arXiv:2509.21310v1 [cs.AI] 25 Sep 2025

SAGE: A Realistic Benchmark for Semantic Understanding

Samarth Goel
University of California, Berkeley
Berkeley, CA 94704
sgoel9@berkeley.edu

Reagan J. Lee
University of California, Berkeley
Berkeley, CA 94704
reaganjlee@berkeley.edu

Kannan Ramchandran
University of California, Berkeley
Berkeley, CA 94704
kannanr@eecs.berkeley.edu

Abstract

As large language models (LLMs) achieve strong performance on traditional benchmarks, there is an urgent need for more challenging evaluation frameworks that probe deeper aspects of semantic understanding. We introduce SAGE (Semantic Alignment & Generalization Evaluation), a rigorous benchmark designed to assess both embedding models and similarity metrics across five categories: Human Preference Alignment, Transformation Robustness, Information Sensitivity, Clustering Performance, and Retrieval Robustness. Unlike existing benchmarks that focus on isolated capabilities, SAGE evaluates semantic understanding through adversarial conditions, noisy transformations, and nuanced human judgment tasks across 30+ datasets. Our comprehensive evaluation of 9 embedding models and classical metrics reveals significant performance gaps, with no single approach excelling across all dimensions. For instance, while state-of-the-art embedding models like OpenAI's `text-embedding-3-large` dominate in aligning with human preferences (0.682 vs. 0.591 for the best classical metric), they are significantly outperformed by classical metrics on information sensitivity tasks, where Jaccard Similarity achieves a score of 0.905 compared to the top embedding score of 0.794. SAGE further uncovers critical trade-offs: OpenAI's `text-embedding-3-small` achieves the highest clustering performance (0.483) but demonstrates extreme brittleness with the lowest robustness score (0.011). SAGE exposes critical limitations in current semantic understanding capabilities and provides a more realistic assessment of model robustness for real-world deployment.[1]

1 Introduction

The rapid advancement of large language models has been accompanied by increasingly sophisticated benchmarks [4], yet current evaluation frameworks often fail to capture the nuanced, multifaceted nature of semantic understanding required for real-world applications. While benchmarks like MTEB [26] and BEIR [37] provide valuable model rankings, they primarily assess performance under ideal conditions and focus narrowly on retrieval tasks, missing critical aspects of semantic robustness and human alignment. This gap becomes particularly problematic as AI systems are deployed in noisy, adversarial environments where robustness [17] and human alignment [43] are paramount [14].

[1] Code available: `https://github.com/sgoel97/neurips-2025-sage`

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Evaluating the Evolving LLM Lifecycle: Benchmarks, Emergent Abilities, and Scaling.

To address these limitations, we introduce SAGE (Semantic Alignment & Generalization Evaluation), a benchmark designed around two core principles: Semantic Alignment (accuracy in reflecting human judgments and preferences) and Generalization (robustness under diverse and adversarial conditions). SAGE's unique contribution lies in its comprehensive evaluation of semantic understanding through deliberately challenging scenarios that expose model limitations invisible to traditional benchmarks.

2 The SAGE Benchmark: A Holistic Evaluation Protocol

SAGE is designed around two core principles required for deep semantic understanding:

1. **Semantic Alignment:** The ability to accurately reflect human judgments, preferences, and the nuanced understanding of meaning.
2. **Generalization:** Robustness and reliability when faced with diverse and challenging conditions, such as noisy data or adversarial perturbations.

To measure these properties, SAGE aggregates performance across five distinct task categories, chosen to be more challenging and comprehensive than those in existing benchmarks.

2.1 Task 1: Human Preference Alignment

Semantic similarity metrics must reflect nuanced human judgments rather than surface-level patterns to be useful in real-world applications. This task ensures metrics align with how humans actually perceive and evaluate text quality and relevance.

We evaluate this alignment using OpenAI's human feedback dataset from the "summarize_from_feedback" collection [35]. The dataset contains two types of human judgments: (1) multi-dimensional ratings of summary quality in the axis_evals table, and (2) pairwise preferences between summaries in the comparisons table.

For multi-dimensional ratings, we measure how well a metric's similarity scores correlate with human quality assessments by computing the Pearson correlation between each summary-source similarity score and its corresponding human ratings. For pairwise preferences, we test whether the metric correctly predicts human choices - selecting the summary with higher similarity to the source as the preferred one - and measure prediction accuracy against ground truth preferences. Further details are provided in Appendix A.2, and complete subtask scores are provided in Table 2 and Table 3.

2.2 Task 2: Transformation Robustness

Real-world text contains noise from OCR errors, typos, and formatting inconsistencies [3, 29]. A robust similarity metric should distinguish between superficial changes that preserve meaning and semantic alterations that fundamentally change content.

We evaluate this capability using three long-form text datasets: academic papers [6], legislation [10], and news articles [33]. For each document and its summary, we apply six transformations: three that preserve meaning (superficial perturbations like typos or synonym replacements) and three that alter meaning (semantic changes like negation or factual modifications).

We then measure three similarity relationships: original-to-superficial (after surface-level changes), original-to-semantic (after semantic changes), and original-to-summary (baseline summary similarity). A robust metric should consistently rank these relationships as: superficial perturbations maintain highest similarity, summaries have intermediate similarity, and semantic alterations show lowest similarity. We report the percentage of datapoints where all three relationships hold simultaneously. Implementation details and full task scores are in Appendix A.3 and Table 4.

2.3 Task 3: Information Sensitivity

Similarity metrics should accurately detect and quantify semantic degradation [5, 12]. Unlike conventional robustness evaluations that test resilience to noise [16], this task measures whether metrics can precisely track how meaning changes as content is modified.

We test six long-form datasets by applying two perturbations: (1) inserting irrelevant content ("needle-in-haystack"), and (2) removing content spans. An information-sensitive metric should show similarity scores that decrease proportionally with the amount of perturbation - more inserted noise or removed content should yield correspondingly lower similarity scores.

