GASE: Generatively Augmented Sentence Encoding

Anonymous ACL submission

Abstract

We propose a training-free approach to improve sentence embeddings leveraging test-time compute by applying generative text models for data augmentation at inference time. Unlike conventional data augmentation that utilises synthetic training data, our approach does not require access to model parameters or the computational resources typically required for fine-tuning state-of-the-art models. Generatively Augmented Sentence Encoding variates the input text by paraphrasing, summarising, or extracting keywords, followed by pooling the original and synthetic embeddings. Experimental results on the Massive Text Embedding Benchmark for Semantic Textual Similarity (STS) demonstrate performance improvements across a range of embedding models using different generative models for augmentation. We find that generative augmentation leads to larger performance improvements for embedding models with lower baseline performance. These findings suggest that integrating generative augmentation at inference time adds semantic diversity and can enhance the robustness and generalisability of sentence embeddings for embedding models. Our results show that performance gains depend on the embedding model and the dataset.

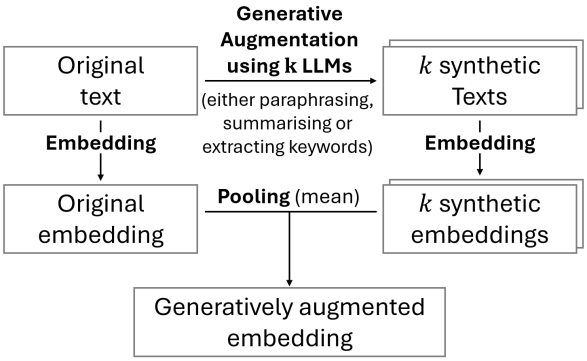

Figure 1: Approach for Generatively Augmented Sentence Encoding.

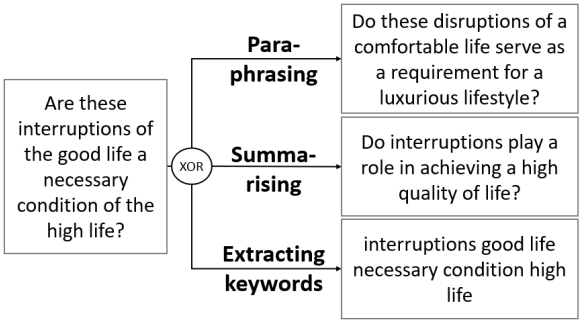

Figure 2: Augmentation examples for paraphrasing, summarising, and extracting keywords.

1 Introduction

Representation learning has emerged as a fundamental technique in natural language processing (NLP). However, the quality and robustness of the embeddings are highly dependent on the richness and diversity of the training data. Recent advancements of generative Large Language Models (LLMs) led to remarkable capabilities in generating human-like text. LLMs were also used for data augmentation by Dai et al. (2023), who applied paraphrasing techniques to train a BERT model (Devlin et al., 2019). Wahle et al. (2022) showed that LLM-generated paraphrases are harder to detect for humans than paraphrases generated with simple techniques like synonym replacement. The integration of generative models with representation learning has been explored by augmenting generative models, e.g., in Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Guu et al., 2020).

In contrast, we introduce an approach to augment embedding models that applies generative models using test-time compute: Generatively Augmented Sentence Encoding (GASE). Instead of generating synthetic training data, GASE creates textual variants of an input text through *paraphrasing*, *summarising*, or *extracting keywords* at inference time. A joint embedding is derived by pooling the embed-

dings of the original text and the generated transformation(s). The underlying hypothesis of our work is that adding textual diversity using generative models increases the ability of current embedding models to model semantics and hence benefits the performance of downstream STS tasks.

Also aiming to propose a training-free method, Lei et al. (2024) introduced Meta-Task Prompting, which employs use-case-specific prompts to generate multiple embeddings via a generative model, after which the embeddings are pooled. For retrieval systems, Gao et al. (2023) and Wang et al. (2023) described query expansion approaches using generative models that generate synthetic documents as a response to a retrieval request. Extending this work, GASE combines generative and embedding models without any assumptions on the type of models (unlike Meta-Task Prompting), nor is it limited to augmentation of retrieval queries such as Gao et al. (2023) and Wang et al. (2023). More recently, GenEOL (Thirukovalluru and Dhingra, 2025), a related approach, was introduced (see Table 7 for a comparison with GASE).

2 Method

GASE performs the following steps (see Figure 1):

Generative Augmentation. We apply k generative models to produce k variations of the original input sequence using exactly one of the following transformations: *Paraphrasing*, which provides a semantically equivalent but lexically or syntactically different variation of the input text. *Summarising*, which produces a shorter output text that captures the most important information of the input text. *Extracting Keywords*, which lists the most relevant words from a given text. Figure 2 provides an example for each. We evaluated $k \in \{0, 1, 2, 3\}$[1] using GPT-3.5 Turbo (OpenAI, 2024b), Reka-Flash (Reka, 2024), and GPT-4o mini (OpenAI, 2024a)[2].

2. Sentence Embedding. Generating embeddings for the original and k synthetic texts with one of 12 encoders.[3]

3. Pooling. The embeddings of the original text and the k synthetic texts are pooled by computing their arithmetic mean.[4]

We evaluate our approach on the English subtasks of the MTEB STS task[5] using cosine similarity and Spearman's rank correlation[6].

3 Results

Table 1 shows the results for different augmentation strategies vs. the respective baseline (see Table 9 in the Appendix for the results per dataset). All embedding models improved their avg. scores through augmentation with the greatest improvements for Llama-3.1 (+6.49pp), BERT (+6.11pp), and GloVe (+2.96pp). Paraphrase augmentation worked best for most models; however, GloVe improved most with keyword extraction, while BERT, MPNET, and Llama-3.1 excelled with summarisation.[7]

Summarising and extracting keywords significantly reduced the average word count across datasets (see Table 2). However, GPT-3.5 Turbo could not maintain STS22's text length when paraphrasing and at the same time only effectively summarised STS22. [8])

Evaluating the different generative models for paraphrase augmentation we find that GPT-3.5 Turbo performed best for all embedding models, except GloVe and Voyage-Large-2 (see Table 3 and for results per dataset Table 13 and Table 14 in the Appendix).

Using weighted averages between paraphrases and originals peaked at 25%/75% (excl. BERT) and was worst with paraphrases only (Figure 3).

$k > 1$ generative models yielded higher scores than a single model (see Table 3). BERT (+7.23pp), Lllama-3.1 (+6.49pp) and GloVe (+4.11pp) showed the largest gains. Using more than one generative models yielded higher scores than a single model (see Table 3). BERT (+7.23pp), Lllama-3.1 (+6.49pp) and GloVe (+4.11pp) showed the largest gains. Moreover, the overall performance of embedding models with lower non-augmented performance was associated with larger performance improvements through generative augmentation.

[1] Using a single generative model for $k = 2$ did not yield textually diverse and meaningful variations.

[2] For model versions and hyperparameters see Appendix F.

[3] See Appendix D for a list incl. references.

[4] Mean pooling dominates max pooling (see Table 8).

[5] See Appendix E for details on the datasets.

[6] See Reimers et al. (2016) for why Spearman's rank correlation is preferable over Pearson correlation for STS tasks.

[7] Additionally, we compared LLM-based keyword extraction with random keyword extraction (RKE) (see Appendix B for implementation details) and stopword removal showing that it generates semantically richer extractions (see Table 10).

[8] A word overlap analysis using the Jaccard Similarity (JS) can be found in Table 12 in the Appendix.

