Improving Sentence Embeddings with Automatic Generation of Training Data Using Few-shot Examples

Soma Sato **Hayato Tsukagoshi** **Ryohei Sasano** **Koichi Takeda**
Graduate School of Informatics, Nagoya University
{sato.soma.y7,tsukagoshi.hayato.r2}@s.mail.nagoya-u.ac.jp
{sasano,takedasu}@i.nagoya-u.ac.jp

Abstract

Decoder-based large language models (LLMs) have shown high performance on many tasks in natural language processing. This is also true for sentence embedding learning, where a decoder-based model, PromptEOL, has achieved the best performance on semantic textual similarity (STS) tasks. However, PromptEOL requires a manually annotated natural language inference (NLI) dataset for fine-tuning. We aim to improve sentence embeddings without using large manually annotated datasets by automatically generating an NLI dataset with an LLM and using it for fine-tuning of PromptEOL. To achieve this, we explore methods of data generation suitable for sentence embedding learning in this study. Specifically, we will focus on automatic dataset generation through few-shot learning and explore the appropriate methods to leverage few-shot examples. Experimental results on the STS tasks demonstrate that our approach outperforms existing models in settings without large manually annotated datasets.

1 Introduction

Sentence embeddings are widely studied as they can be used for many tasks such as text search, entailment recognition, and information extraction (Reimers and Gurevych, 2019; Tsukagoshi et al., 2021; Gao et al., 2021; Jiang et al., 2022; Raffel et al., 2022). Among these, methods based on decoder-based large language models (LLMs) have shown high performance in recent years. For example, SGPT (Muennighoff, 2022), which uses decoder-based LLMs to generate embeddings, and PromptEOL (Jiang et al., 2023), which generates sentence embeddings using a prompt-based method focusing on a single word, have been proposed. PromptEOL achieves the highest performance in STS in a setting using manually annotated data. However, when not using manually annotated NLI datasets, its performance is much lower.

Since the advent of high-performance decoder-based LLMs like GPT-4,[1] many efforts have been made to use data generated by decoder-based LLMs as a substitute for training data in various tasks, and their effectiveness has been reported (Meng et al., 2022; Ye et al., 2022a,b). Similarly, for sentence embedding learning, there are approaches such as GenSE (Chen et al., 2022), which automatically generates datasets using LLMs to augment sentence embedding datasets, and STS-Dino (Schick and Schütze, 2021), which is an automatically generated dataset for training sentence embedding models using LLMs. However, there has not been sufficient investigation on how to generate data using LLMs for sentence embedding learning. It is known that when generating datasets automatically via few-shot learning, the generated datasets are heavily dependent on the few-shot examples (Zhao et al., 2021), and if all the data is generated by using the same few-shot examples, the diversity of the generated datasets may be limited.

In this study, we explore how few-shot examples should be leveraged to automatically generate training data to obtain better sentence embeddings in a framework where NLI datasets generated by an LLM are used for fine-tuning of PromptEOL. Specifically, we examine how the quality of the final sentence embeddings varies when the number of few-shot examples used to generate training data is varied or when multiple sets of few-shot examples are used, and we reveal the optimal way to leverage few-shot examples. Our contributions are two-fold. First, we explored the optimal ways to leverage few-shot examples when using LLMs to generate NLI datasets for sentence embedding learning. Second, we achieved the highest score in the STS tasks in a setting without large manually annotated datasets.

[1] https://openai.com/index/gpt-4-research/

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pages 396–407

2 Related Work

This section introduces PromptRoBERTa (Jiang et al., 2022) and PrompEOL (Jiang et al., 2023), which successfully generate high-performance sentence embeddings by devising prompts.

PromptRoBERTa PromptRoBERTa introduces a new contrastive learning method to improve sentence embedding performance of RoBERTa. Specifically, it takes a sentence like "I have a dog." as input and transforms it using templates as follows: "`This sentence: "I have a dog." means [MASK].`" and "`The sentence: "I have a dog." means [MASK].`". By using the embeddings of the "`[MASK]`", it can represent the same sentence from diverse perspectives using different templates, resulting in reasonable positive pairs of sentence embeddings. By learning to bring these positive pairs of sentence embeddings closer together, PromptRoBERTa significantly reduces the performance gap between supervised and unsupervised settings, achieving better sentence embedding performance compared to traditional methods.

PromptEOL PromptEOL introduces a constraint called the "one-word limitation" and inputs the target sentence into LLMs along with a template. For example, to obtain the embedding of the sentence "I have a dog.", it inputs the prompt "`This sentence: "I have a dog." means in one word: ""`" into a decoder-based LLM. The hidden vector after "`in one word: ""`" is then used as the sentence embedding. Since the decoder-based LLM is pretrained on the next-token prediction task, it can obtain a sentence embedding that captures the meaning of the whole sentence by using the prompt to predict a word that paraphrases the entire sentence. Although PromptEOL demonstrates relatively high performance in an unsupervised setting, it can produce higher quality embeddings through supervised learning. PromptEOL achieved the best performance in the STS tasks by fine-tuning the LLM via contrastive learning on NLI datasets similar to supervised SimCSE (Gao et al., 2021).

3 Automatic NLI Dataset Generation

In this study, we explore how to automatically construct datasets for sentence embedding learning using LLMs. In this section, we explain the process of generating NLI datasets with LLMs.

3.1 Existing NLI Datasets

NLI datasets are widely used in various sentence embedding models, including SimCSE (Gao et al., 2021) and PromptEOL (Jiang et al., 2023). They contain sentence pairs comprising a premise and a hypothesis, which is labeled with either "entailment," "neutral," or "contradiction." The prominent NLI datasets include the Stanford NLI (SNLI) corpus (Bowman et al., 2015), the Multi-Genre NLI (MNLI) corpus (Williams et al., 2018), and the Cross-Lingual NLI (XNLI) corpus (Conneau et al., 2018), which contain approximately 579,000, 433,000, and 112,500 sentence pairs, respectively. Following Jiang et al. (2023), we use a dataset that combines the SNLI and MNLI corpora, and refer to it as the manual NLI dataset.

