Scaling Sentence Embeddings with Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have recently gained significant interest due to their impressive results in various natural language tasks. However, their application to sentence embeddings is still under active research. In this work, we introduce PromptEOL, a simple and efficient method designed to enhance LLM performance on sentence embeddings with a one-word limitation. We further integrate PromptEOL with in-context learning and alignment to leverage LLMs in two settings: without fine-tuning and with fine-tuning. Our extensive experiments show that PromptEOL enables LLMs to generate superior sentence embeddings without fine-tuning, outperforming contrastive learning methods. Additionally, with fine-tuning, a 2.7B parameter model using PromptEOL surpasses the performance of a 4.8B parameter model from previous methods. We also analyze how scaling model parameters, from 125 million to 66 billion, impacts sentence embedding performance.

1 Introduction

Sentence embeddings is a fundamental problem in natural language processing, requiring language models to project sentences into a vector space based on their semantics. Current methods based on contrastive learning, such as SimCSE (Gao et al., 2021), have successfully leveraged pretrained language models to generate high-quality embeddings. A significant amount of research has been devoted to refining the contrastive learning framework in order to further improve sentence embeddings (Chuang et al., 2022; Wu et al., 2022a,b; Cheng et al., 2023).

Recently, LLMs, such as GPT-3 (Brown et al., 2020) and LLaMA (Touvron et al., 2023a), have demonstrated significant potential on various natural language processing tasks such as translation, question answering, and text classification. Current research has also explored the application of LLMs for data augmentation in sentence embeddings. By generating better sentence pairs for contrastive learning, LLMs can help alleviate the scarcity of labeled data (Cheng et al., 2023; Zhang et al., 2023). However, directly utilizing LLMs to generate sentence embeddings presents two primary challenges. Firstly, LLMs, as autoregressive models, produce text instead of vectors, which necessitates vectorizing the output. Secondly, it is crucial to determine an effective approach for incorporating the capabilities of in-context learning into sentence embeddings.

In this work, we aim to investigate the capabilities of current LLMs for sentence embeddings, facilitated by the availability of open-source LLMs (Touvron et al., 2023a; Zhang et al., 2022). We address the following research questions: 1) How can LLMs be used to represent sentence embeddings, and does prompt engineering (Jiang et al., 2022) help? 2) Can in-context learning (Liu et al., 2023) enhance the quality of sentence embeddings? 3) Does the scaling up the model parameters still work when the number of parameters exceeds billions?

To address these questions, we conduct a systematic study by evaluating LLaMA (Touvron et al., 2023a) and OPT (Zhang et al., 2022) on both semantic textual similarity (STS) tasks and transfer tasks. Following (Jiang et al., 2022), we utilize a prompt such as *This sentence:* “ `[text]` ” *means* to enable LLMs to generate sentence embeddings, where `[text]` serves as the input slot. This method outperforms traditional representation methods, such as averaging output tokens to represent sentences. Considering the causal architecture and pretraining tasks of LLMs compared to BERT, we propose a method callled PromptEOL to refine the prompt to generate better representations by instructing LLMs to encapsulate as much semantic information of the sentences as possible within the target token.

Inspired by (Tsukagoshi et al., 2021), which uses definition sentences from a word dictionary to learn sentence embeddings, we find that performance can be further improved by adding definition sentences and corresponding words as examples to perform in-context learning. To mitigate the gap between examples and input sentences, we also use sentences from the STS-B (Cer et al., 2017) training set as examples by instructing ChatGPT to generate a single word to represent the meaning of sentences. By evaluating the demonstration examples based on the STS-B development set, LLMs can outperform previous contrastive learning-based sentence models, which were fine-tuned on unsupervised data.

We scale up the parameters of LLMs in two settings: without and with fine-tuning. For the settings without fine-tuning, we find that transitioning from millions to billions of parameters results in improvements on STS tasks. However, continued scaling may not yield further improvements. One explanation corresponds to anisotropy in embeddings. We note that larger LLMs have greater anisotropy, which may limit their performance. For the settings with fine-tuning, since anisotropy can be alleviated by contrastive learning (Gao et al., 2021), LLMs with tens of billions of parameters exhibit strong performance.

Our main contributions are as follows:

1. We propose a method called PromptEOL that leverages LLMs to enhance the representation of sentences. Additionally, we further improve it by our in-context learning framework.

2. We conduct an analysis of scaling up the parameters of LLMs from millions to tens of billions in sentence embeddings with and without fine-tuning.

3. We propose a method to refine sentence representation with alignment. Based on these methods, we achieve 86.76 Spearman correlation on STS tasks, a 1.8 improvement over the previous method.

2 Related Work

Sentence Embeddings Sentence embeddings is to convert a sentence into a fixed-size vector, which captures the semantic meaning and context of the sentence. It allows for the efficient retrieval of similar sentences through the similarity between vectors. Recently, SimCSE (Gao et al., 2021) demonstrated that contrastive learning is an effective approach for learning sentence embeddings using BERT. DiffCSE (Chuang et al., 2022) incorporates a replaced token detection loss into the contrastive learning framework. PromptBERT (Jiang et al., 2022) reveals that prompts can enhance BERT's ability to represent sentences. Additionally, several studies (Cheng et al., 2023; Zhang et al., 2023) have investigated data augmentation for sentence embeddings using LLMs. SentenceT5 (ST5) (Ni et al., 2021) leverages the encoder-decoder structure of models for generating sentence embeddings and demonstrates improvements by scaling T5 from millions to billions of parameters. However, directly using LLMs to generate sentence embeddings remains an area of ongoing research.

Large Language Models LLMs (Zhang et al., 2022; Scao et al., 2022; Chowdhery et al., 2022; Touvron et al., 2023a) recently show impressive performance on various natural language process, benefiting from their large parameter sizes compared to previous pretrained language models. LLMs can efficiently learn a new task with in-context learning by using training data as demonstrations (Brown et al., 2020). Without any gradient updates, LLMs with in-context learning can solve challenging tasks like multitask language understanding (Hendrycks et al., 2020), commonsense reasoning (Lin et al., 2021), and math problems (Cobbe et al., 2021). This performance can be further improved by scaling up language models (Hoffmann et al., 2022; Kaplan et al., 2020).

