansal, E. Blanco, and S. Karmaker. ALIGN-SIM: A task-free test bed for evaluating and interpreting sentence embeddings through semantic similarity alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7393–7428, 2024. doi: 10.18653/v1/2024.findings-emnlp.436. URL https://aclanthology.org/2024.findings-emnlp.436/.

N. Muennighoff, N. Tazi, L. Magne, and N. Reimers. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2014–2037, 2023. doi: 10.18653/v1/2023.eacl-main.148. URL https://aclanthology.org/2023.eacl-main.148/.

Y. Park and W. You. A pretrained language model for cyber threat intelligence. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 113–122, 2023. doi: 10.18653/v1/2023.emnlp-industry.12. URL https://aclanthology.org/2023.emnlp-industry.12/.

V. S. R. Parupudi. Anisotropy decides cosine vs. rank metrics for text embeddings. *arXiv preprint arXiv:2606.29571*, 2026. URL https://arxiv.org/abs/2606.29571.

A. Priyanshu, S. Vijay, K. Majd, X. He, F. Burch, T. Matsumoto, et al. Vulnerability Localization Benchmark: Measuring agentic security analysis at repository scale. *arXiv preprint arXiv:2609.15939*, 2026. URL https://arxiv.org/abs/2609.15939.

R. Rajapaksha, S. Rani, and E. Karafili. A RAG-based question-answering solution for cyber-attack investigation and attribution. In *Computer Security – ESORICS 2024 International Workshops*, volume 15264 of *LNCS*, pp. 238–256. Springer, 2024. doi: 10.1007/978-3-031-82362-6_15. URL https://doi.org/10.1007/978-3-031-82362-6_15.

N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

M. Romero, S. Ding, C. D. Barret, G. Dinu, and G. Karypis. Beyond instruction-conditioning, MoTE: Mixture of task experts for multi-task embedding models. *arXiv preprint arXiv:2506.17781*, 2025. URL https://arxiv.org/abs/2506.17781.

J. Shi, E. Tsien, Y. Di, H. Zhang, Y. Tang, R. Dey, et al. VEX-Bench: Benchmarking LLM agents for assessing exploitability of software supply chain vulnerabilities. *arXiv preprint arXiv:2609.08040*, 2026. URL https://arxiv.org/abs/2609.08040.

N. Thakur, N. Reimers, A. Rücklé, et al. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. URL https://openreview.net/forum?id=wCu6T5xFjeJ.

N. A. Tihanyi, M. A. Ferrag, T. Bisztray, et al. CyberMetric: A benchmark dataset for evaluating large language models knowledge in cybersecurity. In *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*, pp. 296–302. IEEE, 2024. doi: 10.1109/CSR61664.2024.10679494. URL https://doi.org/10.1109/CSR61664.2024.10679494.

S. Wan, C. Nikolaidis, D. Menasce, et al. CYBERSECEVAL 3: Advancing the evaluation of cybersecurity risks and capabilities in large language models. *arXiv preprint arXiv:2408.01605*, 2024. doi: 10.48550/arXiv.2408.01605. URL https://arxiv.org/abs/2408.01605.

A. Wang, A. Singh, J. Michael, et al. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 353–355, 2018. doi: 10.18653/v1/W18-5446. URL https://aclanthology.org/W18-5446/.

L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, and F. Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022. URL https://arxiv.org/abs/2212.03533.

B. Wu, D. M. Divakaran, and M. Gurusamy. UniNet: A unified multi-granular traffic modeling framework for network security. *IEEE Transactions on Cognitive Communications and Networking*, 12:2424–2438, 2026. doi: 10.1109/TCCN.2025.3585170. URL https://doi.org/10.1109/TCCN.2025.3585170.

Y. Wu, M. Velazco, A. Zhao, M. R. Meléndez Luján, S. Movva, Y. K. Roy, Q. Nguyen, R. Rodriguez, Q. Wu, M. Albada, J. Kiseleva, and A. Mudgerikar. ExCyTIn-Bench: Evaluating LLM agents on cyber threat investigation. *arXiv preprint arXiv:2507.14201*, 2025. URL https://arxiv.org/abs/2507.14201.

S. Xiao, Z. Liu, P. Zhang, et al. C-Pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 641–649. Association for Computing Machinery (ACM), 2024. doi: 10.1145/3626772.3657878. URL https://doi.org/10.1145/3626772.3657878.

L. Xiong, C. Xiong, Y. Li, K.-F. Tang, et al. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=zeFrfgyZln.

A. K. Zhang, N. Perry, J. Li, et al. Cybench: A framework for evaluating cybersecurity capabilities and risks of language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, pp. 25094–25243, 2025a. URL https://openreview.net/forum?id=tc90LV0yRL.

J. Zhang, J. Xu, Z. Xie, and Y. Fan. Retrieval-constrained policy optimization for attack technique extraction from cyber threat intelligence. *arXiv preprint arXiv:2608.06778*, 2026. URL https://arxiv.org/abs/2608.06778.

Y. Zhang, M. Li, D. Long, X. Zhang, H. Lin, B. Yang, P. Xie, A. Yang, D. Liu, J. Lin, F. Huang, and J. Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025b. URL https://arxiv.org/abs/2506.05176.

Z. Zhou, M. Huang, and Q. Miao. Rank-awareness and angular constraints: A new perspective on learning sentence embeddings from NLI data. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 22195–22209, 2025. doi: 10.18653/v1/2025.emnlp-main.1129. URL https://aclanthology.org/2025.emnlp-main.1129/.

A Reproducibility and Quality Assurance

A.1 Inter-Annotator Agreement and Data Quality

Two independent annotators evaluated stratified samples from four tasks under written guidelines. Table 6 summarizes the results.

Table 6: Inter-annotator agreement and human ceiling assessments, measured on pilot samples during protocol development.

Task	Samples	Method	IAA/Ceiling
Phrase-pair relation	300	Dual annotators	$\kappa = 0.851$
Sentence-pair relation	302	Dual annotators	$\kappa = 0.939$ (binary)
CWE class assignment	300	Dual annotators	Acc. = 85.3% (37-way)

Verification differs by task, so we state it per task rather than in aggregate. C2 requires no annotation. Its labels are read directly off the CWE View-1000 hierarchy by tracing ChildOf edges to a Pillar, so they are as correct as the MITRE catalogue. STS-3, C1 and C3 draw their labels from NVD and vendor advisory fields rather than from human judgment. For C1 we additionally report the rate at which an independent LLM pass agrees with the NVD Primary CWE assignment (64.1% on real records, 76.5% on generated ones, Table 21), which bounds how much of the residual disagreement is labelling noise rather than model error. STS-1 and STS-2 are generated to a specification and verified by a second pass that does not see the generation target, agreeing on 90.7% of STS-2 candidates and 89.2% of STS-2-en candidates. Retrieval relevance comes from LLM adjudication over a model-pooled candidate set (§A.9).

A.2 Classification Stability Across Random Seeds

We re-evaluated all 12 models under five random seeds (1, 2, 3, 4, 42) on the same cached embeddings, to measure stability across splits and classifier initializations. Results for each subtask and the aggregate macro-F1 are in Tables 7, 8, and 9, and Figure 5 plots the aggregate with its seed-to-seed dispersion. Points are each model's three-task mean macro-F1 averaged over five seeds, with error bars for the standard deviation over seeds and a shaded band at one standard deviation around the leader.