Embedding model	No augmentation		Paraphrasing		Summarising		Extracting keywords	
	Avg.	STS22	Avg.	STS22	Avg.	STS22	Avg.	STS22
glove.840B.300d	57.86	54.08	59.98	57.78	60.64	**61.21**	**60.82**	56.94
bert-large-cased	60.78	55.48	66.29	58.57	**66.89**	**63.47**	62.62	54.64
all-MiniLM-L6-v2	78.91	67.26	**80.32**	68.47	79.90	**69.27**	77.96	64.57
all-mpnet-base-v2	80.28	68.00	81.39	69.01	**81.48**	**70.12**	79.95	66.19
embed-english-light-v3.0	78.75	67.88	**80.30**	68.42	80.21	**68.97**	78.53	67.27
embed-english-v3.0	81.25	68.22	**82.10**	68.63	82.08	**70.29**	81.26	66.91
voyage-2	82.50	65.26	**83.38**	66.80	82.74	**68.50**	81.21	65.29
voyage-lite-02-instruct	85.95	78.62	**85.99**	79.08	85.43	**79.87**	83.91	76.60
voyage-large-2	83.60	63.97	**83.92**	64.69	83.66	**65.90**	81.98	63.68
voyage-large-2-instruct	84.61	66.59	**84.80**	67.72	84.54	**68.66**	83.40	65.73
mxbai-embed-large-v1	84.63	68.73	**85.01**	70.03	84.45	**70.36**	83.42	67.83
llama-3.1-8b	71.36	33.39	75.28	50.55	**76.74**	**59.44**	74.95	51.79

Table 1: Average scores across MTEB STS datasets and STS22 individually for all embedding models with and without generative augmentation using GPT-3.5 Turbo in %, bold: highest average scores for average and STS22 respectively).

Dataset	Original	Para-phrases	Summa-ries	Extracted keywords
STSB	10.1	11.5	10.7	5.7
STS12	11.1	11.6	11.1	6.1
STS13	9.0	10.5	11.1	5.4
STS14	9.3	11.0	11.4	5.7
STS15	10.6	12.0	11.5	5.7
STS16	11.6	12.6	12.8	5.7
STS17	8.7	9.6	9.0	4.5
STS22	477.2	216.5	58.4	103.8
SICK-R	9.6	10.0	9.3	4.7
BIOSSES	24.5	25.2	19.2	13.9
Average	58.2	33.1	16.4	16.1

Table 2: Mean word count (using GPT-3.5 Turbo for paraphrasing, summarising and extracting keywords).

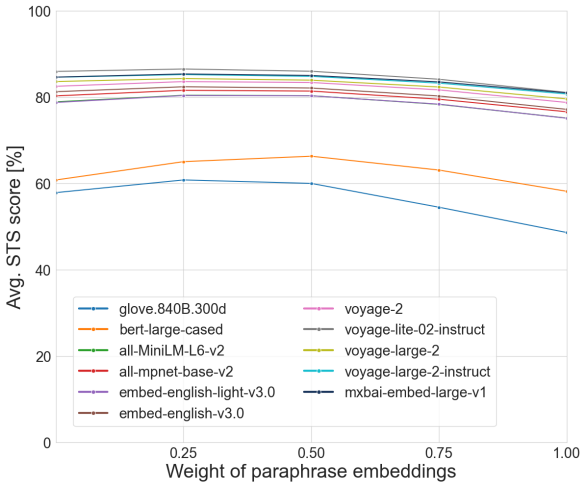

Figure 3: Scores for different weights for a weighted average between embeddings of paraphrases generated with GPT-3.5-Turbo and original texts ($k = 1$).

3.1 Extensions

To evaluate the generalisability of GASE we extended our work as follows:

Multilingual. Consistent with prior findings, GASE applied to 2 multilingual datasets with 3 multilingual models yields greater improvements for weaker encoders and summarisation remains most effective for BERT and STS22. (Table 4).[9]

Generative model ablation. Ablating the generative model size shows that GASE's performance correlates with generative model size (Table 5).[10]

Additional task. We evaluated GASE on the Pair Classification (PC) task (measured in average precision). In line with STS, improvements are negatively correlated with baseline performance and lower for STS22 (Table 6)[11].

[9]See Table 15 and Table 16 for results per dataset and Appendix D for model properties.

[10]See Appendix F for implementation details.

[11]See Table 17 for scores per dataset and (Muennighoff et al., 2023) for details for score calculations.

4 Discussion

We believe that embedding models with lower baseline performance benefited more from the semantic diversity induced by augmentation due to their inherent lower capability to model diverse semantics. Similarly, they gained more from the ensemble effect using with $k \geq 2$ generative models.

The effectiveness of keyword extraction for GloVe, BERT, and Llama-3.1 may be explained by the model's limited capability to handle complex semantics. Therefore, the reduction to keywords might help the model to reduce noise.

Longer texts likely caused summary augmentation to outperform paraphrase augmentation on

Embedding model	No Augmentation	gpt-3.5-turbo	reka-flash	gpt-4o-mini	gpt-3.5-turbo + reka-flash	gpt-3.5-turbo + reka-flash + gpt-4o-mini
glove.840B.300d	57.86	59.98	60.73	60.04	61.97	**62.18**
bert-large-cased	60.78	66.29	65.76	65.01	68.01	**68.41**
all-MiniLM-L6-v2	78.91	80.32	79.74	79.81	80.64	**80.76**
all-mpnet-base-v2	80.28	81.39	81.01	81.07	81.72	**81.85**
embed-english-light-v3.0	78.75	80.30	79.72	79.90	80.54	**80.66**
embed-english-v3.0	81.25	82.10	82.00	81.82	**82.53**	82.08
voyage-2	82.50	83.38	83.37	83.07	**83.80**	83.72
voyage-lite-02-instruct	85.95	85.99	85.92	85.97	**86.05**	86.01
voyage-large-2	83.60	83.92	83.98	83.95	84.12	**84.17**
voyage-large-2-instruct	84.61	84.80	84.68	84.78	84.86	**84.91**
mxbai-embed-large-v1	84.63	85.01	84.80	84.90	**85.17**	85.16
llama-3.1-8b	71.36	75.28	77.02	74.71	**77.85**	77.52

Table 3: Average STS scores without and with paraphrase augmentation using different generative models in % (bold: highest score per emb. model, underlined: highest score for $k <= 1$ per emb. model).

Embedding model	Augment.	STS17	STS22
bert-base-multilingual-cased	None	37.16	29.50
	Paraphrase	43.93	30.36
	Summarise	**47.13**	**42.03**
paraphrase-multilingual-mpnet-base-v2	None	82.51	61.56
	Paraphrase	**82.95**	62.72
	Summarise	82.88	**66.70**
multilingual-e5-large-instruct	None	83.39	69.64
	Paraphrase	**83.95**	66.48
	Summarise	83.60	**70.26**
voyage-multilingual-2	None	85.14	69.88
	Paraphrase	**85.73**	70.21
	Summarise	85.41	**71.92**

Table 4: Avg. STS scores for multi-lingual datasets with and without augmentation with GPT-4o mini in % (bold: highest scores per embedding model and dataset).

Generative Model Size (Qwen2.5)					
None	**0.5b**	**1.5b**	**3b**	**7b**	**14b**
80.28	79.2	79.79	80.26	80.33	80.79

Table 5: Avg. STS scores of all-mpnet-base-v2 without and with paraphrase augmentation using Qwen2.5 models with different number of parameters (in %).

STS22, as summarising significantly reduced the length. On other datasets, however, summaries were only slightly shorter, diminishing its augmentation value. Our results show pooling original and synthetic embeddings outperforms individual use (Figure 3). Across datasets, tasks, and languages, GASE particularly benefits embedding models with lower baseline performance and shorter texts, while summary augmentation is more effective for longer texts. Moreover, GASE requires a certain size of the generative model to be effective (Table 5).

Since GenEOL used different, mostly weaker, embedding models, our results cannot be directly compared with their results.

Embedding model	Augmentation	PC Average
glove.840B.300d	None	58.41
	Paraphrase	**64.64**
bert-large-cased	None	55.07
	Paraphrase	**62.36**
all-MiniLM-L6-v2	None	82.34
	Paraphrase	**83.80**
all-mpnet-base-v2	None	83.00
	Paraphrase	**84.61**
embed-english-light-v3.0	None	80.18
	Paraphrase	**82.88**
embed-english-v3.0	None	83.54
	Paraphrase	**85.12**
voyage-2	None	83.81
	Paraphrase	**85.24**
voyage-lite-02-instruct	None	**93.08**
	Paraphrase	92.22
voyage-large-2	None	**92.96**
	Paraphrase	92.36
voyage-large-2-instruct	None	85.21
	Paraphrase	**86.22**
mxbai-embed-large-v1	None	87.16
	Paraphrase	**87.38**
llama-3.1-8B	None	70.10
	Paraphrase	**72.78**

Table 6: Average Pair Classification (PC) scores with and without paraphrase augmentation in % (bold: highest scores per embedding model).