We score each metric based on how closely its similarity changes follow this expected relationship: perfectly linear degradation receives the highest score, while erratic or flat responses score poorly. Detailed methodology and performance scores are in Appendix A.4 and both Table 5 and Table 6.

2.4 Task 4: Clustering Performance

Effective similarity metrics should preserve meaningful categorical structure in unsupervised settings [40], making clustering a valuable proxy for semantic understanding. If a metric truly captures semantic relationships, documents with similar meanings should naturally cluster together.

We evaluate clustering quality across all 11 datasets from the Massive Text Embedding Benchmark (MTEB) [26]. Using agglomerative clustering with each model's similarity scores, we measure clustering quality via V-measure [31]. Complete details are provided in Appendix A.5.

2.5 Task 5: Retrieval Robustness

Real-world retrieval systems must handle documents with various text corruptions including character-level noise, semantic alterations, and content contamination [13]. While traditional benchmarks assume clean corpora, practical applications require robustness to these common perturbations [39].

We stress-test retrieval robustness using BEIR benchmark datasets [37]. For each dataset, we create an adversarially augmented corpus by generating 18 perturbed versions of each document using transformations from tasks 2 and 3. These include 6 robustness transformations, 6 needle insertions at varying positions and sizes, and 6 content removals at varying positions and sizes.

Performance is measured as the ratio of NDCG@10 scores between each perturbed corpus and the original clean corpus. We report the harmonic mean of these retention ratios across all perturbations. Full implementation details are in Appendix A.6.

3 Experimental Setup

We evaluated a suite of popular text embedding models and classical similarity metrics using the SAGE benchmark. Classical metrics included Levenshtein Ratio, ROUGE score [11], Jaccard similarity [8], and BM25 score. For all embedding models, we measured cosine similarity.

We used 5 SOTA embedding models based on various metrics such as industry adoption and performance on existing benchmarks. These include OpenAI's text-embedding-3-small and text-embedding-3-large, Cohere embed-v4.0, Voyage-3-large, and Gemini-embedding-001.

Scores for each task category are normalized to a 0-1 scale. The "Overall SAGE Score" is the unweighted average of the five category scores.

More information on each model and metric used can be found in appendix A.7.

4 Results: Uncovering Nuanced Performance Trade-offs

The results in Table 1 highlight SAGE's ability to reveal nuanced differences in performance that simpler benchmarks might miss. The central finding is that no single approach excels across all dimensions of semantic understanding.

Among embedding models, OpenAI's text-embedding-3-large achieves the highest overall SAGE score (0.524), followed by gemini-embedding-001 (0.504) and voyage-3-large (0.492). Notably, all five embedding models substantially outperform classical similarity metrics, with the best-performing classical approach (Jaccard Similarity at 0.423) trailing the lowest-performing embedding model (text-embedding-3-small at 0.474) by a significant margin.

Table 1: Performance of embedding models and classical similarity metrics on the SAGE benchmark across five evaluation categories. Scores are normalized to [0,1]. The overall score is the unweighted mean across categories.

Model / Metric	Clustering	Human Pref.	Robustness	Sensitivity	Retrieval	Overall
Embedding Models						
embed-v4.0	0.396	0.648	0.070	0.789	0.389	0.458
gemini-embedding-001	0.387	0.674	0.319	0.725	0.417	0.504
text-embedding-3-large	0.443	**0.682**	0.243	0.794	**0.457**	**0.524**
text-embedding-3-small	**0.483**	0.654	0.011	0.794	0.426	0.474
voyage-3-large	0.397	0.668	0.229	0.757	0.411	0.492
Classical Metrics						
BM25 Score	0.209	0.591	0.283	0.673	0.240	0.399
Jaccard Similarity	0.191	0.577	0.163	**0.905**	0.280	0.423
Levenshtein Ratio	0.140	0.532	**0.333**	0.857	0.160	0.404
ROUGE Score	0.190	0.568	0.178	0.875	0.200	0.402

Embedding models dominate in tasks requiring deep semantic understanding, such as human preference alignment (text-embedding-3-large: 0.682 vs. best classical BM25: 0.591), clustering (text-embedding-3-small: 0.483 vs. best classical BM25: 0.209), and retrieval (text-embedding-3-large: 0.457 vs. best classical Jaccard: 0.280). However, classical metrics demonstrate strong advantages in information sensitivity, where Jaccard Similarity achieves 0.905 compared to the top embedding score of 0.794, and in transformation robustness, where Levenshtein Ratio leads at 0.333 while embedding models top out at a score of 0.319. This trade-off reaches an extreme with text-embedding-3-small, which achieves the highest clustering performance (0.483) while simultaneously recording the lowest transformation robustness score across all approaches (0.011).

5 Discussion: The Benchmark-Production Readiness Gap

SAGE reveals a critical disconnect between benchmark performance and real-world readiness. While models achieve impressive scores on pristine datasets like MTEB and BEIR, our results show they fail under realistic conditions - text-embedding-3-small maintains only 1.1% robustness despite strong clustering (0.483), and even the best retrieval model retains just 45.7% effectiveness under adversarial noise. Real-world data is invariably corrupted through OCR errors, user typos, formatting inconsistencies, and transmission artifacts, yet current benchmarks evaluate only carefully curated inputs. This creates overconfidence: practitioners deploy models that excel on clean academic datasets but fail on the noisy production text that dominates real environments. Our finding that classical metrics outperform embeddings by 14% on information sensitivity tasks directly contradicts MTEB rankings, demonstrating that benchmark leadership doesn't necessarily translate to success in all settings.

The performance variation across tasks underscores that model selection must account for both application requirements and data characteristics. Embedding models excel at clustering (2.3× better) and human preference alignment (15.4% higher), yet classical metrics outperform them by 14% on information sensitivity while robustness scores for embeddings can plummet to 0.011 under perturbation. These task-specific trade-offs explain many production failures: teams select models based on aggregate scores without understanding their brittleness. The 67% failure rate of our most robust approach reveals that deploying embedding models without measures like domain-specific data cleaning, reranking, and filtering is premature for high-noise environments.

