

# Causal2Vec: Improving Decoder-only LLMs as Versatile Embedding Models

Ailiang Lin<sup>1\*</sup>, Zhuoyun Li<sup>2\*</sup>, Kotaro Funakoshi<sup>1</sup>

<sup>1</sup>Institute of Science Tokyo <sup>2</sup>Sun Yat-sen University  
 {linailiang, funakoshi}@lir.first.iir.isct.ac.jp  
 lizhy356@mail2.sysu.edu.cn

## Abstract

Decoder-only large language models (LLMs) are increasingly used to build embedding models that effectively encode the semantic information of natural language texts into dense vector representations for various embedding tasks. However, many existing methods primarily focus on removing the causal attention mask in LLMs to enable bidirectional attention, potentially undermining the model’s ability to extract semantic information acquired during pretraining. Additionally, leading unidirectional approaches often rely on extra input text to overcome the inherent limitations of causal attention, inevitably increasing computational costs. In this work, we propose Causal2Vec, a general-purpose embedding model tailored to enhance the performance of decoder-only LLMs without altering their original architectures or introducing significant computational overhead. Specifically, we first employ a lightweight BERT-style model to pre-encode the input text into a single Contextual token, which is then prepended to the LLM’s input sequence, allowing each token to capture contextualized information even without attending to future tokens. Furthermore, to mitigate the recency bias introduced by last-token pooling and help LLMs better leverage the semantic information encoded in the Contextual token, we concatenate the last hidden states of Contextual and EOS tokens as the final text embedding. In practice, Causal2Vec achieves state-of-the-art performance on the Massive Text Embeddings Benchmark (MTEB) among models trained solely on publicly available retrieval datasets, while reducing the required sequence length by up to 85% and inference time by up to 82% compared to best-performing methods.

## 1 Introduction

Text embedding models encode natural language text into dense vector representations that capture contextual semantic information [1, 2], enabling a wide range of downstream natural language processing (NLP) tasks such as information retrieval, semantic textual similarity, and question answering [3, 4, 5]. Moreover, embedding-based retrievers play a crucial role in enhancing the capabilities of large language model (LLM)-based Retrieval-Augmented Generation (RAG) systems [6, 7].

For many years, pretrained language models based on encoder-only or encoder-decoder Transformer architectures, such as BERT [8], RoBERTa [9], and T5 [10], have been the dominant paradigm for building text embedding models. With recent advances in decoder-only LLMs, considerable efforts have focused on adapting these decoder-only architectures for embedding tasks. However, the use of causal attention in Transformer decoders leads to incomplete information encoding for each token except the last one, significantly limiting the model’s representational capacity. To address this issue, many existing LLM-based text embedding models [11, 12, 13] achieve bidirectional attention by

---

\*Equal Contribution

removing the causal attention mask, enabling each token to access the entire sequence and thereby generating rich contextualized representations. Despite notable progress, we argue that modifying the original attention mechanisms of LLMs leads to a pre-train/fine-tune attention mismatch, which may compromise the model’s ability to extract semantic information acquired during pretraining. Moreover, since attention implementations vary across different LLMs [14, 15], replacing them may require a deep understanding of the underlying codebase and substantial modifications, which could hinder the community’s adoption of LLMs for text embedding and limit their feasibility and reliability in real-world applications. Additionally, most leading causal attention-based methods [16, 17] compensate for missing contextual information in the original sequence by introducing additional textual input, which inevitably increases computational costs during training and inference, restricting their applicability in resource-constrained scenarios.

To derive text embeddings from the output hidden states of LLMs, there are two mainstream strategies: mean pooling and last-token pooling. Mean pooling involves averaging all token representations in the output sequence, whereas last-token pooling typically uses the last hidden state of the EOS token. Prior studies [18, 17] show that for approaches based on causal attention, (weighted) mean pooling is less effective than last-token pooling, since autoregressive modeling prevents earlier tokens from attending to future tokens, leading to biased aggregated representations. Consequently, last-token pooling is more commonly used in unidirectional models. However, recent works [16, 13] reveal that last-token pooling can be highly sensitive to noisy information due to its reliance on tokens near the end of sequence, hindering the model’s ability to learn robust representations.