3.2 Automatic Generation Procedure

To automatically build NLI datasets, we generate hypothesis sentences from premise sentences. Specifically, we replace `[premise]` in each of the following prompts with a premise sentence and then feed the prompt to the LLM.

Prompt for entailment

```
Write one sentence that is logically entailed by
[premise] in the form of a statement beginning
with "Answer: ". Answer: "
```

Prompt for contradiction

```
Write one sentence that logically contradicts
[premise] in the form of a statement beginning
with "Answer: ". Answer: "
```

Next, we take the tokens generated between "`Answer: "`" and the next "`"`" as the generated hypothesis sentence.

We further improve the quality of the generated hypothesis sentences by applying few-shot learning (Brown et al., 2020). Specifically, we extract a few sentence pairs from the manual NLI dataset and add them as few-shot examples. The number of examples is around 20 at most, which is a feasible amount even if created manually from scratch.

4 Experiments

We first evaluate automatically generated NLI datasets using NLI classifiers. Next, we evaluate sentence embedding models fine-tuned with automatically generated NLI datasets. We explore how to use few-shot examples specifically for sentence

Dataset	Entailment	Contradiction
0-shot	0.348	0.901
1-shot	0.627	0.830
5-shot	0.883	0.941
20-shot	0.944	0.949
Manual NLI dataset	0.929	0.941

Table 1: The agreement ratio between the predicted and assigned labels of NLI datasets generated with zero-/few-shot learning and the manual NLI dataset

embedding learning. After conducting these experiments, we compare the best-performing method from our exploration with existing methods.

4.1 Evaluation of NLI Dataset

We evaluated the quality of the automatically generated NLI datasets using an NLI classifier. This allows us to assess the quality of NLI datasets generated by LLMs automatically.

Generation Method In our method, we generate hypotheses according to premises as input. Therefore, for the source premise sentences, we randomly extracted one million sentences from Wikipedia, following the unsupervised fine-tuning dataset of SimCSE (Gao et al., 2021). To reduce potential biases from the difference between the manual NLI datasets and sentences from Wikipedia, we used sentences with token counts between 4 and 32 to approximate the distribution of the manual NLI dataset. The frequency distribution of the token count is shown in Appendix A. We used LLaMA-2-7B-Chat (Touvron et al., 2023) as the LLM.

Evaluation Method We used DeBERTa (He et al., 2021) trained on the MNLI corpus.[2] For each sentence pair in the generated dataset, we performed a three-way classification of entailment, neutral, or contradiction. We then calculated the agreement ratio between the classification result and the assigned labels. For the manual NLI dataset and a zero-shot generated dataset, we randomly selected 3,000 sentence pairs for both entailment and contradiction, totaling 6,000 pairs, and calculated the agreement ratios for these pairs. For the few-shot generated datasets, to mitigate randomness due to the few-shot examples, we first created 10 sets of different examples for both entailment and contradiction. Then, each set was given 1,000 different premise sentences to create pairs, resulting in 20,000 pairs for evaluation.

[2] https://huggingface.co/microsoft/deberta-v2-xxlarge-mnli

Experimental Results Table 1 lists the agreement ratios for each dataset. The ratio improved as the number of few-shot examples increased. In the 5-shot setting, the ratio of contradiction is comparable to that of the manual NLI dataset, and in the 20-shot setting, the ratio for both entailment and contradiction reached levels comparable to those of the manual NLI dataset. These results suggest that the automatically generated NLI datasets with 5-shot or 20-shot learning were reasonably high quality. We provide examples of datasets obtained with 0-shot and 20-shot learning in Appendix B.

4.2 Explore How to Use Few-shot Examples

We evaluated sentence embedding models fine-tuned with the automatically generated NLI datasets using the STS tasks.[3] The STS task is to evaluate whether a model could correctly estimate the semantic similarity of sentence pairs. Specifically, we calculated the semantic similarity via the model and tested its closeness to a human evaluation. Following previous studies (Reimers and Gurevych, 2019; Gao et al., 2021; Jiang et al., 2023), the sentence embedding quality was evaluated in terms of Spearman's rank correlation coefficient between the cosine similarity of sentence embeddings and the human ratings.

Experimental Setup We fine-tuned LLaMA-2-7B (Touvron et al., 2023) with NLI datasets. Following Jiang et al. (2023), we used the same seven STS datasets for evaluation: STS 2012–2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS-B (Cer et al., 2017), and SICK-R (Marelli et al., 2014). To investigate the relationship between dataset size and performance, we trained our model with different numbers of examples. The number of examples in the datasets is $4{,}000\times2^n$ (n = 0, 1, ..., 6). For fine-tuning with PromptEOL, we used NLI datasets generated with 0-shot, 1-shot, 5-shot, 20-shot, 1-shot×5 (five combined 1-shot datasets), 5-shot×4 (four combined 5-shot datasets) setups and the manual NLI dataset. We used the same hyperparameters as PromptEOL (Jiang et al., 2023), with a batch size of 256 during training, 10% of the total steps for warm-up, and a learning rate of 5e-4. During training, we calculated Spearman's rank correlation coefficient on the STS-B develop-

[3] Evaluations were also conducted on downstream tasks of SentEval (Conneau and Kiela, 2018), but as reported in Jiang et al. (2023), the effectiveness of fine-tuning with the NLI dataset could not be confirmed. We provide the results of downstream tasks in Appendix C.