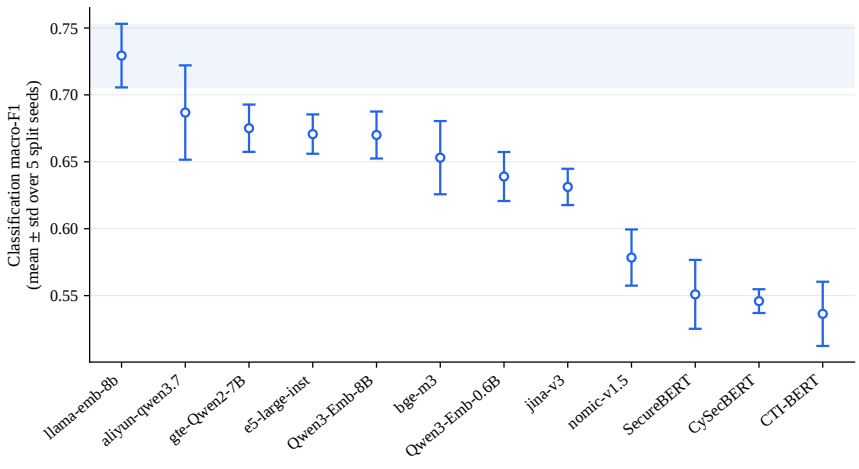

Figure 5: Classification stability across split seeds.

Table 7: Classification Task 1 : per-seed and aggregate results.

Model	s=1	s=2	s=3	s=4	s=42	mean±std	rank
llama-embed-nemotron-8b	0.7993	0.8197	0.8233	0.8410	0.8382	0.8243±0.0167	1
aliyun_qwen3.7-text-embedding	0.7831	0.7871	0.7766	0.8105	0.8065	0.7927±0.0149	2
multilingual-e5-large-instruct	0.7618	0.7539	0.7578	0.7666	0.7916	0.7663±0.0149	3
bge-m3	0.7461	0.7458	0.7320	0.7459	0.7711	0.7482±0.0142	4
gte-Qwen2-7B-instruct	0.7389	0.7466	0.7679	0.7392	0.7468	0.7479±0.0118	5
nomic-embed-text-v1.5	0.7267	0.7526	0.7249	0.7076	0.7641	0.7352±0.0228	6
jina-embeddings-v3	0.7075	0.7143	0.6935	0.7186	0.7449	0.7158±0.0188	7
Qwen3-embedding-8B	0.7349	0.7006	0.7184	0.7023	0.7143	0.7141±0.0139	8
Qwen3-embedding-0.6B	0.6569	0.6980	0.6854	0.6651	0.6854	0.6782±0.0167	9
CTI-BERT	0.6573	0.6728	0.6839	0.6646	0.6698	0.6697±0.0099	10
SecureBERT	0.6324	0.6049	0.6078	0.6270	0.6066	0.6157±0.0129	11
CySecBERT	0.6172	0.6115	0.6026	0.6229	0.6065	0.6121±0.0081	12

Table 8: Classification Task 2 : per-seed and aggregate results.

Model	s=1	s=2	s=3	s=4	s=42	mean±std	rank
llama-embed-nemotron-8b	0.6717	0.5637	0.7096	0.6661	0.6202	0.6463±0.0560	1
Qwen3-embedding-8B	0.5832	0.5825	0.6596	0.6769	0.6600	0.6325±0.0458	2
gte-Qwen2-7B-instruct	0.6262	0.5008	0.5745	0.6396	0.6643	0.6011±0.0650	3
Qwen3-embedding-0.6B	0.5799	0.5446	0.6350	0.6076	0.6148	0.5964±0.0350	4
multilingual-e5-large-instruct	0.5959	0.5226	0.6213	0.6073	0.6218	0.5938±0.0412	5
bge-m3	0.6769	0.4554	0.5577	0.5220	0.6348	0.5694±0.0883	6
jina-embeddings-v3	0.5746	0.5055	0.6121	0.5723	0.5749	0.5679±0.0386	7
aliyun_qwen3.7-text-embedding	0.4934	0.4161	0.6482	0.6159	0.6368	0.5621±0.1024	8
nomic-embed-text-v1.5	0.5862	0.4838	0.5614	0.5930	0.5688	0.5587±0.0437	9
CySecBERT	0.5248	0.4884	0.5762	0.5431	0.5696	0.5404±0.0357	10
SecureBERT	0.5014	0.4800	0.5827	0.6156	0.4753	0.5310±0.0641	11
CTI-BERT	0.5678	0.4092	0.4221	0.5242	0.5494	0.4945±0.0738	12

Table 9: Classification Task 3 : per-seed and aggregate results.

Model	s=1	s=2	s=3	s=4	s=42	mean±std	rank
llama-embed-nemotron-8b	0.6824	0.7179	0.7539	0.7204	0.7129	0.7175±0.0255	1
aliyun_qwen3.7-text-embedding	0.6806	0.7332	0.6902	0.7397	0.6845	0.7056±0.0284	2
gte-Qwen2-7B-instruct	0.6717	0.6858	0.7260	0.6606	0.6378	0.6764±0.0328	3
Qwen3-embedding-8B	0.6380	0.6753	0.7017	0.6480	0.6547	0.6635±0.0253	4
multilingual-e5-large-instruct	0.6394	0.6742	0.6875	0.6323	0.6266	0.6520±0.0271	5
Qwen3-embedding-0.6B	0.6277	0.6488	0.6869	0.6237	0.6256	0.6425±0.0268	6
bge-m3	0.6117	0.6543	0.6782	0.6298	0.6347	0.6418±0.0254	7
jina-embeddings-v3	0.5980	0.6171	0.6408	0.6000	0.5944	0.6100±0.0193	8
SecureBERT	0.4446	0.5314	0.5170	0.5172	0.5206	0.5062±0.0349	9
CySecBERT	0.4774	0.5092	0.4968	0.4865	0.4558	0.4851±0.0202	10
CTI-BERT	0.4659	0.4460	0.4424	0.4195	0.4511	0.4450±0.0168	11
nomic-embed-text-v1.5	0.4347	0.4027	0.4369	0.4536	0.4796	0.4415±0.0281	12

A.3 Non-Neural Reference Points

To establish how much of each classification subtask is reachable without a learned representation, we evaluate two reference points under a protocol identical to the one used for the embedding models: a TF-IDF vectorizer over word unigrams and bigrams, and a random-embedding control that fixes one Gaussian vector per instance and varies only the split. Both feed the same cross-validated logistic-regression classifier. Results are in Table 10.

The three subtasks answer this question differently. C1 should be read as measuring how well an encoder preserves domain-specific lexical evidence, not as evidence that the task demands deep semantics. C2 requires mapping a CWE name onto an abstract Pillar whose own name shares few words with it, and C3 requires reading bilingual advisory prose, neither of which surface n-grams support.

Table 10: Classification reference points under the protocol of the embedding models.

Baseline	C1 (37-way CWE) F1	C1 (37-way CWE) Rank	C2 (10-way Pillar) F1	C2 (10-way Pillar) Rank	C3 (6-way threat) F1	C3 (6-way threat) Rank
TF-IDF	0.7486 ± 0.0075	4 / 13	0.5196 ± 0.0201	12 / 13	0.4096 ± 0.0193	13 / 13
Random-emb.	0.0343 ± 0.0051	13 / 13	0.0543 ± 0.0046	13 / 13	0.1671 ± 0.0262	13 / 13

A.4 Language Composition of Each Task

Table 11 reports the measured language composition of every task.

Table 11: Language composition of SecEmbedEval by task. The two percentage columns give the share of script-bearing characters that are CJK versus Latin.

Task	Instances	CJK	Latin	Dominant source family
STS-1	1,178	88.7%	11.3%	Crawled reports and blogs
STS-2	1,281	72.8%	27.2%	Security term inventories
STS-1-en	866	0.0%	100.0%	Crawled reports and blogs
STS-2-en	1,194	0.0%	100.0%	Security term inventories
STS-3	4,000	0.0%	100.0%	NVD feeds
C1	1,850	0.0%	100.0%	NVD feeds
C2	934	0.0%	100.0%	MITRE CWE catalogue
C3	1,200	51.5%	48.5%	Threat reports and advisories
Clustering	3,600	47.7%	52.3%	Threat reports and advisories
Retrieval (queries)	800	74.5%	25.5%	Derived from crawled corpus
Retrieval (corpus)	6,000	74.9%	25.1%	Mixed crawled sources

A.5 Contamination and Memorization Analysis

Benchmarks assembled from public sources risk overlapping the pretraining corpora of the models they evaluate. We cannot inspect proprietary training corpora directly, so we test for the observable signature of memorization instead. If a model had memorized the association between a CVE identifier and its description, that advantage would be available for identifiers published before its training cutoff and unavailable for identifiers published after it.