Based on the above discussions, we propose Causal2Vec, a simple yet powerful causal attention-based embedding model that significantly enhances the text encoding capabilities of decoder-only LLMs, while circumventing the need to modify their original architectures or introduce significant computational overhead. Specifically, to address the representational bottleneck inherent in causal attention mechanism while preserving the LLMs’ ability to extract semantic information learned during pretraining, we first employ a lightweight, off-the-shelf bidirectional encoder to distill the contextual content of the input text into a single Contextual token, which is then aligned to the dimensionality of LLM’s word embedding space via a trainable MLP layer. By prepending this token to LLM’s input sequence, we enable each token to access contextualized information even under the constraints of causal attention mask, without switching to bidirectional attention or utilizing extra input text. Moreover, we concatenate the last hidden states of Contextual and EOS tokens as the final text embedding, effectively mitigating the recency bias introduced by last-token pooling and encouraging LLMs to better leverage the contextualized information encoded in the Contextual token. We conduct comprehensive experiments on the MTEB benchmark [5] by integrating Causal2Vec into three decoder-only LLMs with parameter sizes ranging from 1.3B to 7B (S-LLaMA-1.3B, LLaMA-2-7B, Mistral-7B). Evaluation across 56 datasets spanning 7 tasks demonstrates that our best model, Causal2Vec-Mistral-7B, achieves state-of-the-art (SOTA) embedding performance among models trained solely on publicly available retrieval data. Notably, compared to leading causal attention-based methods [16, 17], our model reduces the required sequence length by up to 85% and inference time by up to 82%. Furthermore, we empirically conduct extensive analysis to validate the effectiveness and necessity of the proposed mechanism. Overall, our empirical results demonstrate the effectiveness of decoder-only LLMs in producing high-quality contextualized text embeddings, underscoring the significant potential inherent in the models’ original causal attention mechanism and contributing to future advancements in this research area.

## 2 Related Work

### 2.1 Bidirectional Text Embedding Models

Text embeddings have been widely applied in a variety of downstream applications such as fact checking [19], similarity search [20], open-domain question answering [3], and retrieval-augmented generation (RAG) [6]. Over the past few years, embedding methods based on pretrained language models with bidirectional attention, such as BERT [8], RoBERTa [9], and T5 [10], have dominated text embedding tasks. Early notable approaches including SimCSE [21] and Sentence-T5 [22], are pretrained with a masked language modeling objective and finetuned in a contrastive manner with natural language inference (NLI) datasets. Later works like E5 [23] and GTE [24] further improve embedding performance through weakly supervised contrastive training on curated text pair datasets.

More recent methods [25, 26, 27] have shifted toward developing general-purpose embedding models through task instructions, demonstrating strong generalization to unseen tasks.

## 2.2 Decoder-only LLM-Based Text Embedding Models

Since LLMs represent the most advanced language models available and have demonstrated excellent performance across a wide range of language tasks, recent research has increasingly focused on developing embedding models based on decoder-only architectures. Luo et al. [28] highlight that larger models with extensive pretraining consistently improve embedding performance, showing the effectiveness of leveraging LLMs as backbone models for embedding tasks. An intuitive approach to converting decoder-only LLMs into text encoders is to utilize the last hidden state of the EOS token under causal attention as text embedding. Ma et al. [29] finetuned LLaMA-2 to serve as both a dense retriever and a point-wise reranker, demonstrating that LLMs can indeed outperform smaller models in retrieval tasks. Llama2Vec [30] is an unsupervised learning method that enhances dense retrieval by reconstructing the input sentence and leveraging text embeddings to predict the next sentence. Despite these advances, decoder-only LLM-based embedding methods still suffer from inherent architectural drawbacks: causal attention prevents each token from interacting with subsequent tokens, hindering the model’s ability to produce contextual representations. To address this limitation, LLM2Vec [11] transformed the LLM’s attention from unidirectional to bidirectional by removing the causal attention mask. Building upon this approach, NV-Embed [13] introduced a novel latent attention layer over the final hidden states, followed by mean pooling to generate higher-quality representations. Moreover, GRITLM [12] unifies embedding tasks and generative tasks via different attention mechanisms.