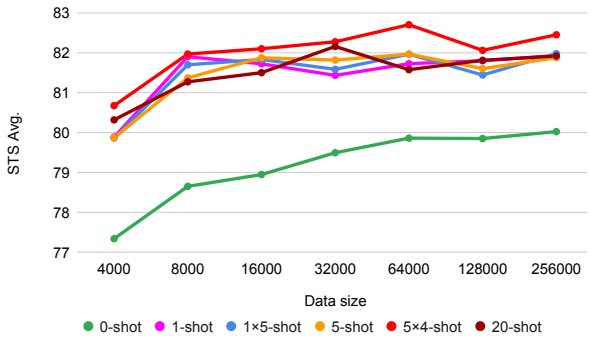

Figure 1: Performances of different few-shot settings

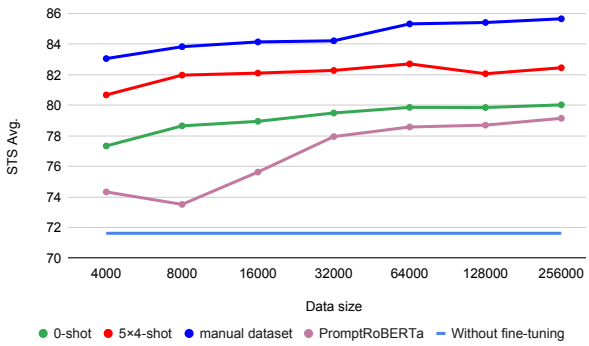

Figure 2: Performances of models fine-tuned with the automatically generated datasets and existing methods

ment set every (number of data / 4000) step and used the model with the highest score for the final evaluation. To minimize randomness from few-shot examples, we generated multiple NLI datasets: 10 for 1-shot, 5 for 1-shot×5 and 5-shot, 4 for 5-shot×4 and 20-shot, and 3 for zero-shot. We report their average scores for the final evaluation.

Experimental Results Figure 1 shows the results. Comparing the zero-shot and few-shot results, the few-shot performances outperformed the zero-shot performance regardless of the amount of data size, thus confirming the effectiveness of few-shot learning. Comparing 1-shot, 5-shot, and 20-shot, there there was no improvement in scores as the number of shots increased. This indicates that merely increasing the number of shots does not necessarily lead to better performance. Although there was little performance difference between 5-shot and 1-shot×5, 5-shot×4 consistently outperformed 20-shot, regardless of data size. According to Section 4.1, although the quality of the generated dataset with 1-shot learning is not sufficient, the generated dataset with 5-shot learning has sufficiently high quality. This suggests that distributing few-shot examples can improve performance, but only when the data quality exceeds a certain threshold. 5-shot×4 successfully introduces diversity while maintaining sufficient quality, and this balance between diversity and quality appears to be crucial for enhancing the effectiveness of sentence embeddings generated from NLI datasets.

4.3 Comparison with Existing Methods

To evaluate the performance of models trained on automatically generated NLI datasets, we compared the following five models: 1) PromptEOL without fine-tuning, 2) PromptEOL fine-tuned with the generated dataset using 0-shot learning, 3) PromptEOL fine-tuned with the generated dataset using 5-shot×4 learning, 4) PromptEOL fine-tuned with the manual NLI dataset, 5) Unsupervised PromptRoBERTa (Jiang et al., 2022), which achieved the highest performance without using manually annotated large-scale datasets. For unsupervised PromptRoBERTa, we used the premise sentences to automatically generate NLI datasets, which are used for training. For PromptRoBERTa and experiments using manually annotated datasets, we conducted experiments three times with different random seeds, and we reported their average scores as the final score. Other experimental settings and evaluation methods were the same as in Section 4.2.

Experimental Results Figure 2 shows the results. Overall, the models trained with automatically generated datasets consistently outperformed unsupervised methods. Specifically, the 5-shot×4 setting achieved the highest score of 82.71. Comparing the performance of PromptEOL without fine-tuning and PromptEOL fine-tuned with the automatically generated dataset using zero-shot learning, the fine-tuned model consistently outperformed. This indicates that fine-tuning with the generated NLI dataset is effective when no manually created examples are available. Moreover, our models outperformed PromptRoBERTa, indicating that our model achieved the best performance without using large manually annotated datasets.

Compared to the model fine-tuned with the manual dataset, the performance of the 5-shot×4 setting was 2–3 points lower. This indicates that there is still a gap between the 5-shot×4 dataset and the manual dataset, suggesting room for improvement. Despite this gap, there was an approximately 10-point performance improvement compared to the model without fine-tuning, confirming the effectiveness of the automatically generated dataset. We

provide the detailed experimental results in Appendix D.

5 Conclusion and Future Work

In this study, we explore optimal ways to leverage few-shot examples when using LLMs to generate NLI datasets for sentence embedding learning. Through experiments, we found that the performance could be enhanced by dividing the few-shot examples, as seen with the 5-shot×4 setting, since it improves dataset diversity. Furthermore, models trained with automatically generated NLI datasets outperformed existing unsupervised methods.

In future work, we will explore more sophisticated ways to generate a diverse and high-quality dataset. For example, instead of just dividing few-shot examples, a set of various overlapping few-shot examples could be generated and used in few-shot learning. It is also future work to apply our data generation procedure, which generates data by dividing few-shot examples, to data generation other than NLI datasets for sentence embedding learning.

Limitations

There are three major limitations in this study. Firstly, we only conducted experiments using LLaMA-2-7B as the LLM for both the automatic generation of the NLI dataset and the generation of sentence embeddings. It is known that the quality of generated sentences improves as the number of parameters in the LLM increases. In this study as well, it may be possible to obtain higher quality NLI datasets and sentence embedding models by using a model larger than LLaMA-2-7B. Since this method is expected to be applicable to many LLMs without depending on a specific LLM, to demonstrate the model-independent usefulness of our observations, we need to conduct experiments using various LLMs, such as the GPT series and OPT (Zhang et al., 2022).

Secondly, we followed the previous research, PromptEOL, and conducted evaluations using SentEval. However, it is not enough to comprehensively assess the quality of sentence embeddings to evaluate only with the STS tasks and SentEval. It is necessary to use various benchmarks, such as MTEB (Muennighoff et al., 2023), which evaluate sentence embeddings from multiple perspectives.

Thirdly, the experiments were conducted only in English. It is potentially applicable to many languages to generate datasets automatically because it does not require large, manually-annotated datasets, but our experiments were conducted only in English. To demonstrate the usefulness of our observation for multiple languages and improve cross-lingual/multi-lingual sentence embeddings, it is helpful to conduct experiments in languages other than English.

Acknowledgements

This work was partly supported by JSPS KAKENHI Grant Number 24H007271.

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.

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.

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.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, 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 2013), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference.

In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 632–642.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS 2020)*, pages 1877–1901.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. 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.