5 Conclusion

We propose GASE, which augments sentence encoders at inference with generative models for paraphrasing, summarising, or keyword extraction. On MTEB STS, GASE significantly improves lower-performing embedding models and incrementally enhances SOTA models. It applies broadly, needing only embedding and generative model outputs.

6 Limitations

This study presents several limitations:

1. Our approach significantly increases the computational cost for sentence encoding since each generative augmentation requires an additional inference by the embedding model plus inference using a generative LLM. For a basic estimate of the runtime increase based on STS17 see Table 18. The results highlight GASE's particular suitability for smaller embedding models, where it yields significant performance gains with minimal encoder runtime.

2. While deterministic embedding models such as GloVe yield consistent results, other models like GPT-3.5 Turbo examined in this research exhibit stochastic behaviour. Consequently, additional experiments are necessary to corroborate our findings, as LLMs can produce diverse outcomes, thereby limiting the conclusiveness of a single experimental run (Reimers and Gurevych, 2018). This caveat is particularly pertinent to observations based on marginal performance differentials.

3. Our experiments demonstrate that the effectiveness of GASE is limited to embedding models with lower baseline performance while encoders models with higher baseline performance exhibit minimal to no improvement. Further investigation is needed to understand the underlying reasons for this observation.

4. Our investigation was confined to evaluating GPT-3.5 Turbo, Reka-Flash, and GPT-4o mini for generative augmentation. Alternative LLMs, including the Claude and Gemini model families, may lead to different results.

5. Furthermore, our experimental design was restricted to $k \in \{0, 1, 2, 3\}$. While values of $k > 3$ were not explored, we posit that such configurations may be impractical for real-world applications due to prohibitive computational costs.

6. Our experiments for generative model ablation are limited to models with a parameter count $\in \{0.5, 1.5, 3, 7, 14\}$. Extending the analysis to larger generative models would yield insights into whether the observed pattern holds for models with more than 14b parameters.

7. Summaries typically comprise 50% or fewer words relative to its source text (Radev et al., 2002). However, due to the concise nature of the texts within the examined STS datasets the summaries generated in this work approximately maintain the original word count. (Except for STS22 which contains longer text sequences.) Hence, the efficacy of augmentation through summarisation may be more pronounced when applied to datasets comprising longer textual inputs (e.g., paragraphs) compared to the predominantly short text sequences examined within the scope of this study.

Acknowledgments

Code development for this work has been assisted by GPT-3.5 Turbo, GPT-4 Omni, Claude 3.5 Sonnet, Gemini Pro 1.5, and Github Copilot.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 Task 10: Multilingual Semantic Textual Similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 Task 1: Semantic Textual Similarity, Monolingual and Cross-Lingual Evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A pilot on semantic textual similarity. In **SEM 2012 -*

1st Joint Conference on Lexical and Computational Semantics, volume 2, pages 385–393.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic Textual Similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017a. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017b. SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Xi Chen, Ali Zeynali, Chico Camargo, Fabian Flöck, Devin Gaffney, Przemyslaw Grabowicz, Scott Hale, David Jurgens, and Mattia Samory. 2022. SemEval-2022 Task 8: Multilingual news article similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1094–1106, Seattle, United States. Association for Computational Linguistics.

Cohere. 2024. Documentation: Model Overview. https://docs.cohere.com/docs/cohere-embed (accessed 2024-08-08).

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Zihao Wu, Lin Zhao, Wei Liu, Ninghao Liu, Sheng Li, Dajiang Zhu, Hongmin Cai, Quanzheng Li, Dinggang Shen, Tianming Liu, and Xiang Li. 2023. ChatAug: Leveraging ChatGPT for Text Data Augmentation.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1(Mlm):4171–4186.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise Zero-Shot Dense Retrieval without Relevance Labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *ICML'20*, pages 3929–3938. JMLR.org.

{Hugging Face}. 2024. MTEB Leaderboard - a Hugging Face Space by mteb. /url{https://huggingface.co/spaces/mteb/leaderboard} (accessed 2024-09-06).

Yibin Lei, Di Wu, Tianyi Zhou, Tao Shen, Yu Cao, Chongyang Tao, and Andrew Yates. 2024. Meta-Task Prompting Elicits Embeddings from Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10141–10157, Bangkok, Thailand. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 9459–9474, Red Hook, NY, USA. Curran Associates Inc.

M. Marelli, Stefano Menini, Marco Baroni, L. Bentivogli, R. Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *International Conference on Language Resources and Evaluation*.

Meta. The Llama 3 Herd of Models | Research - AI at Meta. https://ai.meta.com/research/publications/the-llama-3-herd-of-models/.

Mixedbread. 2024. Documentation: Mxbai-embed-large-v1. \url{https://www.mixedbread.ai/embeddings/mxbai-embed-large-v1} (accessed 2024-08-08).

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. MTEB: Massive Text Embedding Benchmark. *Preprint*, arXiv:2210.07316.

OpenAI. 2024a. GPT-4o mini: Advancing cost-efficient intelligence. \url{https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/} (accessed 2024-07-24).

OpenAI. 2024b. OpenAI - GPT-3.5 Turbo. https://platform.openai.com/docs/models/gpt-3-5-turbo (accessed 2024-10-08).

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1532–1543. Association for Computational Linguistics (ACL).

Dragomir R. Radev, Eduard Hovy, and Kathleen McKeown. 2002. Introduction to the Special Issue on Summarization. *Computational Linguistics*, 28(4):399–408.

Nils Reimers, Philip Beyer, and Iryna Gurevych. 2016. Task-Oriented Intrinsic Evaluation of Semantic Textual Similarity. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 87–96, Osaka, Japan. The COLING 2016 Organizing Committee.

Nils Reimers and Iryna Gurevych. 2018. Why Comparing Single Performance Scores Does Not Allow to Draw Conclusions About Machine Learning Approaches. *preprint, arXiv:1803.09578*.

Nils Reimers and Iryna Gurevych. 2020. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 3982–3992.

Reka. 2024. Model Catalogue. https://www.reka.ai/ourmodels (accessed 2024-06-14).

Sentence Transformers. 2024. Pretrained Models — Sentence Transformers documentation.

Gizem Soğancıoğlu, Hakime Öztürk, and Arzucan Özgür. 2017. BIOSSES: A semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics (Oxford, England)*, 33(14):i49–i58.

Raghuveer Thirukovalluru and Bhuwan Dhingra. 2025. GenEOL: Harnessing the generative power of LLMs for training-free sentence embeddings. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2295–2308, Albuquerque, New Mexico. Association for Computational Linguistics.

Voyage AI. 2024. Voyage AI Embedding Models. https://docs.voyageai.com/docs/introduction (accessed 2024-07-20).

Jan Philip Wahle, Terry Ruas, Frederic Kirstein, and Bela Gipp. 2022. How Large Language Models are Transforming Machine-Paraphrased Plagiarism.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query Expansion with Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore. Association for Computational Linguistics.

Bowen Zhang, Kehua Chang, and Chunping Li. 2024. Simple Techniques for Enhancing Sentence Embeddings in Generative Language Models. *Preprint*, arXiv:2404.03921.

A Comparison of GASE and GenEOL

A comparison of our approach, GASE, and GenEOL can be found in Table 7. Based on 7 STS datasets, Thirukovalluru and Dhingra (2025) report a Spearman's rank correlation for the weakest baseline model of 47.06 and 80.37 for their best model using 32 generative augmentations. Using the same 7 datasets, we find that *without any augmentation* their best performing model is outperformed by 7 of the 12 embedding models evaluated in this work (e.g. by `all-mpnet-base-v2`; see Table 13 and Table 14).