Notably, the aforementioned bidirectional LLM-based embedding methods involve modifications to the model architecture, which may not be compatible with various LLM backbones, thereby significantly limiting their generality in real-world applications. In contrast, some studies preserve the original causal attention mechanism while attempting to address the inherent limitation of decoder-only LLMs. SGPT [31] proposed a position-weighted mean pooling strategy that assigns higher weights to tokens at later positions, serving as an alternative to last-token pooling or regular mean pooling. E5-Mistral [32] finetuned decoder-only LLMs using synthetic data and task-specific instructions. ECHO [16] repeated the input twice in the autoregressive modeling paradigm, allowing the text embedding extracted from the repeated tokens to capture contextualized information. Additionally, PromptEOL [33] and bge-en-icl [17] enhance text embeddings by leveraging the in-context learning (ICL) capabilities of LLMs, augmenting the original input with task-specific examples to provide contextual information. Similarly, our work focuses on improving the text embedding quality of decoder-only LLMs by increasing the information density of each token within the sequence, thereby enabling the final EOS token to capture richer semantic information about the entire text. However, there are several key differences: 1) We employ a lightweight bidirectional encoder to generate the Contextual token without introducing additional overhead to LLMs, whereas ECHO and bge-en-icl suffer from significantly increased computational cost at both training and inference time. 2) We concatenate the last hidden states of Contextual and EOS tokens as the final text embedding, effectively mitigating the recency bias caused by last-token pooling and empowering LLMs to better understand the contextual information encoded in the Contextual token. As a result, our approach achieves state-of-the-art performance on the MTEB benchmark among models that train only on publicly available retrieval data, showing its simplicity and effectiveness in text embedding tasks.

## 3 Method

Figure 1 illustrates the overall pipeline of our proposed Causal2Vec in generating text representations. Given an input text, we first utilize the lightweight bidirectional encoder to produce a Contextual token. This token is concatenated with the word embeddings of a task-specific instruction, the input text, and the EOS token to form the final input sequence, which is then fed into the LLM for causal attention computation. The final text embedding is obtained by concatenating the output hidden states of Contextual and EOS tokens. We elaborate on the Contextual token and representation method in the following sections.

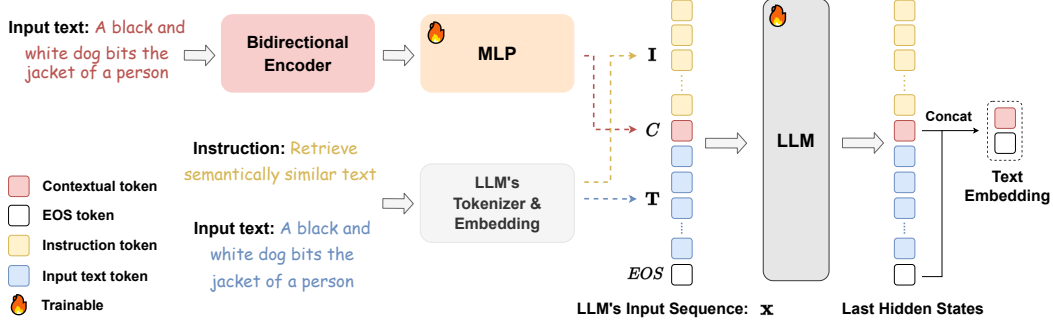


Figure 1: Overview of our proposed Causal2Vec method.

### 3.1 The Contextual Token

The remarkable capacity of large language models (LLMs) for human language understanding and generation is acquired through training on massive amounts of text data [34, 35, 36], showcasing their effectiveness in encoding semantic information. However, the inherent causal attention in LLMs prevents earlier tokens from accessing information about future tokens, thus limiting the model’s representational capacity. As a result, many LLM-based embedding models switch from unidirectional attention to bidirectional by removing the causal attention mask [11, 12, 13]. Although bidirectional attention facilitates effective information flow across the entire sentence, it introduces an attention mismatch between pretraining and finetuning, which somewhat compromises the LLMs’ ability to extract semantic information acquired during pretraining.