6 Conclusion and Future Work

In this work, we argued that a deeper form of semantic evaluation is needed to truly understand the capabilities of modern AI systems. We introduced the SAGE benchmark, a standardized protocol that assesses a wide range of technologies across a challenging and diverse set of tasks. Our results

show that SAGE can uncover critical performance trade-offs, demonstrating that the optimal choice of model or metric is highly dependent on the specific application.

These findings demand a fundamental shift toward benchmarks that mirror production complexity. Future evaluations must incorporate real-world corruptions beyond our tested perturbations, with greater data diversity, adversarial augmentation by default, and production constraints like latency and memory limitations. Until such "production-strength" benchmarks exist, practitioners should assume published scores represent upper bounds achievable only in laboratory conditions and deploy accordingly with defensive architectures, ensemble methods, and appropriate safeguards for critical applications.

We hope SAGE will serve as a valuable tool for researchers and practitioners, fostering a more rigorous and balanced approach to evaluation.

References

[1] abisee. Cnn/dailymail (dataset card). Hugging Face. URL https://huggingface.co/datasets/abisee/cnn_dailymail.

[2] BeIR. Beir/arguana (dataset card). Hugging Face. URL https://huggingface.co/datasets/BeIR/arguana.

[3] Y. Belinkov and Y. Bisk. Synthetic and natural noise both break neural machine translation, 2018. URL https://arxiv.org/abs/1711.02173.

[4] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, W. Ye, Y. Zhang, Y. Chang, P. S. Yu, Q. Yang, and X. Xie. A survey on evaluation of large language models, 2023. URL https://arxiv.org/abs/2307.03109.

[5] S. Goel, R. J. Lee, and K. Ramchandran. Quantifying positional biases in text embedding models, 2025. URL https://arxiv.org/abs/2412.15241.

[6] V. Gupta, P. Bharti, P. Nokhiz, and H. Karnick. SumPubMed: Summarization dataset of PubMed scientific articles. In J. Kabbara, H. Lin, A. Paullada, and J. Vamvas, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 292–303, Online, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-srw.30. URL https://aclanthology.org/2021.acl-srw.30/.

[7] V. Gupta, P. Bharti, P. Nokhiz, and H. Karnick. Sumpubmed: Summarization dataset of pubmed scientific articles. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 292–303, Online, 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.acl-srw.30.pdf.

[8] P. Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912. doi: 10.1111/j.1469-8137.1912.tb05611.x.

[9] A. Kornilova and V. Eidelman. Billsum: A corpus for automatic summarization of us legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, Hong Kong, China, 2019. Association for Computational Linguistics. URL https://aclanthology.org/D19-5406.pdf.

[10] A. Kornilova and V. Eidelman. BillSum: A corpus for automatic summarization of US legislation. In L. Wang, J. C. K. Cheung, G. Carenini, and F. Liu, editors, *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 48–56, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5406. URL https://aclanthology.org/D19-5406/.

[11] C.-Y. Lin and F. J. Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612, Barcelona, Spain, July 2004. doi: 10.3115/1218955.1219032. URL https://aclanthology.org/P04-1077/.

[12] X. Liu, H. Cheng, P. He, W. Chen, Y. Wang, H. Poon, and J. Gao. Adversarial training for large neural language models, 2020. URL https://arxiv.org/abs/2004.08994.

[13] Y.-A. Liu, R. Zhang, J. Guo, M. de Rijke, Y. Fan, and X. Cheng. Robust neural information retrieval: An adversarial and out-of-distribution perspective, 2024. URL https://arxiv.org/abs/2407.06992.

[14] J. Magomere, E. L. Malfa, M. Tonneau, A. Kazemi, and S. Hale. When claims evolve: Evaluating and enhancing the robustness of embedding models against misinformation edits, 2025. URL https://arxiv.org/abs/2503.03417.

[15] B. maintainers. Datasets available in beir (github wiki). GitHub Wiki. URL https://github.com/beir-cellar/beir/wiki/Datasets-available.

[16] M. Moradi and M. Samwald. Evaluating the robustness of neural language models to input perturbations, 2021. URL https://arxiv.org/abs/2108.12237.

[17] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp, 2020. URL https://arxiv.org/abs/2005.05909.

[18] MTEB. mteb/medrxiv-clustering-p2p (dataset card). Hugging Face, . URL https://huggingface.co/datasets/mteb/medrxiv-clustering-p2p.

[19] MTEB. mteb/medrxiv-clustering-s2s (dataset card). Hugging Face, . URL https://huggingface.co/datasets/mteb/medrxiv-clustering-s2s.

[20] MTEB. mteb/reddit-clustering-p2p (dataset card). Hugging Face, . URL https://huggingface.co/datasets/mteb/reddit-clustering-p2p.

[21] MTEB. mteb/reddit-clustering (dataset card). Hugging Face, . URL https://huggingface.co/datasets/mteb/reddit-clustering.

[22] MTEB. mteb/stackexchange-clustering-p2p (dataset card). Hugging Face, . URL https://huggingface.co/datasets/mteb/stackexchange-clustering-p2p.

[23] MTEB. mteb/stackexchange-clustering (dataset card). Hugging Face, . URL https://huggingface.co/datasets/mteb/stackexchange-clustering.

[24] MTEB. mteb/twentynewsgroups-clustering (dataset card). Hugging Face, . URL https://huggingface.co/datasets/mteb/twentynewsgroups-clustering.

[25] 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*, pages 2014–2037, Dubrovnik, Croatia, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.148. URL https://aclanthology.org/2023.eacl-main.148/.

[26] 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*, pages 2014–2037, Dubrovnik, Croatia, 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.eacl-main.148.

[27] OpenAI. openai/summarize-from-feedback. GitHub repository, 2020. URL https://github.com/openai/summarize-from-feedback.

[28] OpenAI. Summarize from feedback (dataset card). Hugging Face, 2022. URL https://huggingface.co/datasets/openai/summarize_from_feedback.

[29] D. Pruthi, B. Dhingra, and Z. C. Lipton. Combating adversarial misspellings with robust word recognition, 2019. URL https://arxiv.org/abs/1905.11268.