3 Methodology

In this section, we first discuss current sentence embeddings methods with LLMs, and then introduce a new Prompt-based method with Explicit One word Limitation (PromptEOL) for LLMs in Section 3.1. Based on this method, we describe methods without fine-tuning in Section 3.2 and with fine-tuning in Section 3.3, respectively.

3.1 PromptEOL

Previous works (Li et al., 2020; Su et al., 2021; Jiang et al., 2022) have extensively studied on improving sentence embeddings from encoder-based pretrained models, like BERT without fine-tuning.

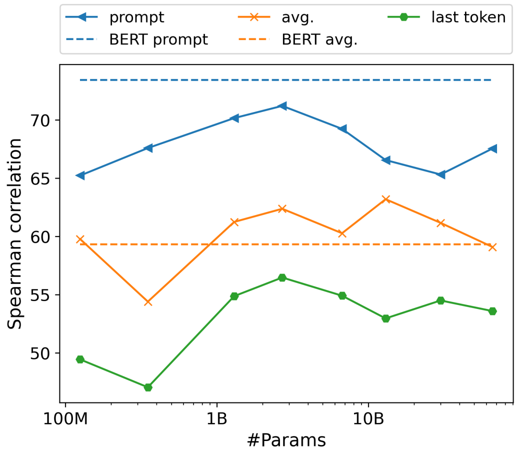

Figure 1: Performances of OPT in STS-B development set with three representation methods. Dash lines represent the results of BERT.

Recently, PromptBERT (Jiang et al., 2022) leverages a prompt-based method to represent sentence. It uses manual templates like *This sentence: "* `[text]` *" means* `[MASK]`., where `[text]` is the placeholder for a sentence. The output vector of `[MASK]` token is used as sentence embeddings. It demonstrates superior results compared to previous sentence representation methods like averaging output hidden vectors or the output vector of `[CLS]` token.

Considering to LLMs as autoregression models, which do not have special tokens like `[CLS]` or `[MASK]`, we modify the prompt-based method in (Jiang et al., 2022) to make it compatible with LLMs. We use *This sentence: "* `[text]` *" means* to prompt LLMs generate next token and extract the hidden vectors of the final token as sentence embeddings. To validate the prompt-based method with LLMs, we compare it with two other methods, such as averaging or using the last token as sentence embeddings. For LLMs, we use OPT (Zhang et al., 2022) from 125 million parameters to 66 billions and evaluate it on STS-B development set in Figure 1. Following the results in (Jiang et al., 2022), we observe that prompt-based method can enhance sentence representation across all OPTs, ranging from millions to billions parameters. Despite that the previous prompt-based method also improved LLMs like OPT on sentence representations, OPT still fails to outperform BERT.

For the bidirectional attention in BERT, we hypothesize that BERT can implicitly condense the entire semantic information corresponding to a sentence into a single `[MASK]` token when using templates like "*This sentence: "* `[text]` *" means* `[MASK]`.". Since the `[MASK]` token follows a period, this implicitly restricts BERT to explain meaning into one word. However, this template fails to add the similar "one word limitation" when it is used in autoregression models like OPT with unidirectional attention. To validate this, we simply remove the period in template to transfer it into "*This sentence: "* `[text]` *" means* `[MASK]`". Despite only one word difference, and no modification to meaning of the template, the performance of BERT on STS-B development set plummeted from 73.44 to 33.89 Spearman correlation, which means BERT without this implicit "one word limitation" fails to represent sentence.

Inspired by this, our objective is to enhance prompt-based method for LLMs by introducing a "one word limitation". We propose a new Prompt-based method with Explicit One word Limitation (PromptEOL) for LLMs. PromptEOL is simple and straightforward by directly adding some tokens in the template to instruct LLMs in predicting the meaning of sentence in one word. The template we used after modification is following:

This sentence: " `[text]` *" means in one word: "*

Note that the one-word limitation does not mean representing the sentence with a single word. Instead, it encourages the LLM to condense the semantic meaning of the sentence into the hidden state of the next token, which we use as the sentence embedding. We find this template improve all OPT models and allow them to match or even outperform BERT with prompt-based method in Figure 5.

3.2 In-context Learning Framework for Sentence Embeddings

In-context learning is widely utilized as an effective method to help LLMs understand problems. It improves their comprehension of inputs and outputs by directly adding a few examples in the prompts. However, when considering the problem of sentence embeddings, we need to project sentences into vectors based on their semantic information, separately. In other word, sentence embeddings lack textual outputs that could be used as examples to perform in-context learning, such

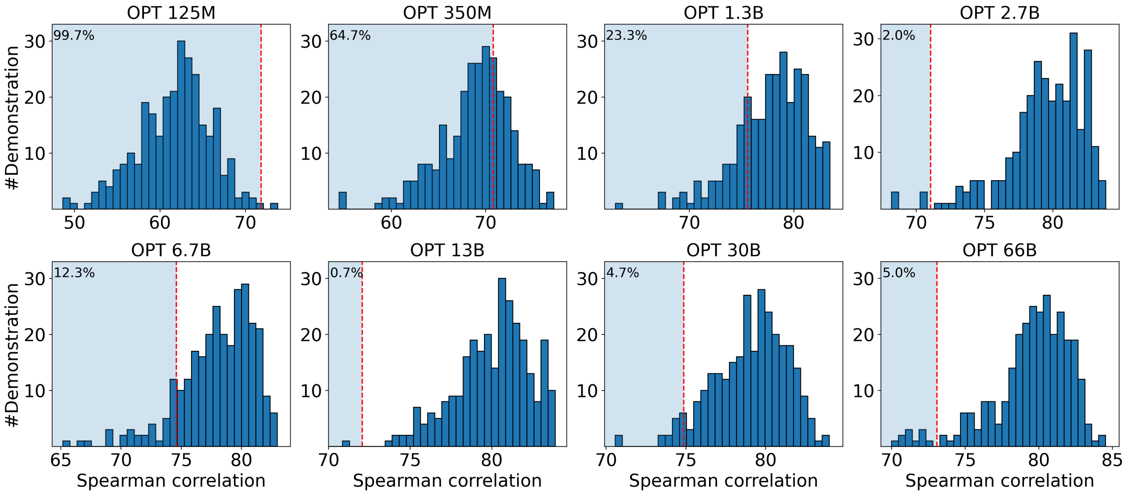

Figure 2: Distribution of Spearman correlations on the STS-B development set with different in-context learning demonstrations. The red dash line represents the Spearman correlation of the corresponding model without any demonstration. The blue area represents demonstrations that negatively impact the performance, and the percentage refers to the proportion of these demonstrations to the total number of demonstrations.