To fully unlock the potential of LLMs for text embedding, it is essential to address the limitations of causal attention while preserving LLMs’ ability to extract well-learned semantic information. To this end, we first introduce a lightweight BERT-style model that encodes the input text into a dense vector representation  $h \in \mathbb{R}^{1 \times k}$ , termed the "Contextual Token". Specifically, this token is generated by applying mean pooling over the last hidden states of the additional bidirectional model, capturing contextualized information about the entire input. Furthermore, to bridge the gap between the BERT-style model and LLM, we employ a simple MLP layer to match the dimensionality of the Contextual token with LLM’s word embedding space, and then encourage LLM to understand the sentence information encoded in this token through contrastive learning. Motivated by [37], the MLP layer consists of two linear transformations with a GELU activation  $\sigma$ , which can be formulated as:

$$C = \sigma(h\mathbf{W}_1^T)\mathbf{W}_2^T, \quad (1)$$

where  $\mathbf{W}_1 \in \mathbb{R}^{d \times k}$  and  $\mathbf{W}_2 \in \mathbb{R}^{d \times d}$  are trainable projection matrices, and  $C \in \mathbb{R}^{1 \times d}$  denotes the language embedding of the Contextual token, which shares the same dimensionality  $d$  as LLM’s word embedding space. Moreover, to leverage LLMs’ instruction-follow capabilities for producing general-purpose text embeddings, we use task-specific instructions for both training and evaluation, following [32, 11, 16]. Consequently, by adding the instruction, Contextual and EOS tokens to the start and end of input sequence, the resulting sequence fed into LLM can be constructed as:

$$\mathbf{x} = [\mathbf{I}; C; \mathbf{T}; EOS] \in \mathbb{R}^{(l+n+2) \times d}, \quad (2)$$

where  $[\cdot; \cdot]$  denotes the vertical concatenation operation,  $\mathbf{I} \in \mathbb{R}^{l \times d}$  and  $\mathbf{T} \in \mathbb{R}^{n \times d}$  represent the word embeddings of the task-specific instruction and input text, respectively. In this way, each token following the Contextual token  $C$  can capture contextualized information even without attending to future tokens. More importantly, the use of Contextual token requires no modifications to the model architecture, which not only preserves LLMs’ ability to extract semantic information learned during pretraining, but also enables seamless integration across different LLMs.

### 3.2 Representation Method

Last-token pooling typically utilizes the final hidden state of the EOS token as text embedding, and has been widely adopted in unidirectional models [18, 32, 17], since only the last EOS token captures information from the entire input. However, recent studies [16, 13] indicate that the EOS token embedding depends heavily on the hidden states of tokens near the end of sequence, leading to potential semantic bias in long-text scenarios.

To address this issue, we introduce a simple yet effective representation method tailored for the proposed embedding framework. Specifically, we concatenate the last hidden states of Contextual and EOS tokens as the final sentence representation for general-purpose embedding generation. Recognizing that the Contextual token has initially captured the semantic content of the input text, concatenating it with the EOS token yields a vector representation with richer contextualized content. Moreover, this approach enables explicit supervision of the Contextual token during training, thereby helping LLMs better understand the semantic information encoded in this added token.

The proposed representation method is optimized through supervised contrastive learning with the standard InfoNCE loss [38], which can be formulated as:

$$\mathcal{L} = -\log \frac{\exp(f(q, p^+)/\tau)}{\exp(f(q, p^+)/\tau) + \sum_{j=1}^N \exp(f(q, p_j^-)/\tau)}, \quad (3)$$

where  $f(q, p^+)$  represents the scoring function that computes cosine similarity between the query-passage pair embeddings from retrieval datasets,  $p^-$  denotes both in-batch and hard negative examples, and  $\tau$  is a temperature hyperparameter fixed at 0.05 in all experiments.

## 4 Experiments

### 4.1 Training Data

For training, we follow the mainstream practices [32, 11, 13, 17], utilizing a collection of publicly available retrieval datasets curated by [16], which consists of approximately 1.5M samples. More details about the training dataset composition can be found in Appendix A.3.

Notably, many previous works achieve strong performance by using extensive in-domain non-retrieval datasets from MTEB, private datasets, or proprietary synthetic data. To ensure fairness and consistency in comparisons, we only evaluate models trained on public retrieval datasets, enabling the verification of models’ generalization capability to unseen non-retrieval tasks, which serves as a critical criterion for defining a general-purpose embedding model.

### 4.2 Evaluation

For evaluation, we use the English subset of the Massive Text Embeddings Benchmark (MTEB) [5], which comprises 56 datasets spanning 7 embedding task categories, aiming to verify whether the model can produce universal text embeddings suitable for various embedding tasks. Specifically, the task categories include Retrieval (Retr.), Reranking (Rerank.), Clustering (Clust.), Pair Classification (PairClass.), Classification (Class.), Semantic Textual Similarity (STS), and Summarization (Summ.). The main evaluation metrics are nDCG@10, MAP, V-measure (V-meas.), average precision (AP), accuracy (Acc.), and Spearman correlation (Spear., both for STS and Summ.), respectively.