STS-3 supports this test directly, because its instances carry publication years spanning 1999–2026 and 36.0% of them are from 2024 or later. Table 12 stratifies STS-3 AUC by CVE publication year. The result is flat: averaged over the 12 models, AUC is 0.512 for CVEs published in 2022 or earlier (2,334 pairs), 0.532 for 2023–2024 (676), 0.515 for 2025 (378), and 0.517 for 2026 (612). The span is 0.02 with no monotone trend, and every bucket is within noise of the 0.50 chance level. Records from 2026 postdate the training cutoff of every open checkpoint we evaluate, and the commercial API system does not disclose one, yet performance there is no worse than on records from 2022 and earlier. We find no evidence that memorization inflates STS-3. There is no advantage to detect, because no model solves the task at all.

Table 12: STS-3 AUC stratified by CVE publication year.

Model	$\leq$2022	2023–2024	2025	2026	Overall
llama-embed-nemotron-8b	0.552	0.568	0.522	0.526	0.549
CTI-BERT	0.538	0.554	0.507	0.496	0.531
aliyun_qwen3.7-text-embedding	0.520	0.502	0.545	0.544	0.523
Qwen3-embedding-8B	0.507	0.532	0.559	0.524	0.519
multilingual-e5-large-instruct	0.526	0.532	0.516	0.479	0.518
gte-Qwen2-7B-instruct	0.510	0.529	0.478	0.519	0.512
Qwen3-embedding-0.6B	0.502	0.531	0.532	0.507	0.511
jina-embeddings-v3	0.496	0.503	0.477	0.572	0.509
bge-m3	0.503	0.528	0.476	0.525	0.509
CySecBERT	0.504	0.546	0.515	0.480	0.508
SecureBERT	0.485	0.551	0.529	0.509	0.505
nomic-embed-text-v1.5	0.501	0.503	0.524	0.523	0.506
Mean	**0.512**	**0.532**	**0.515**	**0.517**	**0.517**

A.6 English Parallel Subsets

Table 13 reports all 12 models on the English parallel subsets STS-1-en and STS-2-en, whose construction is described in §3.2. It is the per-model backing for the group-gap and rank-agreement analysis of §5. Task 3 has no English counterpart, since it grounds bare CVE identifiers, which carry no language. The two languages are distinct item sets, so Δ gives the direction of a model's language sensitivity rather than a within-model controlled contrast.

Table 13: English parallel subsets of Tasks 1 and 2 over all 12 models, ordered by Task 1-en AUC.

Model	Task 1-en AUC	Task 1-en AP	Task 1-en Δ vs. zh	Task 2-en AUC	Task 2-en AP	Task 2-en Δ vs. zh
bge-m3	**0.9421** [0.927, 0.957]	**0.9406**	−0.0294	0.6430 [0.611, 0.674]	0.6643	+0.0221
aliyun_qwen3.7-text-embedding	**0.9385** [0.924, 0.953]	0.9393	−0.0467	**0.7553** [0.726, 0.784]	**0.7694**	−0.0144
jina-embeddings-v3	0.9231 [0.905, 0.941]	0.9177	−0.0154	0.6921 [0.660, 0.722]	0.7041	+0.0544
llama-embed-nemotron-8b	0.9023 [0.882, 0.922]	0.9012	−0.0707	**0.7524** [0.723, 0.781]	0.7490	+0.1037
multilingual-e5-large-instruct	0.8989 [0.877, 0.920]	0.8773	+0.0083	0.6369 [0.605, 0.667]	0.6426	+0.0475
CTI-BERT	0.8023 [0.772, 0.831]	0.7875	+0.0521	0.5314 [0.499, 0.563]	0.4780	+0.2200
Qwen3-embedding-8B	0.7985 [0.770, 0.828]	0.7851	−0.1222	0.6303 [0.600, 0.665]	0.5953	+0.0801
Qwen3-embedding-0.6B	0.7924 [0.763, 0.823]	0.7767	−0.1652	0.6051 [0.574, 0.637]	0.5640	+0.0480
SecureBERT	0.7795 [0.749, 0.809]	0.7633	−0.1149	0.5215 [0.489, 0.554]	0.5023	+0.0882
CySecBERT	0.7572 [0.724, 0.789]	0.7395	−0.0253	0.5085 [0.474, 0.540]	0.4678	+0.1504
gte-Qwen2-7B-instruct	0.7433 [0.712, 0.775]	0.7284	−0.0729	0.7407 [0.711, 0.770]	0.7441	+0.1323
nomic-embed-text-v1.5	0.7398 [0.707, 0.774]	0.7397	+0.1929	0.6285 [0.596, 0.659]	0.6408	+0.2066

A.7 Statistical Significance Tests

To determine whether top-performing models on each task differ significantly or fall within overlapping confidence intervals, we test the best model against the second-best on every subtask. For similarity and retrieval, where per-item scores are available, we use a paired bootstrap over items (10,000 resamples, two-sided, $\alpha = 0.05$, seed 20260825), resampling the same items for both models so that the comparison is paired at the item level. For classification, where the metric is computed over a whole test split rather than per item, we instead pair on the five random seeds and apply a paired t-test over the resulting five macro-F1 values. This has only four degrees of freedom and is correspondingly conservative. Because the eleven tests are read together as one family of leader comparisons, we also report Holm-corrected p-values and treat those as decisive. Results are reported in Table 14.

Table 14: Paired significance tests. "sig." requires the Holm-corrected p-value at $\alpha = 0.05$.

Task	Top-1 Model	Top-1 Score	Top-2 Model	Top-2 Score	p	p (Holm)	Sig.
Retrieval (NDCG@10)	aliyun-qwen3.7	0.7439	gte-Qwen2-7B	0.7338	0.0488	0.3416	n.s.
Retrieval (Hit@10)	aliyun-qwen3.7	0.9812	Qwen3-Emb-8B	0.9688	0.0096	0.0768	n.s.
Retrieval (MRR)	aliyun-qwen3.7	0.9292	Qwen3-Emb-8B	0.9166	0.0490	0.3416	n.s.
STS-1 (AUC)	aliyun-qwen3.7	0.9852	llama-emb-8b	0.9731	0.0004	0.0040	sig.
STS-2 (AUC)	aliyun-qwen3.7	0.7697	llama-emb-8b	0.6487	< 0.0001	< 0.001	sig.
STS-3 (AUC)	llama-emb-8b	0.5487	CTI-BERT	0.5315	0.1586	0.7930	n.s.
STS-1-en (AUC)	bge-m3	0.9421	aliyun-qwen3.7	0.9385	0.5720	1.0000	n.s.
STS-2-en (AUC)	aliyun-qwen3.7	0.7553	llama-emb-8b	0.7524	0.7060	1.0000	n.s.
C1 (macro-F1)	llama-emb-8b	0.8243	aliyun-qwen3.7	0.7927	0.0028	0.0255	sig.
C2 (macro-F1)	llama-emb-8b	0.6463	Qwen3-Emb-8B	0.6325	0.5944	1.0000	n.s.
C3 (macro-F1)	llama-emb-8b	0.7175	aliyun-qwen3.7	0.7056	0.4852	1.0000	n.s.