Yiming Chen, Yan Zhang, Bin Wang, Zuozhu Liu, and Haizhou Li. 2022. Generate, Discriminate and Contrast: A Semi-Supervised Sentence Representation Learning Framework. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*, page 8150–8161.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, page 1699–1704.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating Cross-lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 2475–2485.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, pages 6894–6910.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In *Proceedings of the Ninth International Conference on Learning Representations (ICRL 2021)*.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2023. Scaling Sentence Embeddings with Large Language Models. *arXiv:2307.16645*.

Ting Jiang, Jian Jiao, Shaohan Huang, Zihan Zhang, Deqing Wang, Fuzhen Zhuang, Furu Wei, Haizhen Huang, Denvy Deng, and Qi Zhang. 2022. PromptBERT: Improving BERT Sentence Embeddings with Prompts. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*, pages 8826–8837.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, pages 216–223.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating Training Data with Language Models: Towards Zero-Shot Language Understanding. In *Advances in Neural Information Processing Systems (NeurIPS 2022)*, pages 462–477.

Niklas Muennighoff. 2022. SGPT: GPT Sentence Embeddings for Semantic Search. *arXiv:2202.08904*.

Niklas Muennighoff, Nouamane Tazi, and Nils Reimers Loic Magne. 2023. MTEB: Massive Text Embedding Benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2023)*, page 2014–2037.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2022. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. In *Findings of the Association for Computational Linguistics (ACL 2022)*, pages 1864–1874.

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

Timo Schick and Hinrich Schütze. 2021. Generating Datasets with Pretrained Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, page 6943–6951.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Moly-

bog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv:2307.09288*.

Hayato Tsukagoshi, Ryohei Sasano, and Koichi Takeda. 2021. DefSent: Sentence Embeddings using Definition Sentences. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021) (Volume 2: Short Papers)*, pages 411–418.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2018)*, pages 1112–1122.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022a. ZeroGen: Efficient Zero-shot Learning via Dataset Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, page 11653–11669.

Jiacheng Ye, Jiahui Gao, Zhiyong Wu, Tao Yu Jiangtao Feng, and Lingpeng Kong. 2022b. ProGen: Progressive Zero-shot Dataset Generation via In-context Feedback. In *Findings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP 2022)*, page 3671–3683.

Susan Zhang, Stephen Roller, Naman Goya, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open Pre-trained Transformer Language Models. *arXiv:2205.01068*.

Tony Z. Zhao, Shi Feng Eric Wallace, Dan Klein, and Sameer Singh. 2021. Calibrate Before Use: Improving Few-Shot Performance of Language Models. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, page 12697–12706.

A Frequency Distribution of Token Counts in the Manual NLI Dataset

Figure 3 shows the frequency distribution of the token counts in the manual NLI dataset. The counts for most sentences are distributed between 1 and 100, with about 83.0% of them having counts between 4 and 32. Accordingly, we used sentences within that range in this work.

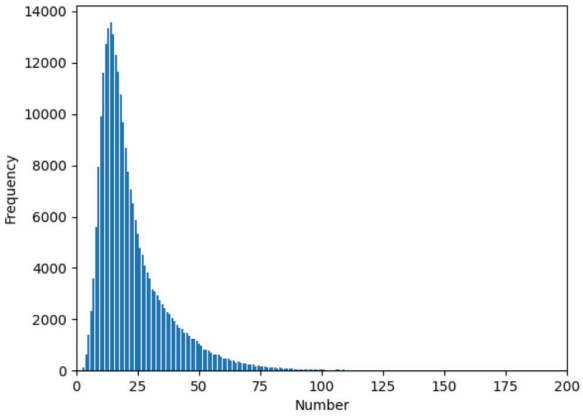

Figure 3: Frequency distribution of token counts in the manual NLI dataset

B Examples of Automatically Generated NLI Datasets

Tables 2 and 3 provide examples of NLI data that were automatically generated with 0-shot and 20-shot learning, respectively, for the same premise sentences.

We observed that some sentences generated in the zero-shot setting are predicted as neutral, but sentences close to entailment and contradiction can also be generated. By shifting the period or using negation, diverse entailment and contradiction sentences can be created. The sentences generated with 20-shot learning tended to strongly refer to the premise sentence, indicating higher precision in generating both entailment and contradiction sentences. Additionally, these sentences tended to be shorter than those generated with 0-shot learning.

C Evaluation of Transfer Tasks

To evaluate the effectiveness of the generated sentence embeddings in transfer tasks, we conducted evaluations with transfer tasks from SentEval (Conneau and Kiela, 2018). These tasks use sentence embeddings as input and train a linear classifier. Specifically, the embeddings generated from each sentence are used as features to train linear classifiers, such as logistic regression. The classification task performance is assessed with the trained classifier, and the accuracy and other related metrics are measured to quantitatively evaluate the effectiveness of the sentence representations.

Table 4 summarizes the results. As reported in Jiang et al. (2023), we could not confirm any performance improvement on the transfer tasks via fine-tuning; however, the scores for both the zero-shot and few-shot settings were comparable

Assigned label	Premise and generated hypothesis sentence	Predicted label
Premise	It concluded in July 2019.	-
Entailment	It was completed in July 2019.	**Entailment**
Contradiction	It did not conclude in July 2019.	**Contradiction**
Premise	He spent several months in prison.	-
Entailment	He was unable to pay his bills while in prison.	Neutral
Contradiction	He was not in prison for any amount of time.	**Contradiction**
Premise	Her last public performance was in 1954.	-
Entailment	She has not performed in public since 1954.	**Entailment**
Contradiction	Her last public performance was last week.	**Contradiction**
Premise	It grows on a many different soils.	-
Entailment	It grows on any soil that is suitable for the growth of other crops.	Neutral
Contradiction	It does not grow on soil with high pH levels.	Neutral

Table 2: Examples of the NLI dataset generated automatically with zero-shot learning. The predicted labels matched the assigned label are shown in **bold**.