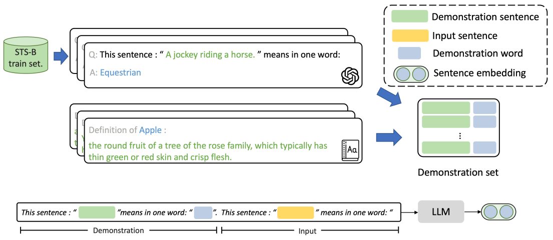
STS-B train set.
Q: This sentence : " A jockey riding a horse. " means in one word:
A: Equestrian
Definition of Apple :
the round fruit of a tree of the rose family, which typically has thin green or red skin and crisp flesh.
Demonstration sentence
Input sentence
Demonstration word
Sentence embedding
Demonstration set
This sentence : " "means in one word: " ". This sentence : " " means in one word: "
LLM
Demonstration
Input

Figure 3: An illustration of in-context learning based sentence embeddings. The green sentences denote the demonstration sentence, and the blue words denote the demonstration words. The corresponding color blocks refer to their slots in the template.

as answers for QA problems or labels for text classification problems. Moreover, there are also no predetermined gold vectors for a given sentence.

To leverage in-context learning in sentence embeddings, we propose an framework to automatically build demonstration sets and search demonstration to improve LLMs sentence embeddings in Figure 3. For the demonstration set, the goal is to create sentence and word pairs, where the word can represents the semantic information of the sentence. We propose two methods to generate pairs.

The first method involves using ChatGPT to generate corresponding words according to the semantic information of given sentences from STS-B training set. By asking ChatGPT with same template in Figure 3, ChatGPT outputs one word summary for the given sentence. We also find "one word limitation" in Section 3.1 is important for ChatGPT. Consider to our prompt-based representation method, we employ the hidden state of the next token as the sentence embeddings. By removing *in one word* from the template, it tends to explain the meaning of a sentence in a lengthy way, and the first word often becomes an article such as "The", which lacks clear meaning. For example, given the sentence "A jockey riding a horse.", the hidden state achieves the highest dot product similarity for "Equestrain" among its word embeddings. However, without "one word limitation", it will achieve the highest dot product similarity for word without specific meaning such as "The" among its word embeddings, which can not represent sentence properly. Inspired by DefSent (Tsukagoshi et al., 2021), which leverages definition sentences with their words as labels to train unsupervised sentence embedding, our second method is also based on a word dictionary. We directly use words and their definition sentences in the Oxford dictionary as word-sentence pairs.

Based on these methods, we construct a demonstration set consisting of 400 pairs of sentences and words. 200 pairs are from STS-B training set, with words labeled by ChatGPT, while the remaining are from the Oxford dictionary. To find demonstration that help model to represent sentences, we directly evaluate each demonstration on the STS-B development set and use the demonstration with the best Spearman correlation as the demonstration for corresponding models. We also visualize the distribution of Spearman correlations for OPT from 125M to 66B parameters in Figure 2. Following the previous study (Kaplan et al., 2020), we notice that in-context learning achieves better performance, when increasing model parameter from 125M to 2.7B. For example, there are only one demonstration that helps the 125M OPT achieve better performance compared to without demonstration. However, around 98% of demonstrations improve the performance of the 2.7B OPT. In-context learning significantly enhance the sentence embeddings, especially for OPT with more than 1B parameters. With only in-context learning, OPT with more than 1.3B parameters even achieve better results on STS tasks compared to contrastive learning based method like SimCSE (Gao et al., 2021) in Table 1.

3.3 Efficient Fine-tuning with Alignment

While in-context learning enhancing the performance of sentence embeddings without fine-tuning, we exploit PromptEOL with fine-tuning, and notice it also improves performance with contrastive learning in Figure 5.b, which also demonstrate the efficient of our representation method.

To further refine the sentence embeddings, we propose a method to align the sentence embeddings with preference sentence pairs, inspired by (Rafailov et al., 2023). Compared to contrastive learning, which teaches the model to distinguish positive and negative sentence pairs, our method considers that positive pairs can have different degrees of similarity. For instance, datasets like NLI, used in sentence embeddings, treat sentence pairs with the entailment label as positive pairs (Gao et al., 2021). Some of these positive pairs might differ by only a few words, while others may have completely different meanings.

The framework of our alignment method is shown in Figure 4. We use a sentence-pair regression model trained on STS-B as the scoring model to choose the preferred positive sentence pairs based on the similarity. Compared to sentence embedding models, this model inputs two sentences together and directly outputs the similarity score, resulting in more accurate similarity predictions for sentence pairs. The loss is defined as follows:

$$\mathcal{L}_{\text{Align}} = \log\sigma\left(\beta\log\frac{sim_{\pi_\theta}(x_3,x_4)}{sim_{\pi_{\text{ref}}}(x_3,x_4)} - \beta\log\frac{sim_{\pi_\theta}(x_1,x_2)}{sim_{\pi_{\text{ref}}}(x_1,x_2)}\right) \tag{1}$$

Where π_{ref} represents the reference model. π_θ denotes the optimal model. We warmup them by using contrastive learning with 500 steps on training data. The term sim refers to the function for computing similarity between sentence pairs. x_1, x_2 and x_3, x_4 are aligned sentence pairs, where regression model prefers first pair as indicated by $sim(x_3, x_4) \succ sim(x_1, x_2)$. To choose the preferred positive pairs, we sort the sentence pairs by the similarity score predicted by the regression model and split them into two groups: the first 50% as preferred positive pairs and the remaining as rejected positive pairs.