[30] M. T. Ribeiro, T. Wu, C. Guestrin, and S. Singh. Beyond accuracy: Behavioral testing of NLP models with CheckList. In D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.442. URL https://aclanthology.org/2020.acl-main.442/.

[31] A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In J. Eisner, editor, *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/D07-1043.

[32] scikit-learn developers. The 20 newsgroups text dataset (documentation). scikit-learn User Guide. URL https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html.

[33] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks, 2017. URL https://arxiv.org/abs/1704.04368.

[34] sgoel9. Paul graham essays (dataset). Hugging Face. URL https://huggingface.co/datasets/sgoel9/paul_graham_essays.

[35] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. Christiano. Learning to summarize from human feedback, 2022. URL https://arxiv.org/abs/2009.01325.

[36] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *NeurIPS 2021 Datasets and Benchmarks Track (Round 2)*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/file/65b9eea6e1cc6bb9f0cd2a47751a186f-Paper-round2.pdf.

[37] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=wCu6T5xFjeJ.

[38] H. Wachsmuth, S. Syed, and B. Stein. Retrieval of the best counterargument without prior topic knowledge. In I. Gurevych and Y. Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 241–251, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1023. URL https://aclanthology.org/P18-1023.

[39] C. Wu, R. Zhang, J. Guo, M. de Rijke, Y. Fan, and X. Cheng. Prada: Practical black-box adversarial attacks against neural ranking models, 2022. URL https://arxiv.org/abs/2204.01321.

[40] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005. doi: 10.1109/TNN.2005.845141.

[41] Y. Yoo, C. Jeong, S. Gim, J. Lee, Z. Schimke, and D. Seo. A novel patent similarity measurement methodology: Semantic distance and technological distance, 2023.

[42] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28 (NeurIPS 2015)*, 2015. URL https://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classification. See also PDF at NeurIPS Proceedings: https://proceedings.neurips.cc/paper/5782-character-level-convolutional-networks-for-text-clasclas-sification.pdf.

[43] K. Zhou, K. Ethayarajh, D. Card, and D. Jurafsky. Problems with cosine as a measure of embedding similarity for high frequency words. In S. Muresan, P. Nakov, and A. Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 401–423, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.45. URL `https://aclanthology.org/2022.acl-short.45/`.

A Technical Appendices and Supplementary Material

A.1 Dataset Details

A.1.1 OpenAI Summarize from Feedback Dataset

Source and Citation: The Summarize from Feedback dataset was introduced by Stiennon et al. [35] as part of their work on learning to summarize from human feedback. The dataset is publicly available through OpenAI and Hugging Face. The dataset includes machine-generated summaries for Reddit TL;DR posts, CNN articles, and Daily Mail articles. In total (across the comparisons and axis-eval parts) the Hugging Face release contains 193,841 rows. [28]

Structure and Size: The dataset comprises two primary components used in our evaluation:

- Axis Evaluations (`openai_summarize_scores`): Contains multi-dimensional human ratings of summary quality across various dimensions including overall quality, accuracy, coverage, and coherence. Summaries are rated on Likert scales. [28, 27]
- Pairwise Comparisons (`openai_summarize_comparisons`): Features 64,832 human preference judgments between summary pairs on the TL;DR dataset, where annotators indicate which of two summaries better represents the source text. [27]

A.1.2 Scientific Papers Dataset (PubMed)

Source and Citation: The scientific papers dataset utilizes biomedical abstracts from the PubMed database, specifically leveraging the SumPubMed corpus introduced by Gupta et al. [6]. In our implementation, this appears as both `scientific_papers_sensitivity` and `scientific_papers_robustness` variants in `datasets.py`.

Structure and Size: The dataset comprises 33,772 biomedical documents from the PubMed archive (BMC). Typical lengths in SumPubMed are: article *raw-text* version averages 4,227 words / 203 sentences; corresponding summary averages 277 words / 14 sentences. For the noun-phrase and hybrid processed versions, the article averages are 1,578 and 1,891 words (with 57 and 71 sentences) and the summaries average 223 words (10 sentences). [7]

A.1.3 CNN/DailyMail Dataset

Source and Citation: The CNN/DailyMail dataset was introduced by Hermann et al. and later refined by See et al. [33]. It is available through the `abisee/cnn_dailymail` repository in our `datasets.py` configuration.

Structure and Size: Contains over 300,000 unique news articles paired with human-written summaries:

- News articles from CNN (Apr 2007–Apr 2015) and Daily Mail (Jun 2010–Apr 2015). [1]
- Human-written summaries ("highlights") for each article. Version 3.0.0 has splits: 287,113 train, 13,368 validation, 11,490 test; mean token counts: 781 per article and 56 per highlight. [1]

A.1.4 BillSum Dataset

Source and Citation: The BillSum dataset was introduced by Kornilova and Eidelman [10] for summarization of U.S. Congressional and California state bills.

Structure and Size: Contains 22,218 U.S. Congressional bills (103rd–115th Congress; 1993–2018), split into 18,949 train and 3,269 test, plus an additional California test set of 1,237 bills (2015–2016). Average preprocessed lengths: U.S. bills 1,382 words / 46 sentences (median 1,253 / 42), California bills 1,684 words / 47 sentences (median 1,498 / 42). [9]

A.1.5 Paul Graham Essays Dataset

Source and Citation: The Paul Graham Essays dataset consists of essays written by Paul Graham, available through the `sgoel9/paul_graham_essays` repository in our configuration.

Structure and Size: The current Hugging Face release lists 215 rows (essays) in total. [34]

A.1.6 Amazon Polarity Dataset

Source and Citation: The Amazon Polarity dataset is derived from Amazon product reviews, commonly used for sentiment classification tasks. It represents a subset of larger Amazon review datasets focusing on binary sentiment classification.

Structure and Size: The original construction includes 3.6M training and 400k test reviews (two classes, balanced), using review *title* and *content* fields. [42]

A.1.7 ArguAna Dataset

Source and Citation: The ArguAna dataset was introduced by Wachsmuth et al. [38] and appears in the BEIR benchmark for argument retrieval.