A.8 Clustering Stability Across Random States

To validate the stability of clustering results, we re-evaluated all models using KMeans with 10 different random state seeds ($\{1, 2, \ldots, 9, 42\}$), keeping all other protocol elements identical. DBSCAN and agglomerative clustering are deterministic and do not require re-evaluation. Table 15 reports the primary metric, BCubed F1, together with NMI and ARI aggregated across the 10 runs.

Table 15: KMeans clustering stability across 10 random state seeds ($\{1, \ldots, 9, 42\}$), ordered by mean BCubed F1.

Model	BC-F1 mean	BC-F1 std	NMI mean	NMI std	ARI mean	ARI std	Rank stab.
aliyun_qwen3.7-text-embedding	0.3652	0.0108	0.2462	0.0472	0.1715	0.0496	0.46 (1–2)
gte-Qwen2-7B-instruct	0.3578	0.0013	0.2096	0.0277	0.1467	0.0313	0.60 (1–3)
multilingual-e5-large-instruct	0.3400	0.0076	0.1382	0.0032	0.0589	0.0076	1.56 (3–7)
Qwen3-embedding-8B	0.3393	0.0121	0.2270	0.0201	0.1544	0.0205	2.14 (2–8)
llama-embed-nemotron-8b	0.3366	0.0011	0.1557	0.0081	0.0575	0.0077	0.87 (4–7)
jina-embeddings-v3	0.3347	0.0004	0.1272	0.0004	0.0549	0.0006	1.02 (5–9)
Qwen3-embedding-0.6B	0.3322	0.0094	0.1651	0.0089	0.1098	0.0122	1.68 (4–10)
bge-m3	0.3267	0.0066	0.1344	0.0102	0.0764	0.0146	1.17 (5–9)
nomic-embed-text-v1.5	0.3251	0.0148	0.1408	0.0040	0.0725	0.0122	2.45 (3–10)
CTI-BERT	0.3084	0.0008	0.1243	0.0005	0.0497	0.0007	0.46 (9–10)
CySecBERT	0.2756	0.0114	0.0479	0.0038	0.0169	0.0031	0.00 (11)
SecureBERT	0.2660	0.0002	0.0361	0.0002	0.0088	0.0001	0.00 (12)

A.9 Evaluation Protocol and Metric Definitions

This appendix expands §4, giving the label scheme, metric family, and reference points of each task family in full.

A.9.1 Semantic Similarity

For sentence-pair and phrase-pair similarity, we compute cosine similarity between paired embeddings and evaluate predictions as binary discrimination, reporting AUC and average precision. The sentence-level subtask targets longer cybersecurity texts, whereas the phrase-level subtask focuses on technical terms and closely related expressions. For CVE identifier–description matching, we use the same embedding-based similarity framework and likewise evaluate it as a binary matching problem over positive and negative pairs. All three similarity subtasks therefore share a single label scheme and a single metric family.

A.9.2 Retrieval

The retrieval task evaluates threat intelligence search over a heterogeneous document collection. The protocol runs five stages, shown in the retrieval lane of Figure 2: each model vectorizes the corpus and the queries, retrieves its top-K passages by cosine similarity, the retrieved candidates are pooled across models, the pool is judged for relevance, and every model is scored against the resulting judgment set (Karpukhin et al., 2020; Xiong et al., 2021). We release this as a single configurable pipeline rather than as a fixed dataset. One configuration file declares the embedding models under test, the instruction prefix each model card specifies, and the judge LLM. The prefix is an explicit per-model field because omitting the prefix an instruction-tuned retriever expects depresses its scores while still yielding plausible-looking numbers. Judging and scoring are separate commands, so cached ranked lists can be rescored against a revised judgment set without re-encoding the corpus.

Relevance is recorded in BEIR-style qrels on a two-level scale. Grade 2 marks a passage that answers the query directly or provides substantial support for answering it, grade 1 marks partial or peripheral support, and a document is counted as relevant at either grade. Each model nominates its top-20 passages per query, which gives 192,000 nominations over the twelve systems, and these collapse to 105,038 distinct query–passage pairs.

A.9.3 Clustering

The clustering task evaluates whether embeddings preserve latent structure in an unsupervised setting. Because clustering outcomes can vary substantially across algorithms, we evaluate each embedding space with three paradigms: KMeans, DBSCAN, and Agglomerative Clustering.

Since 41.9% of instances carry multiple threat categories (§3), the primary external metric is extended BCubed precision and recall (Amigó et al., 2009), which is defined directly over multi-label ground truth.

Two reference points are reported alongside every clustering result, because BCubed F1 is not interpretable in isolation.

The first is the *trivial-solution ceiling*. Assigning all instances to a single cluster yields recall 1.0 by definition and precision equal to the dataset's expected label-agreement rate, which here gives BCubed F1 $= 0.524$, higher than any genuine clustering any model produces. We flag such solutions using two criteria, at most one non-noise cluster or a single cluster absorbing more than 90% of non-noise points, and exclude them from best-algorithm selection.

The second criterion catches solutions that report many clusters while placing nearly everything in one of them. The second reference point is the *meaningful floor*. Clustering random 32-dimensional Gaussian vectors with the same algorithm and k gives BCubed F1 $= 0.228$, the level a model must exceed to have contributed any structure at all. A third reference point, TF-IDF features under the identical clustering protocol, is reported in Appendix A.3.

A.10 Experimental Protocol Details

This section details the evaluation protocol to ensure full reproducibility.

Pooling strategy: Every task family reads out each checkpoint the way that checkpoint was trained to be read out. The read-out is resolved per model from its own $1_Pooling/config.json$.

Instruction prefixes: On the retrieval task, each checkpoint is encoded with the asymmetric query/document prefix its own model card prescribes. The multilingual-e5 family receives "$query: $" / "$passage: $", and nomic-embed-text-v1.5 receives "$search_query: $" / "$search_document: $". The Qwen3-embedding-8B, Qwen3-embedding-0.6B, and gte-Qwen2-7B-instruct checkpoints receive their published "$Instruct: \{task\}\nQuery: $" retrieval template on the query side and no prefix on the document side. The remaining checkpoints prescribe no prefix and receive none, and no prefix is applied on STS, classification, or clustering.

Hardware and Environment: All open-source model inference was conducted on 8 NVIDIA A800-SXM4-80GB GPUs (NVIDIA driver 560.35.03, CUDA 12.6). The commercial API system (aliyun_qwen3.7-text-embedding) was accessed via DashScope API in September 2026 with a per-request batch size limit of 20 (enforced to avoid batch-related errors on the provider side).

Table 16 consolidates the settings needed to reproduce every reported number.

Table 16: Consolidated reproducibility details.

Hardware	NVIDIA A800-SXM4-80GB, FP32 inference
Software	PyTorch 2.2.1 (built with CUDA 12.1. Driver-side CUDA 12.6), transformers 4.46.3, sentence-transformers 3.2.1 scikit-learn 1.3.2, scipy 1.10.1, numpy 1.24.1
Encoding	max length 512, batch 32 (API model: 20), float64 cosine
Pooling	checkpoint-native throughout: CLS (bge-m3). Last-token (gte-Qwen2-7B, Qwen3-Emb-8B/0.6B). Mean (all other local models)
Retrieval read-out	same checkpoint-native pooling. Per-model instruction prefixes (Qwen3-Emb/gte: task instruction, e5: $query:$/$passage:$, nomic: $search_query:$/$search_document:$)
Prefixes on STS	e5 ($query:$/$passage:$) and nomic ($search_query:$/$search_document:$) only. None for the other ten models. None on CLS/CLU
CLS protocol	60/20/20 stratified split. LogisticRegressionCV, 5-fold on the training split, $C \in [10^{-3}, 10^{2}]$
CLS seeds	1, 2, 3, 4, 42 (5 repetitions, mean $\pm$ std reported)
CLU protocol	KMeans (k = #classes) over 10 random states. DBSCAN/Agglomerative deterministic
Retrieval	6,000-document corpus, 800 queries, 12,964 pooled qrels (BEIR format)
Bootstrap CIs	1,000 resamples, seed 20260727
Significance tests	paired bootstrap, 10,000 resamples, two-sided, seed 20260825
Qwen3 loading	Qwen2-backbone compatibility layer (cosine $= 1.0$ vs. reference)
API model	DashScope $qwen3.7\text{-}text\text{-}embedding$, 1024-d, batch ≤ 20

B Dataset Construction Details

B.1 Task-Level Dataset Statistics

Table 17 reports the per-task composition of SecEmbedEval, including instance counts, label spaces, and typical input lengths for each subtask.