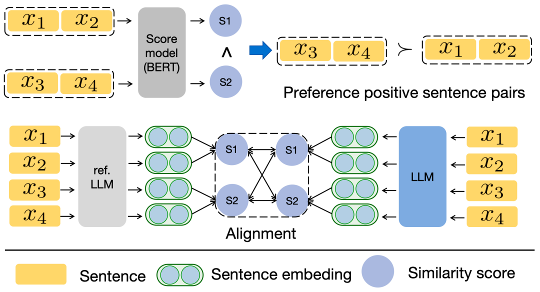

Figure 4: The framework of alignment method for sentence embeddings.

4 Experiment

4.1 Implementation Details

For the setting without fine-tuning, we use OPT from 125M to 66B parameters, and LLaMA from 7B to 65B parameters. All models use the same template in Section 3.1. We use 400 pairs of sentences and words as demonstration set for in-context learning. Among these, 200 pairs are from the STS-B training set, and we use `gpt-3.5-turbo` to label their words. The remaining 200 pairs are from the Oxford dictionary. For each model, we choose only one demonstration that has the highest Spearman correlation on the STS-B development set as their demonstration for evaluation. All results from models with 16-bit weights. We also present results using quantization methods in Appendix A. For the setting with fine-tuning, we following the LoRA settings in QLoRA (Dettmers et al., 2023) and train models on NLI datasets following (Gao et al., 2021) with one epoch for contrastive learning. We use the same training data with roberta-large fine-tuned on STS-B training set as preference model. More training details can be found in Appendix B. For the evaluation datasets, we use 7 STS tasks and 7 transfer tasks following (Gao et al., 2021).

Method	Params	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Without fine-tuning									
BERT avg.†	110M	30.87	59.89	47.73	60.29	63.73	47.29	58.22	52.57
BERT prompt‡	110M	60.96	73.83	62.18	71.54	68.68	70.60	67.16	67.85
ST5-Enc§	4.8B	34.97	60.19	47.59	66.40	70.62	62.83	63.57	58.02
OPT avg.	6.7B	42.52	50.46	44.36	58.18	54.78	44.43	53.13	49.69
OPT last.	6.7B	32.02	45.60	31.08	53.97	66.58	44.21	50.08	46.22
OPT prompt	6.7B	45.56	71.22	52.53	62.96	70.67	54.83	56.21	59.14
PromptEOL OPT	125M	59.90	71.55	60.93	70.76	72.83	67.89	65.14	67.00
	350M	54.70	71.52	59.99	64.51	71.39	66.55	66.58	65.03
	1.3B	64.59	79.06	68.46	78.88	78.64	73.22	69.41	73.18
	2.7B	60.03	75.51	64.30	74.56	77.62	67.73	65.35	69.30
	6.7B	60.91	80.05	67.65	75.49	80.11	72.91	67.57	72.10
	13B	60.21	81.36	69.69	75.46	79.58	70.73	65.99	71.86
	30B	59.99	80.52	69.80	75.20	78.03	73.57	69.87	72.43
	66B	55.66	74.62	64.90	72.34	75.21	71.72	67.43	68.84
PromptEOL+ICL OPT	125M	62.22	73.10	61.84	71.09	72.08	67.80	64.10	67.46
	350M	63.87	73.85	63.41	72.45	73.13	70.84	65.61	69.02
	1.3B	72.78	83.77	73.61	83.42	80.60	78.80	69.69	77.52
	2.7B	68.49	84.72	75.15	83.62	81.34	80.94	72.97	78.18
	6.7B	70.65	84.51	75.01	83.51	82.00	81.12	76.77	79.08
	13B	71.99	85.22	76.04	82.23	81.38	81.42	75.00	79.04
	30B	69.99	83.35	74.75	83.14	82.42	81.45	77.46	78.94
	66B	69.93	83.29	74.88	80.10	81.11	81.76	76.26	78.19

Table 1: Performances of our method on STS tasks without fine-tuning. ICL denotes in-context learning with our demonstration set. †: results from (Gao et al., 2021). ‡: results from (Jiang et al., 2022). §: results from (Ni et al., 2021). More results on other LLMs can be found in Appendix E.

4.2 Main Results

We compare our method with BERT-based methods such as SBERT (Reimers and Gurevych, 2019), SimCSE (Gao et al., 2021), and PromptBERT (Jiang et al., 2022). In addition, we include other sentence methods based on LLMs as baselines, such as ST5 (Ni et al., 2021) and SGPT (Muennighoff, 2022). Among these baselines, ST5 achieves state-of-the-art results on both STS and transfer learning tasks by further finetuning 4.8B parameters T5 encoder with contrastive learning.

STS tasks without fine-tuning Table 1 shows the results of PromptEOL with and without in-context learning on STS tasks. PromptEOL significantly outperforms other sentence representation methods by better leveraging the capabilities of LLMs to express sentence semantics. Compared to the previous prompt-based method, PromptEOL achieves more than a 13-point improvement in average Spearman correlation in the 6.7B OPT. In-context learning further improves the quality of sentence embeddings based on PromptEOL. It helps 6.7B OPT achieve 79.08 spearman correlation without fine-tuning, which significantly outperfoms the previous methods like ST5-Enc or BERT prompt. Moreover, it demonstrates that LLMs without any fine tuning have great potential to represent sentences based on their semantics into embeddings for retrieval purposes.

STS tasks with fine-tuning Table 2 shows the results by fine-tuning with PromptEOL on the supervised dataset. Compared to ST5-Enc, which finetuned all 4.8B parameters on Community QA and NLI datasets, our method with 2.7B OPT achieves superior results through parameter-efficient fine tuning on the 4-bit model with only NLI datasets. Keep scaling up the parameters size, 30B LLaMA achieve the best performance on STS tasks, attaining a Spearman correlation of 86.24 on STS tasks. Moreover, we also report the results of LLaMA-2 (Touvron et al., 2023b) on Appendix C and observe it performs better performance than LLaMA.