	GASE (ours)	GenEOL
Transformations	Paraphrasing Summarising Extracting keywords	Paraphrasing Consice paraphrasing Changing the sentence structure Entailment Each one followed by summarisation.
Number of transformations	$k \in \{0, 1, 2, 3\}$	$m \in \{0, 2, 4, 8, 16, 24, 32\}$
Pooling	Mean pooling over variations generated *using k different generative LLMs*	Mean pooling over variations generated *using m different transformations*

Embeding models	English-language models:
	`glove.840B.300d`
	`bert-large-cased`
	`all-MiniLM-L6-v2`
	`all-mpnet-base-v2`
	`embed-english-light-v3.0`
	`embed-english-v3.0`
	`voyage-2`
	`voyage-lite-02-instruct`
	`voyage-large-2`
	`voyage-large-2-instruct`
	`mxbai-embed-large-v1`
	`llama-3.1-8B`
	
	Multilingual models:
	`paraphrase-multilingual-mpnet-base-v2`
	`multilingual-e5-large-instruct`
	`voyage-multilingual-2`
		`mistral0.1-7B`
		`llama-2-7B`
		`llama-3-8B`
		`bert-large`

Generative models	`gpt-3.5-turbo-0125`
	`reka-flash-20240226`
	`gpt-4o-mini-2024-07-18`
	`qwen2.5-0.5b`
	`qwen2.5-1.5b`
	`qwen2.5-3b`
	`qwen2.5-7b`
	`qwen2.5-14b`
	`qwen2.5-32b`
		`gpt-3.5-turbo-0125`
		`mistral0.1-I-7B`

STS datasets	Complete English MTEB for STS: STSB STS12 STS13 STS14 STS15 STS16 STS17 STS22 Sick-R BIOSSES Multi-lingual datasets: STS17 STS22	Subset of English MTEB for STS: STSB STS12 STS13 STS14 STS15 STS16 Sick-R
Other tasks	TextPairClassification Summarisation	TextPairClassification Classification Clustering Reranking

Table 7: Comparison of GASE and GenEOL.

B Random keyword extraction

Random keyword extraction randomly selects 16.1/58.2 of the words from a text sequence. This share is equal to the average share of LLM-extracted keywords and the total word count of a sentence (see Table 2). As this results in a low number of keywords for short sentences, the minimal number of randomly selected keywords was set to 3. All punctuation has been removed before extracting the keywords.

C Prompts

Paraphrasing with GPT-3.5 Turbo and GPT-4o mini :
"Rephrase the following text while maintaining its original meaning. If the text contains only a single word, provide a definition or a synomym. When done, check and make sure that the length of the original is approximately maintained. Text:"

Paraphrasing with Reka-Flash:
"Rephrase the following text while maintaining its original meaning. Do not provide multiple alternatives. Before you reply, remove any explanations. Do only reply with the paraphrased text. If the text contains only a single word, provide a definition or a synomym. Text:"

Summarising with GPT-3.5 Turbo:
"summarise the following text. Do not include any meta text and only output the summary. If the text is too short to summarise, paraphrase it instead. Text:"

Extracting keywords with GPT-3.5 Turbo:
"Extract the keywords from the following sentence. Do NOT inlcude any commas or fullstops in your response and do not start your answer with "keywords". Text: "

Model-specific post-processing is performed to remove meta-text not used for augmentation.

D Embeddings models

- `glove.840B.300d` (Pennington et al., 2014)
- `bert-large-cased` (Devlin et al., 2019)
- `all-MiniLM-L6-v2` (Sentence Transformers, 2024)
- `all-mpnet-base-v2` (Sentence Transformers, 2024)
- `embed-english-light-v3.0` (Cohere, 2024)
- `embed-english-v3.0` (Cohere, 2024)
- `voyage-2` (Voyage AI, 2024)
- `voyage-lite-02-instruct` (Voyage AI, 2024)
- `voyage-large-2` (Voyage AI, 2024)
- `voyage-large-2-instruct` (Voyage AI, 2024)
- `mxbai-embed-large-v1` (Mixedbread, 2024)
- `llama-3.1-8b` (Meta)
- `paraphrase-multilingual-mpnet-base-v2` (Reimers and Gurevych, 2020)
- `intfloat/multilingual-e5-large-instruct` (Wang et al., 2024)
- `voyage-multilingual-2` (Voyage AI, 2024)

For `llama-3.1-8b` we used the following KEEOL (Zhang et al., 2024) prompt:

> The essence of a sentence is often captured by its main subjects and actions, while descriptive terms provide additional but less central details. With this in mind, this sentence: "`input_text`" means in one word:

Following Zhang et al. (2024), we used the penultimate layer of `llama-3.1-8b` to extract the embeddings.

E Datasets

The following datasets from the MTEB STS leaderboard ({Hugging Face}, 2024) were used:

- STSBenchmark (Cer et al., 2017a)
- STS12 (Agirre et al., 2012)
- STS13 (Agirre et al., 2013)
- STS14 (Agirre et al., 2014)
- STS15 (Agirre et al., 2015)
- STS16 (Agirre et al., 2016)

- STS17 (Cer et al., 2017b)
- STS22 (Chen et al., 2022)
- SICK-R (Marelli et al., 2014)
- BIOSSES (Soğancıoğlu et al., 2017)

The original MTEB paper (Muennighoff et al., 2023) also included the STS11 dataset in Figure 1 of their paper where the authors specify the MTEB datasets, but excluded it in the evaluations. Similarly, the MTEB Leaderbord ({Hugging Face}, 2024) does not include STS11 either. To make our results comparable to other models evaluated on MTEB we therefore excluded STS11 from our evaluations.

F Generative Model Versions and Hyperparameters

The following versions have been used for generative augmentation:

- GPT-3.5 Turbo: `gpt-3.5-turbo-0125`
- Reka-Flash: `reka-flash-20240226`
- GPT-4o mini: `gpt-4o-mini-2024-07-18`
- Qwen2.5 (all models running via Ollama with quantization Q4_K_M):
  - `qwen2.5:0.5b`
  - `qwen2.5:1.5b`
  - `qwen2.5:3b`
  - `qwen2.5:7b`
  - `qwen2.5:14b`

GPT-3.5 Turbo, Reka-Flash, and GPT-4o mini have been accessed through the respective APIs with `temperature` set to 0 and `top_p` to 1 to make the experiments as reproducible as possible. Where applicable, a random seed of 1337 was used. Other hyperparameters were used with default values. The Qwen2.5 model were run with Ollama on a local computer.

G Additional Experimental Results

Embedding model	Pooling	Avg. Score
glove.840B.300d	mean	59.98
	max	59.02
bert-large-cased	mean	66.29
	max	65.16
all-MiniLM-L6-v2	mean	80.32
	max	79.88
all-mpnet-base-v2	mean	81.39
	max	81.03
embed-english-light-v3.0	mean	80.30
	max	79.74
embed-english-v3.0	mean	82.10
	max	81.62
voyage-2	mean	83.38
	max	82.69
voyage-lite-02-instruct	mean	85.99
	max	85.53
voyage-large-2	mean	83.92
	max	83.48
voyage-large-2-instruct	mean	84.80
	max	84.41
mxbai-embed-large-v1	mean	85.01
	max	84.69
llama-3.1-8b	mean	75.28
	max	75,07

Table 8: Average STS scores across tasks for different pooling methods.