Assigned label	Premise and generated hypothesis sentence	Predicted label
Premise	It concluded in July 2019.	-
Entailment	July 2019 occurred.	**Entailment**
Contradiction	It began in January 2020.	**Contradiction**
Premise	He spent several months in prison.	-
Entailment	He was in prison for several months.	**Entailment**
Contradiction	He was never in prison.	**Contradiction**
Premise	Her last public performance was in 1954.	-
Entailment	She performed in 1954.	**Entailment**
Contradiction	She is still actively touring and performing.	**Contradiction**
Premise	It grows on a many different soils.	-
Entailment	The plant grows on various soils.	**Entailment**
Contradiction	It only grows on sandy soils.	**Contradiction**

Table 3: Examples of the NLI dataset generated automatically with 20-shot learning. The predicted labels matched the assigned label are shown in **bold**.

to those with training on the manual NLI dataset. Tables 5 and 6 show detailed scores for each experiment. In the proposed method, it is confirmed that the score is low and unstable when the data size is small, but it stabilizes as the data size increases.

D Detailed STS Scores

Table 7 shows the performances of the STS tasks for each model with 256,000 examples. Additionally, Tables 8 and 9 show detailed performances of the STS tasks for each experiment. It is evident that good sentence embedding models have been created without any extreme highs or lows for any dataset. Furthermore, the 1-shot setting tends to have a larger variance, while the variance tends to decrease as the number of shots increases. This confirms the validity of increasing the number of trials as the number of shots increases.

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
Without fine-tuning (base model: LLaMA-2-7B)								
PromptEOL	90.53	92.45	96.22	91.24	95.39	96.20	74.96	$91.00_{\pm 0.000}$
Fine-tuning on unsupervised dataset								
PromptRoBERTa	82.88	88.14	94.13	87.22	87.97	88.60	74.63	$86.22_{\pm 0.159}$
Fine-tuning on automatically generated dataset (base model: LLaMA-2-7B)								
PromptEOL (0-shot)	90.00	92.58	95.23	90.56	94.07	94.00	73.39	$89.97_{\pm 0.511}$
PromptEOL (1-shot)	89.93	92.63	95.28	90.62	94.32	94.76	72.17	$89.96_{\pm 0.248}$
PromptEOL (1-shot×5)	90.38	92.75	95.49	90.61	94.53	95.28	72.79	$90.26_{\pm 0.312}$
PromptEOL (5-shot)	89.63	92.52	94.74	90.68	93.84	95.16	73.77	$90.05_{\pm 0.086}$
PromptEOL (5-shot×4)	89.63	92.52	94.74	90.68	93.84	95.16	73.77	$90.05_{\pm 0.293}$
PromptEOL (20-shot)	89.33	92.76	94.71	91.19	93.66	93.65	74.20	$89.93_{\pm 0.240}$
Fine-tuning on manual dataset (base model: LLaMA-2-7B)								
PromptEOL	89.94	93.22	96.05	90.83	94.89	95.40	74.26	$90.65_{\pm 0.227}$

Table 4: Transfer task results of different sentence embedding models (measured as accuracy). 256,000 sentence pairs were used for fine-tuning the model. The average performance (Avg.) is provided along with the respective standard deviation.

Dataset size	Setting	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
4000		90.08	36.24	54.58	**90.87**	**94.47**	1.80	33.51	$57.36_{\pm 0.045}$
8000		90.26	92.25	95.65	90.27	93.57	**94.73**	71.92	$89.81_{\pm 0.172}$
16000		90.24	92.30	**95.73**	90.44	79.33	94.60	59.56	$86.03_{\pm 5.630}$
32000	0-shot	76.72	73.47	80.37	90.36	79.39	63.53	60.21	$74.87_{\pm 21.39}$
64000		**90.46**	**92.64**	95.71	90.67	94.23	**94.73**	72.89	$\mathbf{90.19}_{\pm 0.065}$
128000		89.83	92.49	95.61	90.34	94.40	94.33	**73.74**	$90.10_{\pm 0.143}$
256000		90.00	92.58	95.23	90.56	94.07	94.00	73.39	$89.97_{\pm 0.511}$
4000		85.75	48.61	55.01	88.57	58.99	56.98	33.51	$61.06_{\pm 12.59}$
8000		89.84	92.50	95.14	90.44	93.85	94.74	65.68	$88.88_{\pm 2.359}$
16000		89.78	92.62	95.23	90.63	93.78	94.48	**72.89**	$89.92_{\pm 0.500}$
32000	1-shot	89.66	87.21	91.01	90.59	89.67	**94.88**	68.63	$87.38_{\pm 6.002}$
64000		**90.23**	87.18	90.22	**91.11**	89.73	91.86	70.49	$78.27_{\pm 8.334}$
128000		90.09	87.43	82.26	90.58	94.17	94.36	53.26	$84.59_{\pm 6.580}$
256000		89.93	**92.63**	**95.28**	90.62	**94.32**	94.76	72.17	$\mathbf{89.96}_{\pm 0.248}$
4000		**90.46**	48.67	50.00	**90.63**	67.98	57.84	33.51	$62.73_{\pm 9.295}$
8000		90.29	92.23	79.04	90.50	85.45	94.84	72.98	$86.48_{\pm 4.861}$
16000		90.43	92.30	95.46	90.42	94.29	95.08	**74.15**	$\mathbf{90.30}_{\pm 0.221}$
32000	1-shot×5	90.16	92.58	95.03	90.56	93.97	94.96	74.11	$90.20_{\pm 0.055}$
64000		90.20	92.39	95.45	90.62	94.08	94.80	73.14	$90.10_{\pm 0.292}$
128000		90.12	92.57	95.46	90.49	**94.71**	**95.48**	73.19	$90.29_{\pm 0.265}$
256000		90.38	**92.75**	**95.49**	90.61	94.53	95.28	72.79	$90.26_{\pm 0.312}$
4000		81.68	81.45	68.82	86.71	85.06	57.76	58.04	$74.22_{\pm 18.80}$
8000		81.95	70.21	76.97	90.67	76.69	57.96	58.44	$73.27_{\pm 21.13}$
16000		89.83	92.73	94.94	90.76	**94.00**	94.96	74.06	$\mathbf{90.19}_{\pm 0.571}$
32000	5-shot	**89.95**	92.80	94.98	**90.95**	93.81	94.80	73.84	$90.16_{\pm 0.354}$
64000		89.44	92.72	94.82	90.74	93.77	95.00	**74.25**	$90.10_{\pm 0.289}$
128000		89.78	**92.85**	**95.04**	90.72	93.77	94.72	73.43	$90.04_{\pm 0.310}$
256000		89.63	92.52	94.74	90.68	93.84	**95.16**	73.77	$90.05_{\pm 0.293}$
4000		79.60	50.30	50.00	85.72	61.22	25.20	33.51	$55.08_{\pm 13.70}$
8000		**89.86**	80.20	72.67	90.71	83.77	94.60	43.84	$79.38_{\pm 9.888}$
16000		89.52	66.02	73.66	90.79	82.84	71.85	54.02	$75.53_{\pm 16.47}$
32000	5-shot×4	89.64	80.08	83.42	90.71	82.66	**94.65**	63.57	$83.53_{\pm 11.15}$
64000		89.39	**93.10**	94.75	**91.01**	93.37	94.25	**73.57**	$\mathbf{89.92}_{\pm 0.303}$
128000		89.74	92.84	**95.00**	90.89	93.93	93.65	63.49	$88.51_{\pm 2.530}$
256000		89.64	92.76	94.84	90.62	**94.13**	94.10	73.07	$89.88_{\pm 0.264}$
4000		79.61	36.24	50.00	80.78	50.08	1.80	33.51	$47.43_{\pm 3.983}$
8000		70.49	36.24	50.00	75.84	50.08	24.90	33.51	$48.73_{\pm 9.173}$
16000		**89.68**	78.65	72.32	90.54	82.22	70.60	54.20	$76.89_{\pm 18.61}$
32000	20-shot	89.19	**92.86**	94.60	90.51	93.25	**94.65**	74.12	$89.88_{\pm 0.070}$
64000		89.62	92.78	**94.89**	91.00	93.84	94.15	**74.47**	$\mathbf{90.11}_{\pm 0.205}$
128000		88.89	92.82	94.67	90.92	92.92	94.30	74.21	$89.82_{\pm 0.146}$
256000		89.33	92.76	94.71	**91.19**	**93.66**	93.65	74.20	$89.93_{\pm 0.240}$