For the alignment method, we fine-tune the 7B, 13B, and 30B LLaMA models with the same data. Our alignment method enhances

Method	Params	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Fine-tuning on supervised datasets									
SimCSE-RoBERTa†	123M	76.53	85.21	80.95	86.03	82.57	85.83	80.50	82.52
PromptRoBERTa‡	123M	76.75	85.93	82.28	86.69	82.80	86.14	80.04	82.95
SGPT¶	5.8B	74.28	85.35	79.21	85.52	82.54	85.50	79.53	81.70
ST5-Enc§	4.8B	80.10	88.75	84.70	88.86	85.17	86.77	80.39	84.96
PromptEOL+CSE OPT	1.3B	79.01	89.26	84.10	88.30	84.62	87.71	80.52	84.79
	2.7B	79.49	89.64	84.80	89.51	85.91	88.33	81.64	85.62
	6.7B	80.14	90.02	84.94	89.78	85.84	88.75	81.29	85.82
	13B	80.20	90.24	85.34	89.52	85.90	88.56	82.06	85.97
PromptEOL+CSE LLaMA	7B	79.16	90.22	85.40	88.99	86.25	88.37	81.51	85.70
	13B	78.63	90.03	85.46	89.48	86.18	88.45	82.69	85.85
	30B	79.72	90.25	85.85	90.04	86.27	89.14	82.38	86.24
PromptEOL+Align LLaMA	7B	79.75	90.73	86.14	89.35	86.93	88.39	82.84	86.30
	13B	79.49	90.34	86.00	89.71	86.86	88.38	83.46	86.32
	30B	80.17	91.03	86.78	90.15	87.16	89.10	82.93	86.76

Table 2: Performances of our method on STS tasks with fine-tuning. CSE denotes contrastive learning for sentence embeddings. †: results from (Gao et al., 2021). §: results from (Ni et al., 2021). ¶: results from evaluation the public checkpoint (Muennighoff, 2022) on STS tasks.

Method	Params	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
Fine-tuning on supervised datasets									
SimCSE-RoBERTa†	123M	84.92	92.00	94.11	89.82	91.27	88.80	75.65	88.08
PromptRoBERTa‡	123M	85.74	91.47	94.81	90.93	92.53	90.40	77.10	89.00
ST5-Enc§	4.8B	90.83	94.44	96.33	91.68	94.84	95.40	77.91	91.63
Without fine-tuning									
BERT avg.	110M	78.66	86.25	94.37	88.66	84.40	92.80	69.54	84.94
ST5-Enc§	4.8B	91.15	93.33	97.55	90.20	94.07	94.40	74.26	90.71
PromptEOL OPT	1.3B	88.06	91.55	95.90	91.55	93.08	95.00	73.97	89.87
	2.7B	88.83	92.29	95.93	91.76	94.62	96.00	75.94	90.77
	6.7B	90.26	92.50	96.67	91.39	94.67	96.00	77.91	91.34
	13B	90.73	92.90	96.69	91.48	94.01	96.80	75.59	91.17
	30B	90.95	92.77	96.99	91.79	95.28	97.00	73.97	91.25
	66B	90.96	93.40	97.01	91.93	95.22	96.40	75.25	91.45
PromptEOL LLaMA	7B	90.40	92.90	96.88	91.57	95.11	95.40	75.13	91.06
	13B	92.02	93.22	97.29	91.40	95.66	95.80	76.46	91.69
	30B	91.64	93.27	97.10	91.86	95.99	95.80	78.43	92.01
	65B	92.13	93.43	97.16	91.91	95.33	97.40	77.28	92.09

Table 3: Performances of our method on transfer learning tasks. †: results from (Gao et al., 2021). ‡: results from (Jiang et al., 2022). §: results from (Ni et al., 2021).

the performance of all models on STS tasks. Even though PromptEOL+CSE already outperforms other methods, our alignment method still provides additional improvements.

Transfer tasks We report the results of our method on the transfer learning tasks in Table 3. We observe that LLMs achieve better performance as the model size increases. Specifically, the 66B OPT and 65B LLaMA models outperform their smaller counterparts with PromptEOL.

5 Analysis

5.1 Sentence Representation Methods

We present the results obtained using three sentence representation methods, across models ranging in size from 125M to 66B parameters, as shown in Figure 5. Different representation

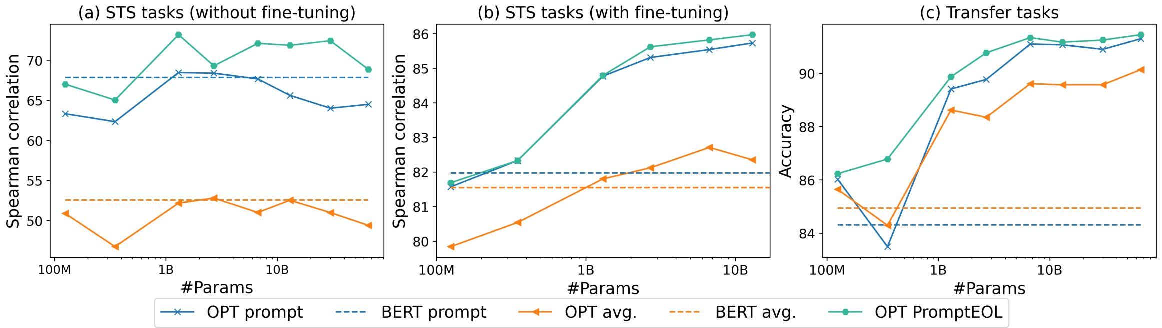

Figure 5: Influence of different sentence representation methods on three settings. "avg." refers to use averaging output tokens as sentence embeddings. "prompt" refers to extract sentence embeddings using the template from (Jiang et al., 2022) . Dash lines represent the results from the base-size BERT.

methods can yield significantly different results. Prompt-based methods outperform direct averaging in three settings. Among these methods, PromptEOL exhibits the best performance, as it introduces an explicit "one-word limitation". More detail results can be found in Appendix D.

5.2 Scaling on Sentence Embeddings

Scaling up the model size can significantly improve the performance of sentence embeddings, as shown in Table 2 and 3. But we also notice that the STS performance without fine-tuning is not scaling with the model size, as shown in Table 1.