Table 17: Task-level dataset statistics for SecEmbedEval

Category	Subtask / Unit	Instances	Classes / Labels	Avg. Length	Notes
Semantic similarity	STS-1 sentence pairs	1,178	binary	94 chars (median)	589 positive / 589 negative over 589 anchors across 12 security domains. Each anchor contributes exactly one positive and one negative, so the two classes share an identical topic distribution.
Semantic similarity	STS-2 phrase / term pairs	1,281	binary	5 chars (median)	612 positive / 669 negative from 138 seed terms. Positives are synonyms, abbreviations and cross-lingual equivalents, negatives are same-family siblings.
Semantic similarity	STS-1-en sentence pairs	866	binary	179 chars / 26 words (median)	English parallel subset: 433 positive / 433 negative over 433 anchors drawn from 435 English source units, same twelve domains at the distribution the English stream supplies.
Semantic similarity	STS-2-en phrase / term pairs	1,194	binary	18 chars (median)	English parallel subset: 526 positive / 668 negative from 136 seed terms. Cross-lingual positives are excluded by construction, so language is not a factor within the subtask.
Semantic similarity	STS-3 CVE–description pairs	4,000	binary	id 14 chars / desc. 263 chars	2,000 positive and 2,000 negative identifier–description pairs. Negatives are 999 same-year and 1,001 random, with a Jaccard < 0.5 constraint on the paired descriptions.
Retrieval	queries + corpus	800 queries / 6,000 docs	graded 0/1/2	29 / 1,441 chars	Relevance from TREC-style pooling over the 12 evaluated models at depth 20.
Classification	C1: description $\rightarrow$ CWE	1,850	37	281 chars	Exactly 50 instances per class, each 40 real NVD records and 10 LLM-generated, a uniform ratio that decorrelates provenance from label.
Classification	C2: CWE $\rightarrow$ parent category	934	10	44 chars	Fine-to-coarse mapping onto the ten top-level Pillars of CWE View-1000. Labels follow the official hierarchy and the natural $32.1\times$ head-to-tail imbalance is preserved.
Classification	C3: threat description $\rightarrow$ threat category	1,200	6	134 chars	Exactly 200 per class, balanced simultaneously across the six categories and the years 2021–2026.
Clustering	threat descriptions	3,600	6 (multi-label)	134 chars	41.9% of instances carry more than one category. The natural $2.5\times$ label skew is retained. Scored with extended BCubed.
Efficiency	benchmark inputs	1,000 texts	–	32 / 128 / 256 tokens	Standardized synthetic workload spanning short, medium, and long inputs.

B.2 Dataset Provenance and Verification Statistics

This section records where each task's text and labels come from. Figure 6 gives an overview of the provenance and verification flow.

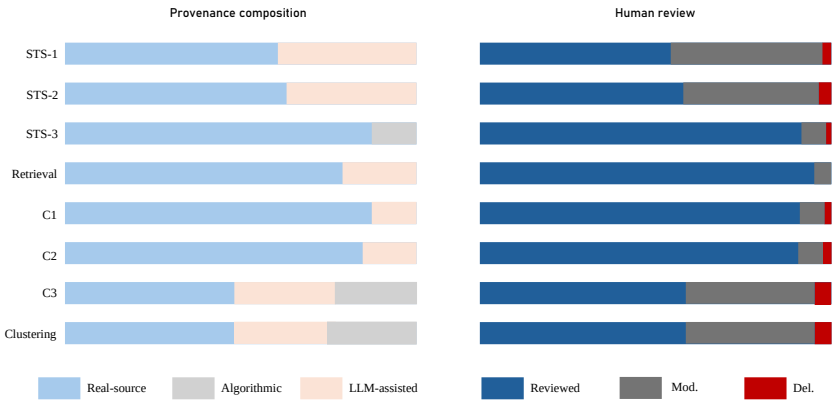

Figure 6: Provenance and verification flow.

B.3 Validation of LLM-Assisted Construction

Table 21 summarizes the admission check applied to each candidate set together with the agreement rate it measured, and Table 22 gives the templates that constrained generation and the review pass. Every rate in the table is produced by a pass that does not see the target it is checking.

C LLM-Assisted Construction Prompts

This section releases the prompt templates used to constrain candidate construction, query drafting, and candidate review. The prompts are written as audit artifacts for reproducibility and should be read together with the verification statistics in this appendix.

D Benchmark Positioning and Baselines

The nine evaluation axes of SecEmbedEval are not exercised evenly by existing benchmarks. Figure 7 compares SecEmbedEval with a general-purpose suite and with the two security-domain benchmarks closest to this work. Panel (a) reads the four resources down six properties that more than one of them possesses. All three security resources supply cybersecurity text, and SEvenLLM matches the Chinese and English coverage, but only SecEmbedEval and MTEB take a text embedding as the unit under test. CTIBench and SEvenLLM target reasoning, generation, and knowledge recall rather than embedding quality, so they hold neither the embedding-level nor the identifier-level properties, and MTEB, being general-purpose, holds no cybersecurity data and no identifier grounding. SecEmbedEval holds all six. Panel (b) counts the SecEmbedEval data types with no counterpart in the other three, from the relevance-judged retrieval pool to the standardized efficiency workload, 17,903 items under one protocol.

Table 18: Cross-task rank correlation on SecEmbedEval, as Spearman ρ over the 12 evaluated models. Bold marks $p < 0.05$, which at $n = 12$ requires $|\rho| > 0.58$.

	CLS	CLU	STS-1	STS-2	STS-3	NDCG@10
CLS	–	**0.87**	**0.64**	**0.83**	0.55	**0.78**
CLU	**0.87**	–	0.45	**0.71**	**0.60**	**0.87**
STS-1	**0.64**	0.45	–	**0.83**	0.34	0.52
STS-2	**0.83**	**0.71**	**0.83**	–	0.36	**0.62**
STS-3	0.55	**0.60**	0.34	0.36	–	0.47
NDCG@10	**0.78**	**0.87**	0.52	**0.62**	0.47	–