Embedding model	Augmentation	Average	STSB	STS12	STS13	STS14	STS15	STS16	STS17	STS22	SICK-R	BIOSSES
glove.840B.300d	none	57.86	50.73	57.50	70.98	60.69	70.85	63.85	62.05	54.08	55.42	32.45
	paraphrasing	59.98	56.78	57.33	**71.17**	61.16	71.52	**67.66**	62.28	57.78	58.92	**35.21**
	summarising	60.64	57.67	55.00	70.24	**61.18**	71.64	66.81	67.08	**61.21**	**60.92**	34.65
	extracting keywords	**60.82**	**59.26**	**58.09**	71.12	60.34	**72.44**	65.31	**75.92**	56.94	55.34	33.50
bert-large-cased	none	60.78	59.18	45.71	63.94	54.42	68.98	65.46	71.30	55.48	64.16	59.14
	paraphrasing	66.29	67.97	**53.62**	68.56	59.73	74.01	**71.97**	74.14	58.57	69.79	**64.56**
	summarising	**66.89**	**68.23**	51.39	**70.58**	**61.98**	**74.27**	70.60	**74.48**	**63.47**	**70.74**	63.23
	extracting keywords	62.62	63.03	53.50	63.32	58.51	72.62	68.13	71.99	54.64	58.24	62.22
all-MiniLM-L6-v2	none	78.91	82.03	72.37	80.60	75.59	85.39	78.99	87.59	67.26	77.58	81.64
	paraphrasing	**80.32**	**82.33**	**73.33**	**83.33**	**77.85**	86.23	**79.88**	87.02	68.47	79.63	**85.07**
	summarising	79.90	82.17	72.14	82.89	77.24	**86.34**	78.99	**87.70**	**69.27**	**79.86**	82.41
	extracting keywords	77.96	81.99	73.29	81.14	75.77	85.87	78.70	87.45	64.57	70.92	79.88
all-mpnet-base-v2	none	80.28	83.42	72.63	83.48	78.00	85.66	80.03	**90.60**	68.00	80.59	80.43
	paraphrasing	81.39	**84.10**	74.04	85.52	**80.52**	**86.82**	**81.15**	89.08	69.01	81.25	82.39
	summarising	**81.48**	83.94	73.76	**85.54**	79.78	86.73	80.20	90.41	**70.12**	**81.32**	**82.96**
	extracting keywords	79.95	82.56	**74.82**	84.82	78.74	86.31	79.01	90.23	66.19	77.07	79.74
embed-english-light-v3.0	none	78.75	83.52	72.81	77.69	76.91	83.51	78.49	**88.49**	67.88	77.98	80.20
	paraphrasing	**80.30**	**84.65**	74.65	80.73	**79.41**	85.45	81.14	87.45	68.42	78.75	**82.36**
	summarising	80.21	84.08	73.45	**81.36**	78.41	**86.26**	**81.52**	88.19	**68.97**	**79.08**	80.79
	extracting keywords	78.53	82.81	**75.03**	80.17	78.29	84.14	76.38	88.14	67.27	74.45	78.64
embed-english-v3.0	none	81.25	86.54	74.76	81.68	78.81	87.18	83.01	**89.70**	68.22	77.52	85.05
	paraphrasing	**82.10**	**86.69**	**76.37**	83.71	**81.29**	87.96	**83.95**	88.47	68.63	**78.21**	**85.75**
	summarising	82.08	86.47	74.40	84.41	80.76	**88.49**	83.55	89.08	**70.29**	78.06	85.25
	extracting keywords	81.26	85.50	74.77	**84.66**	79.98	87.72	81.09	88.75	66.91	77.53	85.68
voyage-2	none	82.50	87.06	77.68	86.25	80.11	88.38	85.72	**89.78**	65.26	78.95	85.83
	paraphrasing	**83.38**	**87.17**	**79.23**	**87.13**	**81.67**	**88.57**	**86.27**	89.07	66.80	80.33	**87.56**
	summarising	82.74	86.51	76.81	86.25	80.89	88.46	84.88	89.10	**68.50**	**80.48**	85.55
	extracting keywords	81.21	85.71	77.76	86.17	79.63	87.49	84.37	88.66	65.29	72.06	85.03
voyage-lite-02-instruct	none	85.95	88.74	**86.19**	88.84	**86.84**	**89.84**	86.07	**87.19**	78.62	77.54	89.60
	paraphrasing	**85.99**	**88.81**	84.47	**89.05**	86.68	89.52	**86.53**	86.79	79.08	79.09	89.89
	summarising	85.43	87.93	82.77	88.06	85.05	88.98	85.49	86.70	**79.87**	**79.16**	**90.32**
	extracting keywords	83.91	86.75	83.82	87.97	84.47	88.98	84.55	86.28	76.60	70.77	88.88
voyage-large-2	none	83.60	**87.84**	78.66	86.98	82.25	**88.94**	85.71	**91.29**	63.97	79.83	**90.49**
	paraphrasing	**83.92**	87.62	**79.18**	**87.66**	**83.60**	88.91	**86.00**	90.22	64.69	81.00	90.29
	summarising	83.66	87.56	77.83	86.93	82.88	88.93	84.88	90.49	**65.90**	**81.09**	90.15
	extracting keywords	81.98	85.92	78.00	86.45	81.33	88.28	84.44	89.90	63.68	72.50	89.33
voyage-large-2-instruct	none	84.61	89.21	76.15	88.49	86.50	91.13	85.68	**90.05**	66.59	83.17	89.09
	paraphrasing	**84.80**	**89.31**	**77.61**	88.40	**86.62**	**91.17**	**86.20**	89.58	67.72	82.83	88.58
	summarising	84.54	88.84	76.27	88.29	85.56	91.06	85.24	89.40	**68.66**	**83.28**	88.78
	extracting keywords	83.40	87.38	76.63	**88.58**	85.55	90.51	84.03	88.57	65.73	76.58	**90.46**
mxbai-embed-large-v1	none	84.63	**89.29**	79.07	**89.80**	85.22	89.34	86.77	**89.21**	68.73	82.78	86.13
	paraphrasing	**85.01**	89.09	**80.83**	89.43	**85.88**	**89.46**	**86.94**	88.75	70.03	82.67	87.02
	summarising	84.45	88.60	78.51	89.07	84.82	89.09	85.96	88.43	**70.36**	**82.82**	86.82
	extracting keywords	83.42	87.43	78.64	88.75	83.82	88.44	84.84	87.40	67.83	78.17	**88.90**
llama-3.1-8b	none	71.36	79.79	62.81	79.08	69.72	78.49	80.88	84.12	33.39	76.27	69.04
	paraphrasing	75.28	**81.37**	67.03	81.29	72.88	80.91	**82.60**	83.04	50.55	78.25	74.84
	summarising	**76.74**	80.39	65.02	81.18	74.51	**82.07**	81.00	84.48	**59.44**	**79.36**	**79.92**
	extracting keywords	74.95	78.57	**67.39**	**82.68**	**74.74**	81.67	80.31	**85.71**	51.79	71.05	75.60

Table 9: Scores per dataset with and without generative augmentation using GPT-3.5 Turbo (Spearman's rank correlation in %, bold: highest scores).

Dataset	No Augmentation	Extracting Keywords (GPT-3.5 Turbo)	Random Keyword Extraction
STSB	50.73	**59.26**	23.95
STS12	57.50	**58.09**	34.78
STS13	70.98	**71.12**	22.25
STS14	**60.69**	60.34	28.48
STS15	70.85	**72.44**	33.86
STS16	63.85	**65.31**	24.99
STS17	62.05	**75.92**	36.47
STS22	54.08	**56.94**	38.44
SICK-R	**55.42**	55.34	40.04
BIOSSES	32.45	**33.50**	29.02
Average	57.86	**60.82**	31.23

Table 10: Scores per dataset for GloVe embeddings without generative augmentation vs. generative augmentation using LLM-extracted keywords with GPT-3.5 Turbo or random keyword extraction (Spearman's rank correlation in %, bold: highest scores).