Consider the STS tasks, which require sentence embeddings to satisfy two criteria: first, they must contain the semantic information of the sentence; second, semantically similar sentences should have small distances in the embedding space. For the first criterion, we observe that larger models achieve better performance on transfer tasks, indicating that the embeddings from larger models can capture more information about the sentence. However, for the second criterion, scaling up can be counterproductive. To validate this point, we calculate the anisotropy of the sentence embeddings from different models, as shown in Figure 6. We find that the anisotropy of the sentence embeddings increases as the model size increases. This demonstrates that larger models exhibit more anisotropy in the embedding space, causing the embeddings to have smaller distances even with random sentences. For the setting with fine-tuning, we can use techniques such as contrastive learning to mitigate the anisotropy of the embeddings (Gao et al., 2021) and achieve better performance on STS tasks by scaling up. For the setting without fine-tuning, since we do not directly address anisotropy, the performance of larger models can sometimes be limited by the anisotropy of the embeddings.

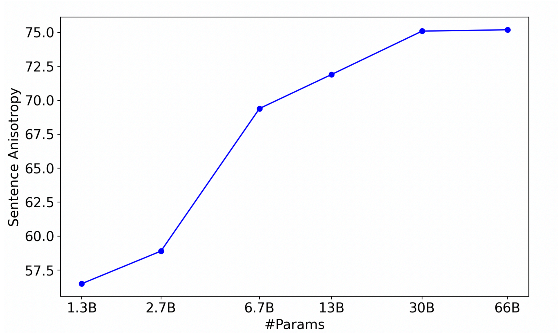

Figure 6: Anisotropy of sentence embeddings in different model sizes of OPT. The anisotropy is the average similarity of 100k random sentence pairs from the Wikipedia corpus.

6 Conclusion

In this paper, we focus on exploiting LLMs to improve sentence embeddings. To achieve this, we propose a new sentence embeddings method called PromptEOL, which adapts previous prompt-based methods to autoregression models. Furthermore, we leverage in-context learning to generate superior sentence embeddings by utilizing ChatGPT and the Oxford dictionary to create sentence embeddings demonstrations. It demonstrates in-context learning allows LLMs to achieve performance comparable to current contrastive learning methods. With our prompt-based method, we also discover that further fine-tuning of LLMs can achieve the state-of-the-art performance using only efficient fine-tuning methods.

7 Limitation

Despite LLMs with PromptEOL exhibiting robust performance, it typically demands more computational resources than smaller language models. Nevertheless, PromptEOL remains an efficient sentence embeddings method, which outperforms previous methods such as ST5 with significantly fewer model parameters and fine-tuning resources. Limited by the hardware, we only scale the LLMs to 30B parameters with QLoRA for the setting of fine-tuning. We expect that performance could be further enhanced with full fine-tuning or larger models.

References

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*, volume 33, pages 1877–1901. Curran Associates, Inc.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Qinyuan Cheng, Xiaogui Yang, Tianxiang Sun, Linyang Li, and Xipeng Qiu. 2023. Improving contrastive learning of sentence embeddings from ai feedback. *arXiv preprint arXiv:2305.01918*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljačić, Shang-Wen Li, Wen-tau Yih, Yoon Kim, and James Glass. 2022. Diffcse: Difference-based contrastive learning for sentence embeddings. *arXiv preprint arXiv:2204.10298*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. 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)*, pages 55–65.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Xinyang Geng and Hao Liu. 2023. Openllama: An open reproduction of llama.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

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. *arXiv preprint arXiv:2201.04337*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

MosaicML. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms. Accessed: 2023-05-05.

Niklas Muennighoff. 2022. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. 2021. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Teven Le Scao, 388 Authors, and Thomas Wolf. 2022. BLOOM: A 176B-parameter open-access multilingual language model. *ArXiv*, abs/2211.05100.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint 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 (Volume 2: Short Papers)*, pages 411–418, Online. Association for Computational Linguistics.

Qiyu Wu, Chongyang Tao, Tao Shen, Can Xu, Xiubo Geng, and Daxin Jiang. 2022a. Pcl: Peer-contrastive learning with diverse augmentations for unsupervised sentence embeddings. *arXiv preprint arXiv:2201.12093*.

Xing Wu, Chaochen Gao, Zijia Lin, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2022b. InfoCSE: Information-aggregated contrastive learning of sentence embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3060–3070, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Junlei Zhang, Zhenzhong Lan, and Junxian He. 2023. Contrastive learning of sentence embeddings from scratch. *arXiv preprint arXiv:2305.15077*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

A Influence of Quantization

We analyze the influence of quantization in Table 4 between the 16bit models and 4bit models, which are quantized by bitsandbytes [1] with 4-bit normalfloat and double quantization. We find large models tend to show better results on STS tasks after 4-bit quantization. For example, PromptEOL+ICL with 6.7B OPT improve Spearman correlation from 79.08 to 79.38.