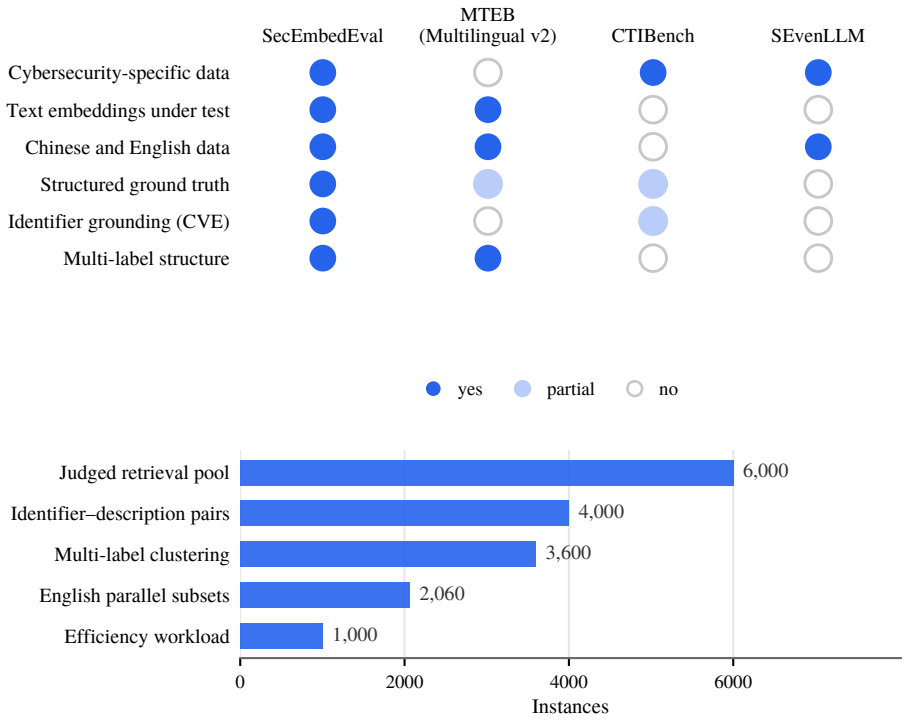

Figure 7: SecEmbedEval against a general-purpose suite and the two closest security-domain benchmarks. Panel (a) marks, for the properties more than one resource possesses, whether a resource has each one (filled), has a partial counterpart (lightly filled), or has none (open). Panel (b) counts the SecEmbedEval data types with no counterpart in the other three.

Table 24 summarizes why the evaluated models were selected. Together, they cover complementary axes including parameter scale, model family, instruction tuning, multilingual retrieval, domain specialization, and API-based deployment. The resulting comparison set is representative rather than exhaustive, supporting interpretable evaluation across open-source, domain-adapted, and commercial embedding systems.

D.1 SEE-Score and Transfer Discordance

Table 19 reports SEE-Score for all twelve evaluated models together with the six component metrics, ordered by the aggregate, where CLS is the three-task macro-F1 mean, CLU the ten-random-state KMeans BCubed F1 mean, STS-1, STS-2 and STS-3 the AUCs of the sentence-pair, term-pair and identifier-grounding subtasks, and RET the NDCG@10.

Table 19: SEE-Score over all twelve evaluated models, ordered by the aggregate. Best value per column in bold.

Model	CLS	CLU	STS-1	STS-2	STS-3	RET	SEE-Score
aliyun_qwen3.7-text-embedding	0.687	**0.365**	**0.985**	**0.770**	0.523	**0.744**	**0.804**
llama-embed-nemotron-8b	**0.729**	0.337	0.973	0.649	**0.549**	0.688	0.741
gte-Qwen2-7B-instruct	0.675	0.358	0.816	0.608	0.512	0.734	0.653
Qwen3-embedding-8B	0.670	0.339	0.921	0.550	0.519	0.732	0.638
bge-m3	0.653	0.327	0.972	0.621	0.509	0.692	0.635
multilingual-e5-large-instruct	0.671	0.340	0.891	0.589	0.518	0.691	0.633
jina-embeddings-v3	0.631	0.335	0.939	0.638	0.509	0.654	0.615
Qwen3-embedding-0.6B	0.639	0.332	0.958	0.557	0.511	0.698	0.605
nomic-embed-text-v1.5	0.578	0.325	0.547	0.422	0.506	0.425	0.273
SecureBERT	0.551	0.266	0.895	0.433	0.505	0.003	0.191
CTI-BERT	0.536	0.308	0.750	0.311	0.532	0.076	0.175
CySecBERT	0.546	0.276	0.783	0.358	0.508	0.014	0.137

Table 20 decomposes TD by task family, alongside Kendall's τ and Spearman's ρ over the same seven models. The gradient is monotone. Discordance doubles from retrieval ($4/21$), the family closest to what a general benchmark exercises, to sentence-level similarity ($8/21$), and reaches $15/21$ on terminology-level similarity, the subtask most specific to security operations.

Table 20: Per-family Transfer Discordance between the MTEB public ranking and the SecEmbedEval ranking, ordered by TD.

Task family	TD	τ	ρ
Retrieval (NDCG@10)	$4/21 = 0.19$	$+0.62$	$+0.71$
Classification (macro-F1)	$6/21 = 0.29$	$+0.43$	$+0.57$
Clustering (BCubed F1)	$7/21 = 0.33$	$+0.33$	$+0.50$
STS-1 sentence (AUC)	$8/21 = 0.38$	$+0.24$	$+0.25$
SEE-Score (aggregate)	$7/21 = 0.33$	$+0.33$	$+0.43$
STS-2 terminology (AUC)	$15/21 = 0.71$	-0.43	-0.25

E Additional Clustering Details

This section expands the clustering discussion from the main text. Table 4 in §5 reports the external metrics, Table 26 the internal ones, and Table 27 the class distribution retained for the clustering dataset.

F Dataset Case Examples

All examples in this appendix are shortened and sanitized from the underlying JSONL datasets. They preserve the semantic pattern being evaluated while removing or compressing operational detail that is unnecessary for benchmark interpretation. Tables 28 through 31 show examples for semantic similarity, classification, retrieval, and clustering respectively.

Table 21: Audit of LLM-assisted candidate construction.

Task	Admission check	Candidates	Admitted	Measured rate
STS-1	Constraint re-check at assembly: language match, length ratio in $[0.5, 2.0]$, bigram-Jaccard inside the band required of the relation, no duplicate generated text. A source is admitted only if *both* of its sides pass	600 sources	589 (1,178 pairs)	98.2% of sources yield an admissible pair. 11 were dropped for a missing admissible negative
STS-2	Second LLM pass judging the relation of each term pair with no access to the relation it was generated under	1,412 pairs	1,281 pairs	90.7% generation–verification concordance
STS-1-en	Same constraint re-check with the English guard: content-word Jaccard in place of character bigrams, length ratio counted in words, band $[0.15, 0.75]$ for negatives (§3.2)	435 sources	433 sources (866 pairs)	99.5% of sources yield an admissible pair. 2 were dropped for a missing admissible side
STS-2-en	Second LLM pass judging the relation of each term pair with no access to the relation it was generated under. Candidates containing CJK characters are additionally rejected	1,338 pairs	1,194 pairs	89.2% generation–verification concordance
C1 (aug.)	Independent LLM pass predicting the CWE class from the vulnerability text alone, compared against the assigned label	1,850 records	1,850 records	76.5% concordance on the 370 generated records, 64.1% on the 1,480 real NVD records
Retrieval	Three-grade relevance rubric applied to the pooled candidates. A subset was independently re-judged under the same rubric	105,038 (depth 20)	105,037 judged. 12,964 relevant	99.0% exact three-grade self-agreement on 15,511 re-judged pairs (99.4% binarized)

Table 22: Representative prompt templates for LLM-assisted construction and review.