Embedding model	Augmentation	Stopword removal	Average	STSB	STS12	STS13	STS14	STS15	STS16	STS17	STS22	SICK-R	BIOSSES
glove.840B.300d	none	no	57.86	50.73	57.50	70.98	60.69	70.85	63.85	62.05	54.08	55.42	32.45
		yes	60.07	57.13	57.43	70.91	60.68	70.86	63.84	75.83	54.66	54.97	34.39
	paraphrasing	no	59.98	56.78	57.33	71.17	61.16	71.52	67.66	62.28	57.78	58.92	35.21
		yes	65.43	67.05	59.98	72.72	63.75	73.70	69.98	75.01	61.01	61.94	49.17
bert-large-cased	none	no	60.78	59.18	45.71	63.94	54.42	68.98	65.46	71.30	55.48	64.16	59.14
		yes	53.08	50.11	41.37	53.10	50.11	63.68	59.66	63.73	49.69	48.75	50.54
	paraphrasing	no	66.29	67.97	53.62	68.56	59.73	74.01	71.97	74.14	58.57	69.79	64.56
		yes	60.25	61.42	50.45	58.43	55.20	69.75	67.19	67.98	57.01	58.25	56.82
all-MiniLM-L6-v2	none	no	78.91	82.03	72.37	80.60	75.59	85.39	78.99	87.59	67.26	77.58	81.64
		yes	75.36	78.80	68.29	77.62	73.05	84.35	74.92	87.05	65.74	65.58	78.24
	paraphrasing	no	80.32	82.33	73.33	83.33	77.85	86.23	79.88	87.02	68.47	79.63	85.07
		yes	77.78	80.69	71.61	81.15	75.92	84.86	77.51	85.89	66.36	70.61	83.21
all-mpnet-base-v2	none	no	80.28	83.42	72.63	83.48	78.00	85.66	80.03	90.60	68.00	80.59	80.43
		yes	75.51	78.32	65.71	79.49	73.99	83.98	73.28	88.98	65.82	67.01	78.47
	paraphrasing	no	81.39	84.10	74.04	85.52	80.52	86.82	81.15	89.08	69.01	81.25	82.39
		yes	78.20	80.87	70.76	83.09	77.53	85.00	76.72	87.05	66.28	71.99	82.70
embed-english-light-v3.0	none	no	78.75	83.52	72.81	77.69	76.91	83.51	78.49	88.49	67.88	77.98	80.20
		yes	74.36	78.16	68.36	76.23	73.84	81.67	71.69	88.18	66.87	64.32	74.34
	paraphrasing	no	80.30	84.65	74.65	80.73	79.41	85.45	81.14	87.45	68.42	78.75	82.36
		yes	77.58	81.43	73.12	79.15	76.81	83.77	76.89	87.55	67.81	70.40	78.87
embed-english-v3.0	none	no	81.25	86.54	74.76	81.68	78.81	87.18	83.01	89.70	68.22	77.52	85.05
		yes	76.70	79.52	67.15	80.84	75.37	83.76	74.40	88.51	66.99	66.56	83.95
	paraphrasing	no	82.10	86.69	76.37	83.71	81.29	87.96	83.95	88.47	68.63	78.21	85.75
		yes	79.26	83.00	72.79	83.09	78.72	85.21	79.03	87.36	68.03	71.45	83.96
voyage-2	none	no	82.50	87.06	77.68	86.25	80.11	88.38	85.72	89.78	65.26	78.95	85.83
		yes	78.11	81.52	71.71	82.73	77.23	85.59	79.12	87.91	66.40	67.22	81.64
	paraphrasing	no	83.38	87.17	79.23	87.13	81.67	88.57	86.27	89.07	66.80	80.33	87.56
		yes	80.42	84.16	76.32	84.89	79.29	86.33	82.93	87.58	67.55	71.42	83.69
voyage-lite-02-instruct	none	no	85.95	88.74	86.19	88.84	86.84	89.84	86.07	87.19	78.62	77.54	89.60
		yes	81.35	83.24	79.33	86.13	82.37	87.95	80.01	85.84	74.72	66.73	87.14
	paraphrasing	no	85.99	88.81	84.47	89.05	86.68	89.52	86.53	86.79	79.08	79.09	89.89
		yes	82.88	85.57	81.64	87.14	83.29	87.58	83.46	85.18	76.02	70.83	88.11
voyage-large-2	none	no	83.60	87.84	78.66	86.98	82.25	88.94	85.71	91.29	63.97	79.83	90.49
		yes	79.39	82.25	72.85	82.94	78.99	86.54	79.68	89.63	65.38	67.42	88.25
	paraphrasing	no	83.92	87.62	79.18	87.66	83.60	88.91	86.00	90.22	64.69	81.00	90.29
		yes	81.19	84.57	76.72	84.73	80.84	87.17	82.95	88.78	65.43	71.64	89.04
voyage-large-2-instruct	none	no	84.61	89.21	76.15	88.49	86.50	91.13	85.68	90.05	66.59	83.17	89.09
		yes	81.35	83.24	79.33	86.13	82.37	87.95	80.01	85.84	74.72	66.73	87.14
	paraphrasing	no	84.80	89.31	77.61	88.40	86.62	91.17	86.20	89.58	67.72	82.83	88.58
		yes	81.88	85.70	74.85	87.76	83.91	88.69	82.40	88.41	67.66	71.72	87.73
mxbai-embed-large-v1	none	no	84.63	89.29	79.07	89.80	85.22	89.34	86.77	89.21	68.73	82.78	86.13
		yes	79.54	83.54	72.88	85.73	81.33	86.72	78.69	86.36	67.41	68.76	83.99
	paraphrasing	no	85.01	89.09	80.83	89.43	85.88	89.46	86.94	88.75	70.03	82.67	87.02
		yes	81.36	86.15	78.11	87.20	83.03	86.93	78.69	86.40	68.47	72.70	85.97
llama-3.1-8b	none	no	71.36	79.79	62.81	79.08	69.72	78.49	80.88	84.12	33.39	76.27	69.04
		yes	69.50	73.48	59.66	77.37	69.66	77.17	72.94	83.00	50.94	62.15	68.58
	paraphrasing	no	75.28	81.37	67.03	81.29	72.88	80.91	82.60	83.04	50.55	78.25	74.84
		yes	74.57	78.29	66.98	80.90	73.61	79.63	78.76	83.86	61.88	68.92	72.90

Table 11: Scores per dataset with and without stopword removal using either no generative augmentation or paraphrase augmentation with GPT-3.5 Turbo (Spearman's rank correlation in %).

Dataset	Paraphrases	Summaries
STSBenchmark	0.39	0.52
STS12	0.39	0.49
STS13	0.33	0.41
STS14	0.37	0.47
STS15	0.42	0.53
STS16	0.37	0.40
STS17	0.41	0.59
STS22	0.39	0.21
SICK-R	0.46	0.67
BIOSSES	0.46	0.52
Average	**0.40**	**0.48**

Table 12: Jaccard Similarity between original texts and paraphrases and summaries generated with GPT-3.5 Turbo respectively.