Structure and Size: In the BEIR format, ArguAna contains 8,674 documents and 1,406 queries; each query has on average 1.0 relevant document (Rel D/Q = 1.0). [15, 2]

A.1.8 MTEB Clustering Datasets

The Massive Text Embedding Benchmark (MTEB) clustering task comprises 11 datasets spanning diverse domains. These datasets provide ground-truth categorical labels for text segments across multiple domains. Several clustering tasks are provided in both sentence-to-sentence (S2S) and paragraph-to-paragraph (P2P) variants (titles only vs. title+content/abstract). [25]

Our evaluation utilizes the following datasets from the MTEB collection:

Scientific Literature Clustering:

- ArXiv Clustering (P2P and S2S): Scientific abstracts/titles from ArXiv. (S2S compares titles; P2P typically concatenates title+abstract.) [25]
- BioRxiv Clustering (P2P and S2S): Biomedical preprints.
- MedRxiv Clustering (P2P and S2S): Medical preprints; the S2S/P2P datasets list 17,647 rows each in the Hugging Face release. [19, 18]

Community Discussion Clustering:

- StackExchange Clustering (Standard and P2P): Titles (standard) and title+body (P2P). The *standard* variant comprises 25 sets, each with 10–50 classes and 100–1000 sentences per class; the *P2P* variant provides 5 sets of 10k paragraphs and 5 sets of 5k paragraphs. [23, 22]
- Reddit Clustering (Standard and P2P): Titles (standard) and title+posts (P2P). The *standard* variant covers 199 subreddits across 25 sets (10–50 classes; 100–1000 sentences per class). The *P2P* variant comprises 10 sets of 50k and 40 sets of 10k paragraphs. [21, 20]

News and General Content Clustering:

- TwentyNewsgroups Clustering: Classical text classification dataset ($\sim$ 18k posts across 20 topics). The MTEB clustering task uses subject-only text for clustering. [32, 24]

A.1.9 BEIR Benchmark Datasets

Source and Citation: The BEIR (Benchmarking Information Retrieval) benchmark was introduced by Thakur et al. [37] as a heterogeneous benchmark for zero-shot evaluation of information retrieval models.

Structure and Size: BEIR originally comprises 18 datasets (spanning 9 IR task types) standardized into {corpus, queries, qrels} format. Example scale points from the official stats: MS MARCO corpus has 8.84M documents and 6,980 dev/test queries; FEVER uses $\sim$ 5.42M Wikipedia passages; TREC-COVID has 171k documents and 50 queries; ArguAna has 8.67k documents and 1,406 queries; CQADupStack has 457k documents and 13,145 queries. [15, 36]

A.2 Details for task 1: Human Preference Alignment

Design Rationale: Human preference alignment is central to evaluating semantic similarity. This task ensures that metrics reflect nuanced human judgments rather than surface-level similarity metrics, making them more suitable for real-world applications where human perception matters. By testing against both multi-dimensional quality ratings and pairwise preferences, we capture different aspects of how humans evaluate text similarity and quality.

Datasets: We utilize OpenAI's human feedback dataset from the "summarize_from_feedback" collection [35], containing machine-generated summaries for Reddit TL;DR posts, CNN articles, and Daily Mail articles (see Section A.1.1 for full details). The dataset contains 193,841 total rows across all components [28]. The axis_evals table provides human ratings on a 1-7 Likert scale across four dimensions—overall quality, accuracy, coverage, and coherence—for summary-text pairs across the TL;DR and CNN/DM evaluation sets. The comparisons table contains 64,832 human preference judgments between summary pairs on the TL;DR dataset [27], where annotators selected their preferred summary or indicated a tie.

Evaluation: We evaluate human preference alignment using two complementary approaches:

Pairwise comparison alignment (human_pref_comparisons): Using the OpenAI summarize comparisons dataset, we predict human preferences by selecting the summary with higher similarity to the source text. For each similarity metric, we assign the preference to summary 1 if it has higher similarity than summary 2, otherwise to summary 2. Performance is evaluated using four classification metrics:

- Accuracy: Overall correctness of preference predictions
- Precision: Proportion of correct predictions among all positive predictions
- Recall: Proportion of correctly identified positive preferences
- F1 Score: Harmonic mean of precision and recall

Multi-dimensional rating correlation (human_pref_scoring): Using the OpenAI summarize scores dataset, we compute Pearson correlation between similarity scores (summary vs. source text) and human ratings across four quality dimensions:

- Overall: General quality assessment
- Accuracy: Factual correctness of the summary
- Coverage: Completeness of information capture
- Coherence: Logical flow and readability

To normalize correlations to a [0, 1] scale suitable for comparison with other metrics, we apply the transformation:

$$\text{score} = 0.5 \times (\text{correlation} + 1)$$

This ensures all scores range from 0 (perfect negative correlation) to 1 (perfect positive correlation), with 0.5 representing no correlation.

Results The results for each of the two datasets used in this task are presented below.

Table 2: Human Preference (pairwise comparisons) subtask metrics on SAGE.

Model / Metric	Accuracy	Precision	Recall	F1
Embedding Models				
embed-v4.0	0.668	0.668	0.670	0.669
gemini-embedding-001	0.702	0.707	0.691	0.699
text-embedding-3-large	**0.714**	**0.721**	**0.702**	**0.711**
text-embedding-3-small	0.668	0.674	0.654	0.664
voyage-3-large	0.702	0.706	0.694	0.700
Classical Metrics				
BM25 Score	0.574	0.577	0.569	0.573
Jaccard Similarity	0.597	0.599	0.593	0.596
Levenshtein Ratio	0.611	0.614	0.603	0.608
ROUGE Score	0.580	0.581	0.582	0.582

Table 3: Human Preference (scoring) subtask metrics on SAGE.