Template	Purpose and inputs	Prompt text
STS pair construction	Input: source sentence(s), source metadata, target label regime, language constraint. Goal: produce a candidate cybersecurity sentence pair grounded in source evidence.	System: You are assisting with benchmark construction for cybersecurity text similarity. Generate one sentence pair grounded only in the provided source evidence. Preserve the technical meaning, avoid adding unsupported vulnerability details, and do not invent products, actors, or identifiers. User: Given SOURCE_TEXT, SOURCE_TYPE, and TARGET_RELATION in {semantic match, hard negative, partial overlap}, write Sentence A and Sentence B. Constraints: (1) keep both sentences cybersecurity-related. (2) Maintain realistic analyst wording. (3) If TARGET_RELATION is hard negative, preserve topical proximity while changing the operational meaning. (4) Return JSON with fields {"sentence_a", "sentence_b", "target_relation", "rationale"}.
STS term construction	Input: seed term, source snippet, target relation. Goal: construct terminology-level pairs covering synonymy, aliasing, and contrast.	System: You are generating cybersecurity term pairs for semantic relatedness evaluation. User: Using TERM and SUPPORTING_CONTEXT, create one paired term or phrase under TARGET_RELATION in {equivalent, related, unrelated, contrastive}. Constraints: (1) stay within common cybersecurity usage. (2) Allow mixed Chinese/English forms when realistic. (3) Avoid trivia or overly rare spellings unless present in the source. (4) Return JSON with {"term_a", "term_b", "relation", "explanation"}.
Retrieval query drafting	Input: source passage, source family, intended information need. Goal: write a retrieval query that can be answered by cross-source evidence.	System: You are drafting evaluation queries for cybersecurity retrieval. Queries should reflect realistic analyst search intent rather than keyword lists. User: Given PASSAGE, SOURCE_FAMILY, and QUERY_FOCUS in {vulnerability cause, exploitation condition, impact, mitigation, attribution, timeline}, write one query. Constraints: (1) the query must be answerable from the source or closely related public evidence. (2) Prefer paraphrastic wording over direct copying. (3) Include identifiers such as CVE, CWE, ATT&CK, product, or malware family only when present or strongly implied by the source. (4) Return JSON with {"query", "focus", "answer_evidence_type", "notes"}.
Classification augmentation	Input: source description, existing label set, minority class target. Goal: produce controllable augmentation without changing the core security behavior.	System: You are creating a benchmark candidate for cybersecurity classification. User: Using SOURCE_DESCRIPTION and TARGET_LABEL, rewrite or lightly augment the text so that the same vulnerability or threat behavior remains intact. Constraints: (1) preserve the label-defining behavior. (2) Do not insert exploit steps, payloads, or unnecessary offensive detail. (3) Keep the style consistent with advisories or analyst summaries. (4) Return JSON with {"augmented_text", "target_label", "preserved_cues"}.
Review and filtering	Input: candidate item, source evidence, provisional label. Goal: decide whether to accept, modify, or reject an LLM-produced candidate before it is admitted to the benchmark.	System: You are assisting quality control for a cybersecurity benchmark. You are not the final annotator. You must provide a conservative review recommendation grounded in the source evidence. User: Evaluate CANDIDATE_ITEM against SOURCE_EVIDENCE and PROVISIONAL_LABEL. Check semantic faithfulness, taxonomy consistency, cross-source coherence, and operational sensitivity. Output one decision in {accept, modify, reject}. If modify or reject is selected, explain the minimal issue precisely. Return JSON with {"decision", "issue_type", "suggested_fix", "confidence"}.

Table 23: Public general-benchmark strength versus SecEmbedEval transfer.

Model	Public standing	SecEmbedEval pattern	Largest gap	Observation
Qwen3-Emb-8B	MTEB / MMTEB comparisons: strong	strong on structural tasks, weak on terminology	terminology-level STS	Holds the highest public score in the comparison set and converts it on the tasks that reward topical organization, at 4th in clustering and 3rd in retrieval, but falls to 8th of 12 on terminology-level similarity (AUC 0.5502), behind models less than a tenth its size.
Qwen3-Emb-0.6B	MTEB / MMTEB comparisons: competitive for size	efficient and competitive	terminology-level STS	The compact model tracks its 8B sibling closely on retrieval (4th, 0.6983) at a fraction of the cost, and shares its weakness on terminology-level discrimination (7th), where it also loses ground on classification (7th).
bge-m3	public multilingual retrieval: strong	sharply task-dependent	clustering	Third of 12 on sentence-level similarity (AUC 0.9715) and 5th in retrieval, but 8th in clustering: its embedding geometry separates pairs well without organizing a corpus into threat categories.
e5-large-inst.	MTEB / MMTEB comparisons: strong	uniformly mid-tier	sentence-level STS	Places between 3rd and 8th on every one of the six metrics. The most consistent model in the study, and the least distinguished on any axis.
llama-emb-8b	model-card reports: strong	strongest open model on similarity and classification	retrieval	The sharpest task mismatch among the strong models: 1st on classification and on identifier grounding, 2nd on both similarity subtasks, yet 7th in retrieval, so its pairwise semantic strength does not convert into ranking a 6,000-document corpus. It also costs 29 GB of GPU memory and 17 samples/s.
aliyun-qwen3.7	vendor MTEB / CMTEB: strong	strongest overall	identifier grounding	Ranks 1st on four of the six metrics (clustering, both similarity subtasks and retrieval) and 2nd on classification. Its one ordinary result is STS-3 (AUC 0.5230, 3rd), where it is as close to chance as everything else, so the capability gap is not one that commercial scale has closed.
gte-Qwen2-7B	MTEB / C-MTEB: strong	strongest open retriever	sentence-level STS	The mirror image of llama-emb-8b: 2nd in retrieval and in clustering, 3rd in classification, but 9th of 12 on sentence-level similarity (AUC 0.8161). Corpus-scale ranking and pairwise similarity are separable capabilities, and this pair of models separates them.
jina-emb-v3	public multilingual retrieval: strong	efficient mid-tier	retrieval	Third on terminology-level similarity and the best retrieval-per-GPU-byte trade-off among general models, but 8th in retrieval itself, so its case is an efficiency argument rather than an accuracy one.
nomic-emb-v1.5	model-card reports: competitive for size	weak on fine-grained tasks	sentence-level STS	The only general-purpose model that breaks from the leading band: last of 12 on sentence-level similarity (AUC 0.5469, near chance) and 9th in retrieval at 0.4249, roughly 40% below the tier above it.

Table 23 (continued)

Model	Public standing	SecEmbedEval pattern	Largest gap	Observation
SecureBERT / CySecBERT / CTI-BERT	domain-encoder literature: domain-specialized	weakest overall	retrieval and STS	Domain pretraining alone does not confer transfer. Two language controls that are independent of each other point the same way: on the entirely English classification subtasks the group gap is about 0.09 macro-F1, and on the English parallel similarity subsets it is 0.0735 and 0.1555, with all three encoders at chance (0.5085–0.5314) on the English security terminology they were pretrained on. Language coverage accounts for roughly a third of the phrase-level deficit. The rest is a property of the encoders. CTI-BERT's 2nd place on STS-3 (AUC 0.5315) is a ranking inside the near-chance band rather than a capability.

Table 24: Baseline selection rationale.

Baseline	Model family	Scale (params)	Deployment setting	Why included
Qwen3-Emb-0.6B	open-source general embedding	0.6B	local	Represents a compact modern embedding model with favorable efficiency and strong public multilingual benchmark competitiveness relative to its size.
Qwen3-Emb-8B	open-source general embedding	8B	local	Represents a large-capacity embedding model from the same family, enabling within-family analysis of parameter scale effects.
bge-m3	multilingual retrieval-oriented embedding	0.6B	local	Included as a widely used multilingual retrieval baseline with strong public performance and broad adoption in dense retrieval settings.
e5-large-inst.	instruction-tuned multilingual embedding	0.6B	local	Represents instruction-tuned retrieval-focused embeddings that are consistently strong on public benchmarks and widely used for zero-shot semantic matching.
llama-emb-8b	decoder-based embedding model	8B	local	Included to cover high-capacity decoder-style embedding architectures rather than only encoder-dominant baselines.
aliyun-qwen3.7	commercial API embedding	undisclosed	API	Provides a production-oriented commercial baseline with publicly reported benchmark performance and realistic deployment relevance.
gte-Qwen2-7B	instruction-tuned decoder embedding	7B	local	Adds a second large instruction-tuned family so that scale effects are not observed only within the Qwen3 line.
jina-emb-v3	multilingual retrieval-oriented embedding	0.6B	local	Represents a recent long-context multilingual retriever with a strong efficiency profile.
nomic-emb-v1.5	open-source general embedding	0.14B	local	Covers the compact open-weight tier below 0.6B, testing how far parameter reduction can go before domain performance degrades.
SecureBERT / CySecBERT / CTI-BERT	cybersecurity-domain models	~0.11B each	local	Serve as domain-adapted cybersecurity baselines, allowing us to test whether domain pretraining alone improves transfer to the benchmark tasks.