Embedding model	Gen model	Average	STSB	STS12	STS13	STS14	STS15	STS16	STS17	STS22	SICK-R	BIOSSES
glove.840B.300d	none	57.86	50.73	57.50	70.98	60.69	70.85	63.85	62.05	54.08	55.42	32.45
	gpt-3.5-turbo	59.98	56.78	57.33	71.17	61.16	71.52	**67.66**	62.28	57.78	58.92	35.21
	reka-flash	60.73	55.02	59.53	71.09	61.22	72.81	65.20	64.27	60.11	61.65	36.43
	gpt-4o-mini	60.04	55.09	60.12	72.65	62.45	71.60	66.49	63.39	55.48	60.36	32.76
	gpt-3.5-turbo + reka-flash	61.97	59.46	60.06	72.01	62.68	72.46	67.62	**65.43**	61.46	61.51	**36.95**
	gpt-3.5-turbo + reka-flash + gpt-4o-mini	**62.18**	**59.99**	**61.79**	**72.88**	**64.04**	**72.03**	67.38	64.75	**60.76**	**61.79**	36.42
bert-large-cased	none	60.78	59.18	45.71	63.94	54.42	68.98	65.46	71.30	55.48	64.16	59.14
	gpt-3.5-turbo	66.29	67.97	53.62	68.56	59.73	74.01	71.97	74.14	58.57	69.80	64.56
	reka-flash	65.76	65.20	52.92	68.55	59.77	72.67	71.80	71.46	58.93	70.09	66.19
	gpt-4o-mini	65.01	65.37	54.27	68.60	58.18	72.26	71.12	73.54	55.72	70.50	60.54
	gpt-3.5-turbo + reka-flash	68.01	69.39	56.86	70.04	62.30	**74.61**	73.86	74.52	60.49	71.32	66.72
	gpt-3.5-turbo + reka-flash + gpt-4o-mini	**68.41**	**69.40**	**59.01**	**71.25**	**62.60**	74.13	**73.86**	**74.99**	**60.35**	**71.74**	**66.73**
all-MiniLM-L6-v2	none	78.91	82.03	72.37	80.60	75.59	85.39	78.99	87.59	67.26	77.58	81.64
	gpt-3.5-turbo	80.32	82.33	73.33	83.33	77.85	86.23	79.88	87.02	68.47	79.63	85.07
	reka-flash	79.74	81.30	73.52	82.42	76.83	85.72	79.02	87.01	68.83	79.66	83.11
	gpt-4o-mini	79.81	81.97	74.03	83.31	76.70	86.03	79.19	**87.69**	67.01	80.08	82.11
	gpt-3.5-turbo + reka-flash	80.64	82.30	74.59	83.72	78.11	86.20	**79.95**	87.23	**69.23**	80.12	84.97
	gpt-3.5-turbo + reka-flash + gpt-4o-mini	**80.76**	**82.31**	**75.16**	**84.29**	**78.22**	**86.18**	79.98	87.31	68.84	**80.34**	**85.02**
all-mpnet-base-v2	none	80.28	83.42	72.63	83.48	78.00	85.66	80.03	90.60	68.00	80.59	80.43
	gpt-3.5-turbo	81.39	84.10	74.04	85.52	80.52	86.82	81.15	89.08	69.01	81.25	82.39
	reka-flash	81.01	83.72	74.27	85.12	79.33	86.41	80.41	90.17	68.42	81.37	80.87
	gpt-4o-mini	81.07	83.75	74.19	85.19	79.28	86.45	80.47	**90.89**	67.81	81.23	81.43
	gpt-3.5-turbo + reka-flash	81.72	84.43	75.46	86.12	80.63	87.00	81.21	89.37	**69.33**	**81.55**	82.13
	gpt-3.5-turbo + reka-flash + gpt-4o-mini	**81.85**	**84.42**	**75.96**	**86.37**	**80.72**	**86.92**	**81.33**	89.68	69.05	81.50	**82.51**
embed-english-light-v3.0	none	78.75	83.52	72.81	77.69	76.91	83.51	78.49	88.49	67.88	77.98	80.20
	gpt-3.5-turbo	80.30	**84.65**	74.65	80.73	79.41	85.45	81.14	87.45	68.42	78.75	**82.36**
	reka-flash	79.72	83.51	75.74	78.60	78.38	84.63	80.09	87.98	69.06	78.76	80.42
	gpt-4o-mini	79.90	84.14	75.76	79.65	77.89	85.12	79.83	87.90	67.91	78.68	82.10
	gpt-3.5-turbo + reka-flash	80.54	84.48	76.45	80.61	**79.65**	85.54	**81.21**	87.60	**69.23**	**79.06**	81.55
	gpt-3.5-turbo + reka-flash + gpt-4o-mini	**80.66**	84.59	**77.10**	**81.00**	79.75	**85.68**	81.16	**87.70**	68.90	79.09	81.61
embed-english-v3.0	none	81.25	86.54	74.76	81.68	78.81	87.18	83.01	89.70	68.22	77.52	85.05
	gpt-3.5-turbo	82.10	86.69	76.37	83.71	81.29	87.96	83.95	88.47	68.63	78.21	85.75
	reka-flash	82.00	86.31	76.91	82.22	80.48	87.82	83.69	88.93	69.70	78.06	85.90
	gpt-4o-mini	81.82	86.52	76.80	82.57	80.11	87.77	83.72	**89.42**	68.36	77.90	85.05
	gpt-3.5-turbo + reka-flash	**82.53**	86.68	77.88	83.56	81.64	88.05	84.13	88.59	**69.87**	**78.44**	**86.48**
	gpt-3.5-turbo + reka-flash + gpt-4o-mini	82.08	**86.69**	**78.49**	**83.60**	**81.65**	**87.97**	**84.10**	88.76	69.67	78.35	86.36

Table 13: Scores per dataset with and without paraphrase augmentation using GPT-3.5 Turbo, Reka-Flash, and GPT-4o mini (Spearman's rank correlation in %, bold: highest scores) (part 1).

Embedding model	Gen model	Average	STSB	STS12	STS13	STS14	STS15	STS16	STS17	STS22	SICK-R	BIOSSES
voyage-2	none	82.50	87.06	77.68	86.25	80.11	88.38	85.72	**89.78**	65.26	78.95	85.83
	gpt-3.5-turbo	83.38	87.17	79.23	87.13	81.67	88.57	86.27	89.07	66.80	80.33	**87.56**
	reka-flash	83.37	86.93	79.65	87.83	81.79	88.48	86.18	88.98	67.42	80.67	85.72
	gpt-4o-mini	83.07	86.89	79.59	87.27	81.23	88.46	85.81	89.37	65.25	80.85	86.02
	gpt-3.5-turbo + reka-flash	**83.80**	**87.27**	80.62	87.99	82.40	88.51	**86.40**	88.77	**67.94**	80.87	87.28
	gpt-3.5-turbo + reka-flash + gpt-4o-mini	83.72	87.14	**81.06**	**88.14**	**82.50**	**88.40**	86.14	88.81	66.91	**81.00**	87.08
voyage-lite-02-instruct	none	85.95	88.74	86.19	88.84	86.84	**89.84**	86.07	**87.19**	78.62	77.54	89.60
	gpt-3.5-turbo	85.99	**88.81**	84.47	89.05	86.68	89.52	86.53	86.79	**79.08**	79.09	89.89
	reka-flash	85.92	88.24	84.88	88.79	86.58	89.46	86.28	86.74	78.76	78.97	90.53
	gpt-4o-mini	85.97	88.25	85.24	89.06	86.62	89.57	85.99	86.90	78.59	**79.45**	90.08
	gpt-3.5-turbo + reka-flash	**86.05**	88.74	84.83	89.14	86.69	89.36	**86.57**	86.72	78.81	79.34	90.34
	gpt-3.5-turbo + reka-flash + gpt-4o-mini	86.01	88.48	**84.95**	**89.26**	**86.74**	89.25	86.32	86.75	78.31	79.50	**90.55**
voyage-large-2	none	83.60	**87.84**	78.66	86.98	82.25	88.94	85.71	**91.29**	63.97	79.83	90.49
	gpt-3.5-turbo	83.92	87.62	79.18	87.66	83.60	88.91	86.00	90.22	64.69	81.00	90.29
	reka-flash	83.98	87.32	79.52	88.25	83.49	88.77	85.90	90.40	64.46	81.35	90.37
	gpt-4o-mini	83.95	87.54	80.02	87.91	83.22	**89.11**	85.68	90.47	63.99	81.54	90.08
	gpt-3.5-turbo + reka-flash	84.12	87.56	80.05	88.31	83.98	88.76	**85.97**	90.04	**64.84**	81.55	90.14
	gpt-3.5-turbo + reka-flash + gpt-4o-mini	**84.17**	87.49	**80.53**	**88.42**	**84.00**	88.73	85.77	89.82	64.69	**81.75**	**90.49**
voyage-large-2-instruct	none	84.61	89.21	76.15	88.49	86.50	91.13	85.68	**90.05**	66.59	83.17	**89.09**
	gpt-3.5-turbo	84.80	**89.31**	77.61	88.40	86.62	91.17	**86.20**	89.58	**67.72**	82.83	88.58
	reka-flash	84.68	88.81	78.41	88.30	86.40	90.95	85.42	89.71	67.04	**83.22**	88.52
	gpt-4o-mini	84.78	88.83	78.71	88.43	86.45	**91.17**	85.89	89.14	67.09	83.19	88.87
	gpt-3.5-turbo + reka-flash	84.86	89.16	79.11	88.56	**86.67**	90.97	85.86	89.67	67.56	83.00	88.08
	gpt-3.5-turbo + reka-flash + gpt-4o-mini	**84.91**	88.98	**79.79**	**88.59**	86.69	90.89	85.83	89.23	67.62	82.97	88.51
mxbai-embed-large-v1	none	84.63	**89.29**	79.07	89.80	85.22	89.34	86.77	**89.21**	68.73	82.78	86.13
	gpt-3.5-turbo	85.01	89.09	80.83	89.43	85.88	**89.46**	**86.94**	88.75	**70.03**	82.67	87.02
	reka-flash	84.80	88.93	81.04	89.36	85.51	89.27	86.48	89.04	68.67	82.66	87.07
	gpt-4o-mini	84.90	88.95	81.08	**89.76**	85.37	89.43	86.58	89.14	69.10	**82.73**	86.89
	gpt-3.5-turbo + reka-flash	85.17	89.16	81.96	89.37	85.96	89.39	86.79	88.97	69.51	82.72	87.81
	gpt-3.5-turbo + reka-flash + gpt-4o-mini	**85.16**	89.13	**82.24**	89.54	**85.98**	89.36	86.70	88.92	69.44	82.65	**87.63**
llama-3.1-8b	none	71.36	79.79	62.81	79.08	69.72	78.49	80.88	84.12	33.39	76.27	69.04
	gpt-3.5-turbo	75.28	81.37	67.03	81.29	72.88	80.91	82.60	83.04	50.55	78.25	74.84
	reka-flash	77.02	80.44	68.26	82.29	74.75	81.65	82.53	**85.77**	58.68	78.43	77.38
	gpt-4o-mini	74.71	80.38	68.15	82.09	73.39	81.20	81.87	84.60	41.34	**78.82**	75.23
	gpt-3.5-turbo + reka-flash	**77.85**	81.68	69.82	82.99	75.64	82.10	**83.27**	84.46	59.90	78.91	**79.68**
	gpt-3.5-turbo + reka-flash + gpt-4o-mini	77.52	**81.41**	**70.96**	**83.47**	**75.76**	**81.99**	82.84	84.14	**56.86**	78.99	78.77