Table 5: The results of PromptEOL-LLaMA-2-7B fine-tuned with the automatically generated dataset (measured as accuracy). The average performance (Avg.) is provided along with the respective standard deviation.

Model	Dataset size	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
Without fine-tuning (base model: LLaMA-2-7B)									
PromptEOL	-	59.91	78.86	68.74	75.71	73.39	73.48	71.26	$71.62_{\pm 0.000}$
Fine-tuning on unsupervised dataset									
PromptRoBERTa	4000	**83.90**	**88.78**	**95.31**	86.72	**89.16**	**93.73**	74.07	$\mathbf{87.38}_{\pm 0.076}$
	8000	83.58	88.54	95.31	86.53	88.65	91.67	74.36	$86.95_{\pm 0.081}$
	16000	83.19	87.41	94.93	86.66	88.17	90.40	73.43	$86.32_{\pm 0.017}$
	32000	83.06	87.58	94.64	86.84	87.99	88.47	73.35	$85.99_{\pm 0.080}$
	64000	82.96	87.72	94.42	87.01	87.66	88.80	74.14	$86.10_{\pm 0.033}$
	128000	82.58	87.73	94.18	86.94	87.66	88.47	74.13	$85.95_{\pm 0.186}$
	256000	82.88	88.14	94.13	**87.22**	87.97	88.60	**74.63**	$86.22_{\pm 0.159}$
Fine-tuning on automatically generated dataset (base model: LLaMA-2-7B)									
PromptEOL (0-shot)	4000	90.08	36.24	54.58	**90.87**	**94.47**	1.80	33.51	$57.36_{\pm 0.045}$
	8000	90.26	92.25	95.65	90.27	93.57	**94.73**	71.92	$89.81_{\pm 0.172}$
	16000	90.24	92.30	**95.73**	90.44	79.33	94.60	59.56	$86.03_{\pm 5.630}$
	32000	76.72	73.47	80.37	90.36	79.39	63.53	60.21	$74.87_{\pm 21.39}$
	64000	**90.46**	92.64	95.71	90.67	94.23	**94.73**	72.89	$\mathbf{90.19}_{\pm 0.065}$
	128000	89.83	92.49	95.61	90.34	94.40	94.33	**73.74**	$90.10_{\pm 0.143}$
	256000	90.00	**92.58**	95.23	90.56	94.07	94.00	73.39	$89.97_{\pm 0.511}$
PromptEOL (5-shot×4)	4000	79.60	50.30	50.00	85.72	61.22	25.20	33.51	$55.08_{\pm 13.70}$
	8000	**89.86**	80.20	72.67	90.71	83.77	94.60	43.84	$79.38_{\pm 9.888}$
	16000	89.52	66.02	73.66	90.79	82.84	71.85	54.02	$75.53_{\pm 16.47}$
	32000	89.64	80.08	83.42	90.71	82.66	**94.65**	63.57	$83.53_{\pm 11.15}$
	64000	89.39	**93.10**	94.75	**91.01**	93.37	94.25	**73.57**	$\mathbf{89.92}_{\pm 0.303}$
	128000	89.74	92.84	**95.00**	90.89	93.93	93.65	63.49	$88.51_{\pm 2.530}$
	256000	89.64	92.76	94.84	90.62	**94.13**	94.10	73.07	$89.88_{\pm 0.264}$
Fine-tuning on manually annotated dataset (base model: LLaMA-2-7B)									
PromptEOL	4000	88.06	92.14	66.25	90.53	92.39	93.27	73.93	$85.23_{\pm 2.435}$
	8000	88.18	92.82	94.90	90.39	93.14	94.00	**74.80**	$89.75_{\pm 0.215}$
	16000	88.81	92.93	81.54	90.34	79.11	92.93	73.51	$85.59_{\pm 5.488}$
	32000	89.68	93.13	95.34	90.20	79.74	95.00	73.39	$88.07_{\pm 3.070}$
	64000	89.78	**93.30**	96.02	90.54	**94.89**	95.33	73.80	$90.52_{\pm 0.041}$
	128000	**90.12**	93.07	**96.07**	90.82	94.33	95.33	74.30	$90.57_{\pm 0.131}$
	256000	89.94	93.22	96.05	**90.83**	**94.89**	**95.40**	74.26	$\mathbf{90.65}_{\pm 0.227}$

Table 6: Transfer task results of different sentence embedding models (measured as accuracy). The average performance (Avg.) is provided along with the respective standard deviation.