Model / Metric	Overall	Accuracy	Coverage	Coherence
Embedding Models				
embed-v4.0	0.662	0.599	0.662	0.587
gemini-embedding-001	0.692	0.620	0.688	0.593
text-embedding-3-large	**0.694**	**0.629**	**0.691**	**0.596**
text-embedding-3-small	0.685	0.613	0.683	0.591
voyage-3-large	0.674	0.615	0.672	0.582
Classical Metrics				
BM25 Score	0.654	0.577	0.662	0.544
Jaccard Similarity	0.567	0.549	0.565	0.548
Levenshtein Ratio	0.420	0.475	0.415	0.513
ROUGE Score	0.562	0.547	0.558	0.551

A.3 Details for task 2: Transformation Robustness

Design Rationale: This evaluation framework specifically targets the brittleness observed in embedding models when confronted with character-level variations common in real-world text processing scenarios [3, 29]. Unlike conventional adversarial evaluations that primarily focus on semantic preservation [30], our transformation methodology maintains surface readability while testing whether models understand content semantically rather than relying on token-level patterns.

Datasets: We utilize three long-form text corpora with distinct linguistic characteristics: biomedical abstracts from the SumPubMed corpus (Section A.1.2) containing 33,772 documents with average raw-text length of 4,227 words [7], news articles from the CNN/DailyMail dataset (Section A.1.3) using the 11,490 test split articles with mean length of 781 tokens [1], and legislative documents from the BillSum corpus (Section A.1.4) using the 3,269 test split U.S. bills with average length of 1,382 words [9]. Together, these provide 48,531 document-summary pairs across formal academic, journalistic, and legal writing styles.

Evaluation: We apply six systematically designed transformations:

Superficial perturbations (preserve meaning):

- Random capitalization: 25% of characters randomly capitalized (not case-switched)
- Character deletion: Every 10th character removed (with special handling for spaces to preserve word boundaries)

- Numerization: Character substitutions - 'e' → '3', 'i' → '1', 'a' → '4', 'o' → '0'

Semantic alterations (change meaning):

- Negation toggling: Systematic reversal of affirmative/negative statements using regex patterns (e.g., "is" ↔ "is not", "can" ↔ "cannot")
- Sentence shuffling: Random permutation of all sentences in the document
- Word shuffling: Random permutation of all words within the entire text

For each document, we compute four similarity scores: original-to-original (baseline), original-to-superficial (averaged across three superficial perturbations), original-to-semantic (averaged across three semantic alterations), and original-to-summary. A robust metric should maintain the hierarchy: superficial similarity > summary similarity > semantic similarity. We evaluate three specific ordinal relationships:

- `summary_over_semantic`: Summary similarity exceeds all semantic alteration similarities
- `superficial_over_summary`: All superficial perturbation similarities exceed summary similarity
- `superficial_over_semantic`: All superficial perturbation similarities exceed all semantic alteration similarities

The final robustness score is the average percentage of test instances satisfying each of these three conditions.

Results The results for each model, metric, and dataset are presented below.

Table 4: Robustness subtask metrics on SAGE.

Model / Metric	BillSum	CNN/DailyMail	Scientific Papers
Embedding Models			
embed-v4.0	0.164	0.030	0.015
gemini-embedding-001	0.327	**0.333**	0.298
text-embedding-3-large	0.311	0.295	0.123
text-embedding-3-small	0.030	0.001	0.004
voyage-3-large	0.316	0.324	0.047
Classical Metrics			
BM25 Score	0.256	0.260	**0.333**
Jaccard Similarity	0.185	0.174	0.128
Levenshtein Ratio	**0.335**	**0.333**	0.331
ROUGE Score	0.225	0.145	0.163

A.4 Details for task 3: Information Sensitivity

Design Rationale: This evaluation task specifically measures semantic change detection - a critical capability for applications requiring fine-grained content monitoring such as document version-control systems, compliance tracking, and LLM watermarking [41]. Unlike conventional robustness evaluations that assess resilience to noise [16], we evaluate whether similarity metrics can monotonically detect semantic noise or degradation and accurately reflect this in their output.

Datasets: We use six diverse text domains with varying rhetorical structures: biomedical abstracts from PubMed (Section A.1.2) containing 33,772 documents averaging 4,227 words [7], technological essays from the Paul Graham corpus (Section A.1.5) with 215 essays [34], news articles from CNN/DailyMail test split (Section A.1.3) with 11,490 articles averaging 781 tokens [1], consumer reviews from Amazon Polarity dataset (Section A.1.6) using 400,000 test reviews [42], legislative documents from BillSum test split (Section A.1.4) with 3,269 bills averaging 1,382 words [9], and argumentative texts from the ArguAna corpus (Section A.1.7) containing 8,674 documents [2]. In total, we evaluate 457,420 documents across these domains.

Evaluation: We apply two controlled perturbation strategies:

Irrelevant content insertion ("needle-in-haystack"):

- Content source: Lorem Ipsum text of varying lengths from `https://www.lipsum.com/`
- Proportions: 15%, 50%, 100% of original document token length
- Positions: Beginning (position 0), middle (position 0.5), end (position 1.0) of document
- Implementation: Inserts needle text at exact character position calculated as `position * len(text)`

Token-based content removal:

- Removal levels: 15%, 50%, 90% of document tokens
- Selection strategy: Contiguous token removal starting from position-adjusted locations
- Position adjustment: Starting position is calculated as `position * (1 - removal_size)` to account for document shortening
- Implementation: Removes tokens at token-level using tokenizer encoding/decoding

We model expected similarity degradation using the theoretical relationship:

$$\text{similarity} = 1 - \frac{p}{1+p},$$

where p represents the perturbation proportion. This formula assumes diminishing marginal impact of additional perturbations, reflecting that initial changes have greater relative effect than subsequent ones.

Performance is computed as:

$$\text{sensitivity_score} = 1 - \text{MAE},$$

where MAE is the mean absolute error between observed and theoretical similarity values across all perturbation levels:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} \left|\text{observed}_i - \text{theoretical}_i\right|.$$

Scores range from 0 (completely insensitive) to 1 (perfectly calibrated sensitivity).