Table 14: Scores per dataset with and without paraphrase augmentation using GPT-3.5 Turbo, Reka-Flash, and GPT-4o mini (Spearman's rank correlation in %, bold: highest scores) (part 2).

Dataset	paraphrase-multilingual-mpnet-base-v2		intfloat/multilingual-e5-large-instruct		voyage-multilingual-2	
	None	Paraph.	None	Paraph.	None	Paraph.
STS17_en-en	**86.99**	86.01	**87.64**	87.32	**90.81**	90.20
STS17_ar-ar	79.09	**81.04**	81.61	**83.36**	80.53	**81.58**
STS17_en-ar	80.85	**81.76**	78.66	**79.50**	81.59	**83.48**
STS17_en-de	83.28	**83.81**	**85.09**	85.05	88.09	**88.11**
STS17_en-tr	74.90	**75.44**	77.04	**78.88**	74.12	**77.09**
STS17_es-en	86.11	**86.25**	84.72	**85.35**	**88.24**	87.74
STS17_es-es	**85.14**	85.00	**88.09**	87.87	**88.71**	88.63
STS17_fr-en	81.17	**82.47**	83.26	**84.09**	87.10	**87.65**
STS17_it-en	**84.24**	84.20	84.94	**85.00**	88.98	**89.43**
STS17_ko-ko	83.29	**83.83**	81.58	**82.47**	80.17	**80.69**
STS17_nl-en	82.51	**82.65**	**84.65**	84.59	88.22	**88.38**

Table 15: Multilingual STS17 Scores with and without paraphrase augmentation using GPT-4o mini in % (bold: highest score per embedding model and dataset).

Dataset	paraphrase-multilingual-mpnet-base-v2		multilingual-e5-large-instruct		voyage-multilingual-2	
	None	Paraph	None	Paraph.	None	Paraph.
STS_ar-ar	**64.77**	56.36	**81.61**	65.02	59.65	**60.19**
STS_de-de	**62.30**	48.16	**78.66**	62.00	63.01	**63.08**
STS_de-en	**62.85**	57.03	**85.09**	61.72	**62.87**	62.56
STS_de-fr	**64.19**	62.68	**77.04**	65.52	**64.88**	64.12
STS_de-pl	**56.01**	41.26	**84.72**	56.97	54.14	**56.35**
STS_es-es	**73.27**	65.03	**88.09**	73.07	75.32	**75.87**
STS_es-en	**80.45**	74.22	**83.26**	79.99	82.43	**83.79**
STS_es-it	**78.71**	56.99	**84.94**	78.76	79.51	**79.78**
STS_fr-fr	**82.34**	76.56	81.58	**82.35**	**81.36**	81.35
STS_fr-pl	73.25	**84.52**	**84.65**	28.17	**73.25**	73.25
STS_it-it	**80.69**	62.82	64.77	**80.23**	79.92	**80.15**
STS_pl-pl	**46.83**	39.36	**62.30**	45.29	**46.30**	46.00
STS_pl-en	76.42	**76.90**	62.85	**75.61**	80.01	**80.81**
STS_ru-ru	**70.26**	63.33	64.19	**70.74**	68.35	**68.91**
STS_tr-tr	**69.56**	63.49	56.01	**70.16**	67.98	**68.34**
STS_zh-zh	**66.92**	66.89	**73.27**	68.53	72.22	**72.49**
STS_zh-en	69.56	**70.68**	**80.45**	66.06	**76.81**	76.53

Table 16: Multilingual STS22 Scores with and without paraphrase augmentation using GPT-4o mini in % (bold: highest score per embedding model and dataset).

Embedding model	Augmentation	Pair Classification		
		Twitter SemEval 2015	Twitter URL-Corpus	Sprint duplicate questions
glove.840B.300d	none	43.81	63.09	68.33
	paraphrase	**50.04**	**70.49**	**73.39**
bert-large-cased	none	47.27	69.81	48.15
	paraphrase	**57.21**	**76.28**	**53.61**
all-MiniLM-L6-v2	none	67.86	84.61	94.55
	paraphrase	**71.50**	**84.70**	**95.20**
all-mpnet-base-v2	none	73.85	84.99	90.15
	paraphrase	**77.34**	**85.08**	**91.42**
embed-english-light-v3.0	none	68.26	83.75	88.54
	paraphrase	**72.76**	**84.49**	**91.40**
embed-english-v3.0	none	73.43	84.98	92.21
	paraphrase	**76.58**	**85.45**	**93.33**
voyage-2	none	71.29	**86.25**	93.88
	paraphrase	**74.83**	86.19	**94.70**
voyage-lite-02-instruct	none	**91.38**	**89.14**	98.35
	paraphrase	90.31	88.37	**98.39**
voyage-large-2	none	75.10	86.57	93.96
	paraphrase	**76.82**	**86.58**	**95.25**
voyage-large-2-instruct	none	**92.87**	**90.10**	96.28
	paraphrase	90.78	89.09	**96.78**
mxbai-embed-large-v1	none	78.55	**86.12**	96.82
	paraphrase	**79.17**	86.03	**96.95**
llama-3.1-8b	none	69.70	82.73	57.87
	paraphrase	**73.00**	**84.26**	**61.09**

Table 17: Performance on Pair Classification with and without paraphrase generation using GPT-3.5-Turbo (in %).

H Runtime analysis

To provide a basic estimate of the additional runtime introduced by GASE over a non-augmented baseline, we performed five runs using GPT-3.5-Turbo for generation and all-mpnet-base-v2 for embeddings and report the mean and standard deviation. Because the difference in runtimes between the augmented and non-augmented setups was substantial and the standard deviation across runs was small, we did not carry out statistical significance testing (see Table 18).

All runtime experiments were conducted on a computer with the following specification: GeForce RTX 3090 MSI Gaming X Trio 24GB GPU, AMD Ryzen 9 9950X CPU, DDR5-6400 64GB RAM.

Augmentation	Average runtime [sec]	Stand. deviation of runtime [sec]
None	2.30	0.40
paraphrase	132.57	15.87
summarise	140.48	14.15
keywords	127.17	18.16

Table 18: Average runtime and corresponding standard deviation without and with generative augmentation using GPT-3.5-Turbo and, as an encoder, all-mpnet-base-v2 on STS17 (based on runs, in seconds).