Method	Params	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
PromptEOL OPT(16-bit)	125M	59.90	71.55	60.93	70.76	72.83	67.89	65.14	67.00
	350M	54.70	71.52	59.99	64.51	71.39	66.55	66.58	65.03
	1.3B	64.59	79.06	68.46	78.88	78.64	73.22	69.41	73.18
	2.7B	60.03	75.51	64.30	74.56	77.62	67.73	65.35	69.30
	6.7B	60.91	80.05	67.65	75.49	80.11	72.91	67.57	72.10
	13B	60.21	81.36	69.69	75.46	79.58	70.73	65.99	71.86
	30B	59.99	80.52	69.80	75.20	78.03	73.57	69.87	72.43
	66B	55.66	74.62	64.90	72.34	75.21	71.72	67.43	68.84
PromptEOL OPT(4-bit)	125M	60.53	70.03	59.02	69.77	72.38	66.47	65.17	66.20
	350M	58.03	72.61	61.34	66.14	72.99	67.27	65.10	66.21
	1.3B	63.72	79.32	68.13	77.92	78.56	72.03	68.80	72.64
	2.7B	57.80	72.45	61.09	73.33	76.22	64.71	64.07	67.10
	6.7B	63.81	81.45	69.90	77.68	80.92	75.51	69.28	74.08
	13B	60.91	80.97	70.22	76.93	79.46	72.84	66.34	72.52
	30B	59.33	79.65	69.25	73.87	77.79	71.72	69.07	71.53
	66B	59.35	77.33	68.33	74.45	77.25	73.93	69.27	71.42
PromptEOL+ICL OPT(16-bit)	125M	62.22	73.10	61.84	71.09	72.08	67.80	64.10	67.46
	350M	63.87	73.85	63.41	72.45	73.13	70.84	65.61	69.02
	1.3B	72.78	83.77	73.61	83.42	80.60	78.80	69.69	77.52
	2.7B	68.49	84.72	75.15	83.62	81.34	80.94	72.97	78.18
	6.7B	70.65	84.51	75.01	83.51	82.00	81.12	76.77	79.08
	13B	71.99	85.22	76.04	82.23	81.38	81.42	75.00	79.04
	30B	69.99	83.35	74.75	83.14	82.42	81.45	77.46	78.94
	66B	69.93	83.29	74.88	80.10	81.11	81.76	76.26	78.19
PromptEOL+ICL OPT(4-bit)	125M	61.02	71.00	59.75	69.67	70.52	65.14	63.45	65.79
	350M	64.14	72.45	62.58	71.05	70.18	67.67	65.52	67.66
	1.3B	73.45	82.55	73.11	83.63	80.60	78.72	69.06	77.30
	2.7B	68.50	84.73	74.62	82.23	80.87	80.81	72.30	77.72
	6.7B	70.23	84.64	76.08	83.73	82.06	81.66	77.29	79.38
	13B	71.79	84.23	75.57	81.75	80.71	80.89	74.46	78.49
	30B	70.61	84.05	75.27	83.23	82.77	81.45	77.31	79.24
	66B	71.67	83.95	75.67	81.33	81.86	82.58	76.54	79.09

Table 4: Influence of quantization on STS tasks. ICL denotes in-context learning with our demonstration set.

[1] https://github.com/TimDettmers/bitsandbytes

B Training Details

We use QLoRA to fine-tune OPT and LLaMA with contrastive learning. Following QLoRA, we use LoRA $r = 64, \alpha = 16$, dropout $= 0.05$, and add LoRA modules on all linear layers of the 4-bit quantized model. We fine-tune models on the NLI datasets (Gao et al., 2021) with one epoch, temperature $\tau = 0.05$ and learning rate 5e-4. We report the training time and inferencee time in Table 5.

Model size	Train (minutes)	Inference (ms/sample)
SimCSE (110M)	8	0.57
350M	11	1.09
1.3B	22	1.31
2.7B	40	2.55
6.7B	75	8.07
13B	150	14.53

Table 5: Training time and inference time of different model sizes. Inference time is measured on a single GPU with 8 batch size.

C Results of PromptEOL+CSE on LLaMA2

We report the results on LLaMA-2(Touvron et al., 2023b) on Table 6.

Method	**Params**	**STS12**	**STS13**	**STS14**	**STS15**	**STS16**	**STS-B**	**SICK-R**	**Avg.**
PromptEOL+CSE	7B	79.16	90.22	85.40	88.99	86.25	88.37	81.51	85.70
LLaMA	13B	78.63	90.03	85.46	89.48	86.18	88.45	82.69	85.85
PromptEOL+CSE	7B	78.48	90.07	84.86	89.43	86.16	88.44	83.20	85.81
LLaMA-2	13B	78.84	90.35	85.88	89.72	86.68	88.91	82.64	86.15

Table 6: Influence of quantization on STS tasks. ICL denotes in-context learning with our demonstration set.

D Sentence Representation Methods

We supplemented detail results in Table 7 and 8 for different sentence representation methods.

Method	Params	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Without fine-tuning									
OPT avg.	125M	44.27	50.38	44.95	62.39	55.52	45.39	53.24	50.88
	350M	40.61	47.25	40.45	55.12	55.57	40.53	47.66	46.74
	1.3B	45.12	54.01	46.52	62.94	55.96	46.31	54.32	52.17
	2.7B	44.11	54.35	47.89	63.91	57.02	47.85	54.44	52.80
	6.7B	43.61	51.69	45.86	60.11	55.41	45.42	54.93	51.00
	13B	46.95	54.92	48.74	60.13	54.96	48.07	53.93	52.53
	30B	43.93	52.44	46.04	58.80	55.15	47.13	53.46	50.99
	66B	40.81	47.98	44.21	59.37	56.37	43.80	53.19	49.39
OPT prompt	125M	56.25	71.61	58.62	63.47	70.29	59.77	63.23	63.32
	350M	56.56	69.27	55.81	60.05	68.73	61.75	64.15	62.33
	1.3B	60.26	75.64	62.93	70.63	76.52	67.31	65.95	68.46
	2.7B	59.34	75.47	62.64	69.76	75.65	68.35	67.48	68.38
	6.7B	55.20	76.91	62.53	69.41	76.39	67.33	65.86	67.66
	13B	49.60	75.43	61.58	67.33	75.53	65.98	63.79	65.61
	30B	46.69	72.42	58.00	67.52	72.98	64.77	65.66	64.01
	66B	50.21	69.65	56.78	70.20	73.37	64.31	66.93	64.49
PromptEOL OPT	125M	59.90	71.55	60.93	70.76	72.83	67.89	65.14	67.00
	350M	54.70	71.52	59.99	64.51	71.39	66.55	66.58	65.03
	1.3B	64.59	79.06	68.46	78.88	78.64	73.22	69.41	73.18
	2.7B	60.03	75.51	64.30	74.56	77.62	67.73	65.35	69.30
	6.7B	60.91	80.05	67.65	75.49	80.11	72.91	67.57	72.10
	13B	60.21	81.36	69.69	75.46	79.58	70.73	65.99	71.86
	30B	59.99	80.52	69.80	75.20	78.03	73.57	69.87	72.43
	66B	55.66	74.62	64.90	72.34	75.21	71.72	67.43	68.84
Fine-tuning on unsupervised datasets									
OPT avg.	125M	74.08	82.70	77.76	83.65	79.74	82.43	78.55	79.84
	350M	74.07	83.78	78.06	84.62	80.70	83.93	78.61	80.54
	1.3B	75.38	84.99	80.34	86.10	81.49	84.35	79.98	81.80
	2.7B	75.31	85.66	80.73	86.71	81.84	84.92	79.66	82.12
	6.7B	76.02	86.22	81.30	87.07	82.54	85.28	80.53	82.71
	13B	75.86	86.32	80.73	86.25	82.13	85.55	79.62	82.35
OPT prompt	125M	76.05	85.24	79.82	85.27	81.30	84.56	79.09	81.62
	350M	76.28	86.01	80.96	86.13	81.87	85.33	79.73	82.33
	1.3B	78.56	89.21	84.21	88.71	84.17	87.39	81.16	84.77
	2.7B	78.89	89.21	84.43	89.43	85.75	88.07	81.40	85.31
	6.7B	78.66	89.81	84.45	89.70	85.71	88.63	81.79	85.54
	13B	79.66	89.84	84.88	89.54	85.59	88.65	81.93	85.73
PromptEOL OPT	125M	76.53	85.56	79.75	85.43	81.17	84.32	79.04	81.69
	350M	75.96	85.51	81.32	86.50	81.42	85.24	80.35	82.33
	1.3B	79.01	89.26	84.10	88.30	84.62	87.71	80.52	84.79
	2.7B	79.49	89.64	84.80	89.51	85.91	88.33	81.64	85.62
	6.7B	80.14	90.02	84.94	89.78	85.84	88.75	81.29	85.82
	13B	80.20	90.24	85.34	89.52	85.90	88.56	82.06	85.97