Results Results for each of the insertion and removal perturbations across models, metrics, and datasets are shown below.

Table 5: Sensitivity (insertion perturbation) subtask metrics on SAGE.

Model / Metric	Amazon Polarity	ArguAna	BillSum	CNN/DailyMail	Paul Graham	Scientific Papers
Embedding Models						
embed-v4.0	0.749	0.716	0.701	0.700	0.706	0.710
gemini-embedding-001	0.698	0.699	0.719	0.702	0.702	0.698
text-embedding-3-large	0.760	0.752	0.770	0.749	0.777	0.739
text-embedding-3-small	0.809	0.782	0.765	0.739	0.751	0.753
voyage-3-large	0.722	0.720	0.697	0.727	0.702	0.700
Classical Metrics						
BM25 Score	0.596	0.604	0.642	0.755	0.734	0.685
Jaccard Similarity	**0.974**	**0.972**	**0.941**	**0.964**	**0.963**	**0.947**
Levenshtein Ratio	0.883	0.854	0.838	0.842	0.844	0.851
ROUGE Score	0.888	0.884	0.881	0.881	0.881	0.882

Table 6: Sensitivity (removal perturbation) subtask metrics on SAGE.

Model / Metric	Amazon Polarity	ArguAna	BillSum	CNN/DailyMail	Paul Graham	Scientific Papers
Embedding Models						
embed-v4.0	**0.894**	**0.877**	0.844	0.846	0.864	**0.868**
gemini-embedding-001	0.765	0.750	0.738	0.740	0.747	0.738
text-embedding-3-large	0.879	0.856	0.810	0.812	0.806	0.821
text-embedding-3-small	0.878	0.852	0.783	0.802	0.794	0.821
voyage-3-large	0.856	0.807	0.812	0.806	0.763	0.765
Classical Metrics						
BM25 Score	0.591	0.595	0.639	0.779	0.763	0.693
Jaccard Similarity	0.829	0.835	**0.872**	0.849	0.864	0.852
Levenshtein Ratio	0.860	0.857	0.858	0.864	0.865	0.865
ROUGE Score	0.864	0.867	0.869	**0.868**	**0.868**	0.867

A.5 Details for task 4: Clustering Performance

Design Rationale: Clustering evaluation provides a comprehensive assessment of semantic representation quality by testing whether similarity metrics preserve meaningful categorical structure in an unsupervised setting [40]. V-measure provides a principled evaluation framework that harmonically combines cluster homogeneity (ensuring each cluster contains data points from a single class) and completeness (ensuring all data points from the same class are assigned to the same cluster) while maintaining invariance to cluster label permutations and symmetric properties under label exchange.

Datasets: We utilize all 11 clustering datasets from the Massive Text Embedding Benchmark (MTEB) [26] (see Section A.1.8 for complete details). These include: scientific literature clustering with ArXiv (S2S and P2P variants), BioRxiv (S2S and P2P), and MedRxiv (S2S and P2P, each with 17,647 rows) [19, 18]; community discussion clustering with StackExchange Standard (25 sets with 10-50 classes and 100-1000 sentences per class) and P2P (5 sets of 10k paragraphs, 5 sets of 5k paragraphs) [23, 22], Reddit Standard (199 subreddits across 25 sets) and P2P (10 sets of 50k, 40 sets of 10k paragraphs) [21, 20]; and news clustering with TwentyNewsgroups (approximately 18,000 posts across 20 topics) [24]. Documents range from 50 to 2,000 tokens with both balanced and imbalanced cluster distributions.

Evaluation: We employ agglomerative hierarchical clustering with the following specifications:

Clustering parameters:

- Linkage criterion: Complete linkage (maximum distance between clusters)
- Distance computation: Metric-specific implementations
  - Cosine: distance $= 1 - (\text{embeddings} \cdot \text{embeddings}^T)$
  - Jaccard: distance $= 1 - \frac{|\text{tokens}_i \cap \text{tokens}_j|}{|\text{tokens}_i \cup \text{tokens}_j|}$
  - ROUGE: distance $= 1 - F_1$, where $F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$
  - BM25: TF-IDF based distance with parameters $k_1 = 1.5$, $b = 0.75$, $\epsilon = 0.25$, normalized to [0,1]
  - Others: distance $= 1 -$ similarity
- Number of clusters: Set to the ground-truth number of categories for each dataset

V-measure computation:

$$V = 2 \cdot \frac{\text{homogeneity} \cdot \text{completeness}}{\text{homogeneity} + \text{completeness}}$$

where

$$\text{homogeneity} = 1 - \frac{H(C|K)}{H(C)} \quad \text{and} \quad \text{completeness} = 1 - \frac{H(K|C)}{H(K)},$$

with H representing entropy, C the cluster assignments, and K the ground-truth classes.

We report V-measure scores ranging from 0 (random clustering) to 1 (perfect clustering alignment).

A.6 Details for task 5: Retrieval Robustness

Design Rationale: Traditional retrieval evaluation assumes pristine textual conditions, yet real-world document corpora invariably contain OCR errors, typographical mistakes, formatting inconsistencies, and potentially malicious perturbations [13]. Our adversarial augmentation methodology comprehensively assesses retrieval robustness by evaluating similarity metrics' ability to maintain effectiveness when confronted with textual corruptions encountered in practical deployment environments [39].

Datasets: We utilize the complete BEIR benchmark [37] (see Section A.1.9), comprising 18 standardized retrieval datasets across 9 IR task types. Key datasets include: MS MARCO with 8.84M documents and 6,980 queries, FEVER with approximately 5.42M Wikipedia passages, TREC-COVID with 171k documents and 50 queries, ArguAna with 8,674 documents and 1,406 queries, and CQADupStack with 457k documents and 13,145 queries [15, 36]. Document lengths range from single sentences (20 tokens) to full articles (5,000+ tokens), providing comprehensive coverage of retrieval scenarios.