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Without fine-tuning (base model: LLaMA-2-7B)								
PromptEOL	59.91	78.86	68.74	75.71	73.39	73.48	71.26	$71.62_{\pm 0.000}$
Fine-tuning on unsupervised dataset								
PromptRoBERTa	73.64	84.97	77.44	85.11	81.61	82.12	69.09	$79.14_{\pm 0.175}$
Fine-tuning on automatically generated dataset (base model: LLaMA-2-7B)								
PromptEOL (0-shot)	71.76	86.47	80.53	83.26	83.75	82.45	71.95	$80.02_{\pm 0.485}$
PromptEOL (1-shot)	73.30	87.61	81.52	85.35	83.85	83.63	76.86	$81.73_{\pm 1.140}$
PromptEOL (1-shot×5)	73.27	87.90	81.74	85.72	84.11	84.66	76.45	$81.98_{\pm 0.837}$
PromptEOL (5-shot)	73.72	87.75	81.94	85.71	83.85	84.49	75.72	$81.88_{\pm 0.846}$
PromptEOL (5-shot×4)	74.16	87.75	82.65	85.95	84.97	85.26	76.44	$82.45_{\pm 0.385}$
PromptEOL (20-shot)	74.24	87.39	82.55	85.49	84.36	85.24	74.20	$81.92_{\pm 0.183}$
Fine-tuning on manual dataset (base model: LLaMA-2-7B)								
PromptEOL	78.75	89.99	84.98	88.82	86.27	88.37	82.44	$85.66_{\pm 0.101}$

Table 7: Spearman's rank correlation coefficient between the cosine similarity of the sentence embeddings and the human ratings. All values in the table are multiplied by 100. 256,000 sentence pairs were used for fine-tuning the model. The average performance (Avg.) is provided along with the respective standard deviation.

Dataset size	few-shot	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
4000		65.68	83.92	76.18	80.00	79.95	78.82	**76.80**	$77.34_{\pm 0.185}$
8000		68.80	85.60	78.42	81.51	81.79	81.26	73.16	$78.65_{\pm 0.159}$
16000		69.37	85.62	77.97	81.56	81.99	81.76	74.37	$78.95_{\pm 0.362}$
32000	0-shot	71.59	85.93	78.40	82.34	82.24	81.51	74.44	$79.49_{\pm 0.553}$
64000		71.21	86.09	80.28	**83.60**	83.21	81.43	73.19	$79.86_{\pm 0.137}$
128000		70.84	86.40	80.15	83.12	82.44	82.29	73.71	$79.85_{\pm 0.310}$
256000		**71.76**	**86.47**	**80.53**	83.26	**83.75**	**82.45**	71.95	$\mathbf{80.02}_{\pm 0.485}$
4000		70.14	85.97	78.94	82.05	82.42	82.30	**77.38**	$79.89_{\pm 1.038}$
8000		73.03	**87.82**	81.52	85.47	83.79	**84.57**	77.13	$\mathbf{81.90}_{\pm 0.764}$
16000		72.92	87.63	81.32	85.29	83.67	84.28	76.96	$81.73_{\pm 0.959}$
32000	1-shot	72.82	87.34	81.05	84.99	83.38	83.71	76.76	$81.44_{\pm 2.185}$
64000		**73.30**	87.61	81.52	85.35	**83.85**	83.63	76.86	$81.73_{\pm 1.608}$
128000		73.01	87.59	**81.55**	**85.51**	83.77	84.42	76.76	$81.80_{\pm 0.785}$
256000		**73.30**	87.61	81.52	85.35	**83.85**	83.63	76.86	$81.73_{\pm 1.140}$
4000		69.67	86.03	78.96	82.63	82.74	82.28	76.72	$79.86_{\pm 0.846}$
8000		73.05	87.43	81.40	84.95	83.60	84.30	77.15	$81.70_{\pm 0.920}$
16000		72.94	87.79	81.57	85.03	83.70	**84.56**	**77.26**	$81.83_{\pm 0.544}$
32000	1-shot×5	72.83	87.81	81.26	84.88	83.54	83.97	76.81	$81.59_{\pm 0.198}$
64000		**74.28**	**87.85**	**81.92**	85.53	**84.13**	84.21	75.87	$81.97_{\pm 0.370}$
128000		72.60	87.50	81.32	85.35	83.45	84.18	75.72	$81.44_{\pm 0.379}$
256000		73.27	87.90	81.74	**85.72**	84.11	84.66	76.45	$\mathbf{81.98}_{\pm 0.837}$
4000		70.40	85.99	79.35	82.48	82.61	82.15	76.11	$79.87_{\pm 3.850}$
8000		72.58	87.23	80.92	84.63	83.47	84.22	76.55	$81.37_{\pm 1.962}$
16000		73.34	87.52	81.52	85.53	83.57	84.72	**76.93**	$81.88_{\pm 1.081}$
32000	5-shot	**73.81**	87.33	**81.78**	85.27	83.67	**84.86**	76.01	$81.82_{\pm 0.947}$
64000		73.50	87.75	81.67	**85.76**	83.71	84.69	76.67	$\mathbf{81.97}_{\pm 1.109}$
128000		73.68	87.52	81.65	85.25	83.57	84.85	74.71	$81.60_{\pm 1.111}$
256000		73.72	**87.75**	81.94	85.71	**83.85**	84.49	75.72	$81.88_{\pm 0.846}$
4000		71.82	87.20	80.36	83.39	83.63	83.46	74.88	$80.68_{\pm 0.686}$
8000		73.10	87.83	82.08	84.98	84.21	84.89	**76.70**	$81.97_{\pm 0.207}$
16000		73.63	**88.01**	82.38	85.55	84.05	85.25	75.87	$82.10_{\pm 0.416}$
32000	5-shot×4	74.67	87.92	82.02	85.68	84.52	85.43	75.72	$82.28_{\pm 0.445}$
64000		**74.88**	87.93	**82.83**	**86.23**	84.68	**86.00**	76.40	$\mathbf{82.71}_{\pm 0.322}$
128000		74.11	87.55	82.00	85.51	84.22	85.30	75.78	$82.06_{\pm 0.209}$
256000		74.16	87.75	82.65	85.95	**84.97**	85.26	76.44	$82.45_{\pm 0.385}$
4000		71.48	86.75	80.11	82.72	83.08	83.67	74.44	$80.32_{\pm 0.472}$
8000		72.55	87.38	81.46	83.75	83.14	84.62	76.01	$81.27_{\pm 0.093}$
16000		73.17	87.10	81.54	84.79	83.47	84.97	75.50	$81.50_{\pm 0.575}$
32000	20-shot	74.11	87.67	82.14	**85.59**	84.22	**85.56**	**75.83**	$\mathbf{82.16}_{\pm 0.315}$
64000		73.73	87.24	81.69	84.78	83.76	85.00	74.83	$81.58_{\pm 0.182}$
128000		74.15	**87.44**	82.36	85.16	83.91	85.30	74.38	$81.81_{\pm 0.558}$
256000		**74.24**	87.39	**82.55**	85.49	**84.36**	85.24	74.20	$81.92_{\pm 0.183}$