Table 7: Comparison of three sentence representation methods on STS tasks.

Method	**Params**	**MR**	**CR**	**SUBJ**	**MPQA**	**SST**	**TREC**	**MRPC**	**Avg.**
OPT avg.	125M	80.63	86.41	93.91	87.85	86.22	92.60	71.83	85.64
	350M	80.73	85.16	93.42	87.26	86.11	87.80	69.57	84.29
	1.3B	85.89	90.04	95.71	90.10	91.38	94.20	72.99	88.62
	2.7B	87.55	90.76	95.78	90.26	91.71	94.40	68.00	88.35
	6.7B	87.93	91.07	96.58	90.65	92.70	96.20	72.17	89.61
	13B	88.33	91.76	96.74	90.78	93.25	95.20	70.90	89.57
	30B	88.54	92.11	96.85	90.61	93.74	94.40	70.72	89.57
	66B	89.17	92.00	96.86	90.80	94.67	96.40	71.07	90.14
OPT prompt	125M	83.54	87.60	94.28	89.36	88.74	91.60	67.01	86.02
	350M	80.99	84.08	93.30	89.38	86.88	88.80	60.99	83.49
	1.3B	87.31	90.68	95.73	91.30	93.47	94.40	72.99	89.41
	2.7B	88.58	91.60	96.22	91.36	93.90	95.80	70.96	89.77
	6.7B	90.55	92.21	97.09	91.31	95.06	96.60	74.90	91.10
	13B	90.45	92.66	96.85	91.57	95.44	96.00	74.55	91.07
	30B	90.56	92.79	97.28	91.93	94.78	96.00	72.93	90.90
	66B	90.95	92.48	97.27	91.72	95.55	95.80	75.30	91.30
PromptEOL OPT	125M	80.86	87.66	93.19	89.77	87.31	92.20	72.64	86.23
	350M	84.14	88.08	93.17	89.77	89.73	91.20	71.36	86.78
	1.3B	88.06	91.55	95.90	91.55	93.08	95.00	73.97	89.87
	2.7B	88.83	92.29	95.93	91.76	94.62	96.00	75.94	90.77
	6.7B	90.26	92.50	96.67	91.39	94.67	96.00	77.91	91.34
	13B	90.73	92.90	96.69	91.48	94.01	96.80	75.59	91.17
	30B	90.95	92.77	96.99	91.79	95.28	97.00	73.97	91.25
	66B	90.96	93.40	97.01	91.93	95.22	96.40	75.25	91.45

Table 8: Comparison of three sentence representation methods on STS tasks.

E Resulst of PromptEOL and PromptEOL+ICL on Current Popular LLMs

We supplemented results of STS tasks with PromptEOL and PromptEOL+ICL in Table 9 on current popular LLMs include open-LLaMA (Geng and Liu, 2023), LLaMA (Touvron et al., 2023a), LLaMA-2 (Touvron et al., 2023b), MPT (MosaicML, 2023), Mistral (Jiang et al., 2023).

Params	Avg.	Prompt	PromptEOL	PromptEOL+ICL
Open-LLaMA				
3B	51.75	66.45	68.22	78.85
7B	52.03	63.40	76.35	79.17
13B	49.58	64.11	70.03	78.04
LLaMA				
7B	46.94	42.18	68.76	77.63
13B	47.53	48.73	65.62	73.40
30B	50.70	47.10	70.60	77.61
65B	44.80	51.69	69.39	75.73
LLaMA-2				
7B	46.34	45.87	69.30	75.99
13B	49.07	58.80	68.87	78.31
70B	44.34	45.14	70.90	74.97
MPT				
7B	49.39	57.25	71.06	79.08
30B	42.31	54.45	71.08	75.74
Mistral				
7B	49.32	66.23	73.32	78.35

Table 9: Results of PromptEOL and PromptEOL+ICL on current popular LLMs. We report averaging Spearman correlation over seven STS tasks with four sentence representation methods: avg., prompt, PromptEOL and PromptEOL+ICL. “Avg.” refers to use averaging output tokens as sentence embeddings. “Prompt” refers to extract sentencne embeddings using the template from (Jiang et al., 2022). For simplicity, we do not search demonstration for PromptEOL+ICL but use the best demonstration from the PromptEOL+ICL OPT directly. We expect that PromptEOL+ICL can achieve better results by searching for demonstrations according to the model.