Evaluation: We create adversarially augmented corpora through systematic perturbation:

Augmentation process:

- For each original document, generate 18 perturbed versions using transformations from Tasks 2 and 3.
- This increases corpus size by a factor of 19 (original + 18 perturbations).
- Apply transformations with reproducible implementation.
- Perturbations include:
  - Character-level and semantic alterations (6 types): sentence shuffling, word shuffling, negation toggling, character pruning (every 10th), random capitalization (25%), and numerization
  - Needle insertion (6 variations): 3 positions (0, 0.5, 1.0) $\times$ 2 sizes (15%, 50%)
  - Content removal (6 variations): 3 positions (0, 0.5, 1.0) $\times$ 2 sizes (15%, 50%)

Performance measurement:

- Compute NDCG@10 (Normalized Discounted Cumulative Gain) for both original and augmented corpora.
- NDCG@10 calculation:

$$\text{DCG@k} = \sum_{i=1}^{k} \frac{\text{rel}_i}{\log_2(i+1)}$$

  where rel_i is the relevance score of the document at rank i.
- Retention ratio:

$$\text{Retention ratio} = \frac{\text{NDCG@10}_{\text{perturbed}}}{\text{NDCG@10}_{\text{original}}}$$

- Aggregate using the harmonic mean across all perturbation types:

$$H = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

  where x_i are the individual retention ratios.
- The harmonic mean is chosen to penalize poor performance on any single transformation more severely than the arithmetic mean.

Implementation details:

- Embeddings are generated using cosine similarity for retrieval scoring.
- Queries with no relevant documents in the corpus are excluded from evaluation.
- The final robustness score is the harmonic mean of all retention ratios.

A.7 Model and Metric Information

A.7.1 Classical Text Similarity Metrics

Levenshtein Ratio The implementation uses the `ratio` function from the `python-Levenshtein` library. For strings x, y with edit distance $d_L(x, y)$ and lengths $|x|, |y|$:

$$\text{LevRatio}(x, y) = \frac{2 \cdot \text{matches}}{|x| + |y|} = \frac{(|x| + |y|) - d_L(x, y)}{|x| + |y|}$$

where matches are the number of character matches in the optimal alignment.

Cost (per pair): time $O(|x||y|)$; memory $O(\min\{|x|, |y|\})$.

ROUGE Score The implementation uses `rouge_score` library with ROUGE-1 and ROUGE-2, returning the average of their F-measures:

$$\text{ROUGE}_{\text{impl}} = \frac{\text{ROUGE-1}_{\text{fmeasure}} + \text{ROUGE-2}_{\text{fmeasure}}}{2}$$

where ROUGE-1 measures unigram overlap and ROUGE-2 measures bigram overlap. The scorer uses `tiktoken` tokenizer (OpenAI's `text-embedding-3-small` tokenizer).

Cost: tokenization + n-gram counting in time $O(\text{total tokens})$; memory $O(V_n)$ for n-gram maps.

Jaccard Similarity For text strings, the implementation first tokenizes using `tiktoken`, then computes Jaccard similarity on token sets A, B:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \qquad \text{distance} = 1 - J(A, B)$$

Cost (per pair): $O(|A| + |B|)$ time after tokenization; memory $O(|A| + |B|)$.

BM25 Uses `BM25Plus` from `rank_bm25` library, a variant of BM25 with a delta parameter. For document D and query Q:

$$\text{score}(D, Q) = \sum_{t \in Q} \text{IDF}(t) \; \frac{tf_{t,D}(k_1 + 1)}{tf_{t,D} + k_1\left(1 - b + b\,\frac{|D|}{\text{avgDL}}\right)} + \delta$$

where δ is an additional parameter in BM25Plus. The implementation normalizes batch scores to $[0, 1]$ range.

Cost: similar to standard BM25; with inverted index, query-time proportional to postings traversed.

Cosine Similarity For embeddings $u, v \in \mathbb{R}^d$, implemented using PyTorch:

$$\cos(u, v) = \frac{u \cdot v}{\|u\|_2 \, \|v\|_2}$$

Cost (per pair): time $O(d)$; memory $O(d)$. With normalized embeddings where $\|u\|_2 = \|v\|_2 = 1$, cosine equals dot product.

A.7.2 Embedding Models (used with cosine similarity)

Basic runtime & storage notes. Embedding inference time scales roughly linearly with input tokens L; downstream similarity/retrieval scales with dimension d. Per-vector storage is $d \cdot b$ bytes (type size b: float32$= 4$, float16$= 2$, int8$= 1$).

Models evaluated

- **OpenAI** `text-embedding-3-small`: default d=1536 (typical options: 512 or 1536); max input $\sim$8192 tokens. *Cost:* pairwise cosine $O(d)$; storage $\propto d$.
- **OpenAI** `text-embedding-3-large`: default d=3072 (options: e.g., 256/1024/3072); max input $\sim$8192 tokens. *Cost:* higher d improves headroom at $\sim 2\times$ storage vs 1536-D.
- **Cohere** `embed-v4.0`: $d \in \{256, 512, 1024, 1536\}$ (default 1536); context up to 128k. Supports int8/uint8/binary outputs to reduce storage/I/O.
- **Voyage** `voyage-3-large`: default d=1024 (256/512/2048 options); context $\sim$32k. Offers compact dtypes (e.g., int8/binary).
- **Google** `gemini-embedding-001`: default d=3072 (typical 768/1536/3072 via `output_dimensionality`); input limit $\sim$2048 tokens.

A.8 NDCG Formula

The Normalized Discounted Cumulative Gain at rank 10 (NDCG@10) is a metric that quantifies the ranking quality by giving higher importance to relevant documents at the top of the search results. The formula is defined as:

$$\text{NDCG@10} = \frac{\text{DCG@10}}{\text{IDCG@10}} = \frac{\sum_{i=1}^{10} \frac{2^{\text{rel}_i}-1}{\log_2(i+1)}}{\sum_{i=1}^{10} \frac{2^{\text{rel}_i^*}-1}{\log_2(i+1)}},$$

where rel_i represents the relevance score of the item at position i, and rel_i^* denotes the relevance score in the ideal ranking.