Table 8: The detailed results of the experiments conducted in Section 4.2. Spearman's rank correlation coefficient between the cosine similarity of the sentence embeddings of PromptEOL-LLaMA-2-7B fine-tuned with an automatically generated dataset with few-shot and the human evaluation. All values in the table are multiplied by 100. The average performance (Avg.) is provided along with the respective standard deviation.

Model	Dataset size	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Without fine-tuning (base model: LLaMA-2-7B)									
PromptEOL	-	59.91	78.86	68.74	75.71	73.39	73.48	71.26	$71.62_{\pm 0.000}$
Fine-tuning on unsupervised dataset									
PromptRoBERTa	4000	62.74	80.97	70.30	80.75	76.78	77.49	**71.24**	$74.33_{\pm 0.078}$
	8000	61.86	79.56	69.67	81.05	75.52	76.15	70.78	$73.51_{\pm 0.156}$
	16000	66.67	81.03	72.17	82.87	77.67	78.90	70.08	$75.63_{\pm 0.264}$
	32000	71.25	84.01	75.29	84.43	80.39	81.00	69.26	$77.95_{\pm 0.064}$
	64000	72.90	84.59	76.47	84.92	80.85	81.60	68.71	$78.58_{\pm 0.111}$
	128000	72.98	84.71	76.80	84.98	80.96	81.68	68.77	$78.70_{\pm 0.273}$
	256000	**73.64**	**84.97**	**77.44**	**85.11**	**81.61**	**82.12**	69.09	$\mathbf{79.14}_{\pm 0.175}$
Fine-tuning on automatically generated dataset (base model: LLaMA-2-7B)									
PromptEOL (0-shot)	4000	65.68	83.92	76.18	80.00	79.95	78.82	**76.80**	$77.34_{\pm 0.185}$
	8000	68.80	85.60	78.42	81.51	81.79	81.26	73.16	$78.65_{\pm 0.159}$
	16000	69.37	85.62	77.97	81.56	81.99	81.76	74.37	$78.95_{\pm 0.362}$
	32000	71.59	85.93	78.40	82.34	82.24	81.51	74.44	$79.49_{\pm 0.553}$
	64000	71.21	86.09	80.28	**83.60**	83.21	81.43	73.19	$79.86_{\pm 0.137}$
	128000	70.84	86.40	80.15	83.12	82.44	82.29	73.71	$79.85_{\pm 0.310}$
	256000	**71.76**	**86.47**	**80.53**	83.26	**83.75**	**82.45**	71.95	$\mathbf{80.02}_{\pm 0.485}$
PromptEOL (5-shot×4)	4000	71.82	87.20	80.36	83.39	83.63	83.46	74.88	$80.68_{\pm 0.686}$
	8000	73.10	87.83	82.08	84.98	84.21	84.89	**76.70**	$81.97_{\pm 0.207}$
	16000	73.63	**88.01**	82.38	85.55	84.05	85.25	75.87	$82.10_{\pm 0.416}$
	32000	**74.67**	87.92	82.02	85.68	84.52	85.43	75.72	$\mathbf{82.28}_{\pm 0.445}$
	64000	74.88	87.93	82.83	**86.23**	84.68	**86.00**	76.40	$82.71_{\pm 0.322}$
	128000	74.11	87.55	82.00	85.51	84.22	85.30	75.78	$82.06_{\pm 0.209}$
	256000	74.16	87.75	**82.65**	85.95	**84.97**	85.26	76.44	$82.45_{\pm 0.385}$
Fine-tuning on manually annotated dataset (base model: LLaMA-2-7B)									
PromptEOL	4000	73.68	87.41	81.45	86.06	83.74	86.18	**82.85**	$83.05_{\pm 0.423}$
	8000	74.93	87.90	82.61	86.72	84.67	87.17	82.83	$83.83_{\pm 0.298}$
	16000	76.03	87.99	82.88	87.24	84.83	87.21	82.82	$84.14_{\pm 0.504}$
	32000	76.71	88.26	83.08	87.22	84.75	87.52	81.99	$84.22_{\pm 1.270}$
	64000	78.26	89.64	84.71	**88.86**	85.67	88.18	81.95	$85.32_{\pm 0.117}$
	128000	78.28	89.89	84.80	**88.86**	85.83	88.35	81.88	$85.41_{\pm 0.172}$
	256000	**78.75**	**89.99**	**84.98**	88.82	**86.27**	**88.37**	82.44	$\mathbf{85.66}_{\pm 0.101}$

Table 9: The detailed results of the experiments conducted in Section 4.3. Spearman's rank correlation coefficient between the cosine similarity of the sentence embeddings and the human evaluation. All values in the table are multiplied by 100. The average performance (Avg.) is provided along with the respective standard deviation.