Table 25: Degenerate-solution audit over all six clustering configurations. A run is degenerate if it yields at most one non-noise cluster or if a single cluster absorbs more than 90% of non-noise points. DBSCAN ε values are 0.3, 0.5 and 0.7 under cosine distance with min_samples $= 5$.

Configuration	Degenerate runs	BCubed F1 range	Status
KMeans ($k = 6$)	0/12	0.266–0.365	reported
Agglomerative, Euclidean ($k = 6$)	0/12	0.258–0.435	reported
DBSCAN, $\varepsilon = 0.3$	10/12	0.402–0.524	excluded
DBSCAN, $\varepsilon = 0.5$	12/12	0.499–0.524	excluded
DBSCAN, $\varepsilon = 0.7$	12/12	0.524–0.524	excluded
Agglomerative, cosine ($k = 6$)	12/12	0.524–0.525	excluded

Table 26: Clustering internal metrics for the two non-degenerate configurations, in the same model order as Table 4. Sil is silhouette, CH Calinski–Harabasz and DB Davies–Bouldin. KMeans internal metrics are computed on a single initialization.

Model	KMeans ($k = 6$)			Agglomerative ($k = 6$)		
	Sil	CH	DB	Sil	CH	DB
aliyun-qwen3.7	0.035	84.5	4.47	0.022	75.6	3.88
gte-Qwen2-7B	0.045	94.7	4.13	0.038	85.6	4.43
e5-large-inst.	0.031	76.1	4.58	0.022	67.5	4.35
Qwen3-Emb-8B	0.048	123.1	3.30	0.033	111.5	3.07
llama-emb-8b	0.046	98.4	3.93	0.042	89.5	3.56
jina-emb-v3	0.023	96.4	3.80	0.013	81.6	4.47
Qwen3-Emb-0.6B	0.058	129.7	3.18	0.037	108.2	3.59
bge-m3	0.038	84.5	4.33	0.032	74.5	3.86
nomic-emb-v1.5	0.028	69.9	4.66	0.012	58.6	4.99
CTI-BERT	0.050	168.3	3.16	0.038	131.3	3.44
CySecBERT	0.071	362.1	2.55	0.024	311.2	3.29
SecureBERT	**0.120**	**832.1**	**1.89**	**0.085**	**749.1**	**2.08**

Table 27: Composition and label overlap of the clustering set. The right-hand block is asymmetric, so cell (A, B) gives the percentage of the advisories carrying A that also carry B. Column keys are CI code injection, IL information leakage, DoS denial of service, AB authentication bypass, PE privilege escalation, CmdI command injection.

Threat category	Seed	Occ.	% inst.	Also carries (% of row category)					
				CI	IL	DoS	AB	PE	CmdI
Code injection (代码注入)	600	1,527	42.4	—	51	56	2	6	3
Information leakage (信息泄漏)	600	1,157	32.1	68	—	38	8	3	4
Denial of service (拒绝服务)	600	1,076	29.9	79	41	—	1	9	3
Authentication bypass (身份验证绕过)	600	664	18.4	4	14	2	—	5	1
Privilege escalation (权限提升)	600	604	16.8	15	7	16	5	—	1
Command injection (命令注入)	600	600	16.7	6	8	5	1	1	—

Categories per instance: 1: 2,093 (58.1%) 2: 1,028 (28.6%) 3: 437 (12.1%) 4: 42 (1.2%)
Publication year: 2021: 606 2022: 603 2023: 593 2024: 599 2025: 608 2026: 591

Table 28: Representative semantic similarity examples

Subtask	Field 1 (e.g., original_term / Sentence1 / cve_id)	Field 2 (e.g., generated_term / Sentence2 / cve_description)	Label	Interpretation
phrase-pair similarity	零日漏洞	*0day* 漏洞	1	Near-synonym and mixed-script normalization, a positive (related) pair.
phrase-pair similarity	零日漏洞	已修复漏洞	0	Hard negative with lexical overlap but different operational status, an unrelated pair.
CVE–description matching	*CVE-2023-50572*	*An issue in the component GroovyEngine.execute of jline-groovy v3.24.1 allows attackers to cause an OOM (OutofMemory) error.*	match	Identifier grounding between symbolic reference and natural-language description.
CVE–description matching	*CVE-2023-51075*	*hutool-core v5.8.23 was discovered to contain an infinite loop in the StrSplitter.splitByRegex function. This vulnerability allows attackers to cause a Denial of Service (DoS) via manipulation of the first two parameters.*	match	Long-form vulnerability description linked to a CVE identifier.

Table 29: Representative classification examples.

Subtask	Input field(s) (e.g., cve_description / cwe_id / title+comment)	Target label field
description → CWE	某 *ERP* 系统将用户输入直接写入审计日志，恶意换行符被解析为独立事件。	*CWE-117*
description → CWE	*Apache Log4j2 2.0-beta9 through 2.15.0 does not protect against attacker controlled LDAP and other JNDI related endpoints. This vulnerability allows attackers to execute arbitrary code by crafting a special request that gets logged by the application.*	*CWE-117*
CWE → parent category	*CWE-79*	输入验证和表示
description → threat category	*title:* 多款 *Cisco* 产品安全漏洞 *(CVE-2021-1291). Comment excerpt: ...* 没有正确验证 *HTTP* 请求，可能允许未经身份验证的远程攻击者执行任意代码。	不正确的输入验证

Table 30: Representative retrieval examples in query–positive format.

Positive passage / evidence	Query	Subtask
... 大型组织往往会推迟安装 *Windows* 更新补丁程序，因为从历史上看，一些更新会导致与第三方软件不兼容。... 不安装更新会有权衡和严重的后果。... 仍然有一些机器的所有者在两年多的时间里没有安装操作系统更新...	为什么 *WannaCry* 的流行在过去和现在都得益于大型组织的更新延迟？	query → positive evidence
漏洞名称：*Windows* 远程桌面授权服务远程代码执行漏洞（*CVE-2024-38077*）... 官方解决方案：微软已发布官方补丁包... 官方修复建议：通过微软官方链接下载并更新补丁。	针对 *Windows* 远程桌面授权服务的 *CVE-2024-38077* 漏洞，微软发布了什么补丁？	query → positive evidence
WannaCry detections continue to rise, implying that some machine owners have still not installed operating-system updates after more than two years. Large organizations often delay patch installation, and some administrators or individual users remain reluctant to apply updates despite the risk.	*What types of owners left their machines unpatched for more than two years, making them vulnerable to WannaCry?*	query → positive evidence
... 任何未授权的监听都会破坏量子系统的状态，从而被立即检测到。... 任何未经授权的测量都会引起量子系统的熵增，这种熵增是可以被检测到的。	如何通过量子密钥分发（*QKD*）检测到未授权的监听行为？	query → positive evidence

Table 31: Representative clustering examples.

Subtask	title	comment excerpt	Label field
threat description grouping	*Cisco Small Business* 安全漏洞 *(CVE-2021-1295)*	... 基于 *web* 的管理界面中的多个漏洞可能允许未经身份验证的远程攻击者作为受影响设备上的根用户执行任意代码。存在这些漏洞是因为没有正确验证 *HTTP* 请求。	不正确的输入验证
threat description grouping	*Axis Devices AXIS OS* 输入验证错误漏洞 *(CVE-2021-31988)*	... 没有正确验证与 *SMTP* 测试功能相关的控制参数，因此可能添加回车换行字符，并在生成的电子邮件中包含任意 *SMTP* 标头。	*CRLF* 注入
threat description grouping	*Domoticz injection vulnerability (CVE-2019-10678)*	*A vulnerability was discovered in Domoticz and classified as critical. The issue affects unknown code in the End of Line Handler component. Upgrading to version 4.10579 resolves the vulnerability.*	*CRLF* 注入