The full evaluation of a 7B parameter model on the MTEB benchmark is GPU resource intensive, requiring over 300 A100 80GB GPU hours. To speed up the evaluation, we follow [11] and [16]. Specifically, we introduce the MTEB-MINI for ablation studies and analysis, which combines 41 datasets covering all task categories in MTEB. Details of the MTEB-MINI composition can be found in Appendix B.1.

### 4.3 Implementation Details

For the base model, we integrate the proposed Causal2Vec with three decoder-only LLMs with parameter sizes ranging from 1.3B to 7B, including Sheared-LLaMA-1.3B (S-LLaMA-1.3B) [39], Llama-2-7B-chat (LLaMA-2-7B) [40], and Mistral-7B-Instruct-v0.2 (Mistral-7B) [41]. Regarding the off-the-shelf bidirectional encoder, we adopt E5-base-v2 [23], a lightweight model with only 110M parameters. All LLMs are finetuned using low-rank adaptation (LoRA) [42] on A100 80GB GPUs. In particular, LoRA is also applied to the bidirectional encoder when using the 1.3B LLM (refer to section 4.5.2 for the reason). Refer to Appendix A and B for further experimental details.

Table 1: Performance comparison on the full MTEB for models trained on publicly available retrieval data. Scores are averaged across each task category. S-LLaMA-1.3B, LLaMA-2-7B, and Mistral-7B refer to embedding methods built upon these decoder-only LLMs. The models grouped under *Miscellaneous* show reported scores from various recent works that use an encoder and/or a decoder other than the aforementioned LLMs. The best results are highlighted in **bold**, and the second-best are underlined. See Appendix C.5 for detailed results for each dataset.

Task (# of datasets) Metric	Retr. (15) nDCG@10	Rerank. (4) MAP	Clust. (11) V-Meas.	PairClass. (3) AP	Class. (12) Acc.	STS (10) Spear.	Summ. (1) Spear.	Avg (56)
<i>Miscellaneous</i>								
SGPT <sub>3.8B</sub> [31]	50.25	56.56	40.34	82.00	68.13	78.10	31.46	58.93
GTR <sub>xxl</sub> [43]	48.48	56.66	42.42	86.12	67.41	78.38	30.64	58.97
Sentence-T5 <sub>xxl</sub> [22]	42.24	56.42	43.72	85.07	73.42	82.63	30.08	59.51
UDEVER <sub>bloom-7b1</sub> [18]	49.34	55.91	40.81	85.40	72.13	83.01	30.97	60.63
Instructor <sub>xl</sub> [44]	49.26	57.29	44.74	86.62	73.12	83.06	<b>32.32</b>	61.79
BGE <sub>large-en-v1.5</sub> [45]	54.29	60.03	46.08	87.12	75.97	83.11	31.61	64.23
UAE <sub>large-v1</sub> [46]	54.66	59.88	46.73	87.25	75.58	84.54	<u>32.03</u>	64.64
S-LLaMA-1.3B								
LLM2Vec [11]	51.44	55.38	43.57	86.20	72.21	83.58	30.01	61.85
ECHO [16]	-	-	-	-	-	-	-	62.01
Causal2Vec	52.69	56.54	44.35	86.18	72.94	83.76	31.45	62.63
LLaMA-2-7B								
LLM2Vec [11]	54.60	57.38	45.24	88.03	76.33	83.73	28.49	64.14
Causal2Vec	55.28	58.18	47.23	87.85	75.95	84.90	31.09	64.94
Mistral-7B								
E5 <sub>Mistral-7b</sub> [32]	52.78	<u>60.38</u>	47.78	88.47	76.80	83.77	31.90	64.56
ECHO [16]	55.52	58.14	46.32	87.34	<u>77.43</u>	82.56	30.73	64.68
GRITLM [12]	53.10	<b>61.30</b>	<u>48.90</u>	86.90	77.00	82.80	29.40	64.70
LLM2Vec [11]	55.99	58.42	45.54	87.99	76.63	84.09	29.96	64.80
NV-Embed [13]	59.00	59.59	45.44	87.59	73.93	79.07	30.16	64.18
bge-en-icl (zero-shot) [17]	<u>59.59</u>	56.85	42.61	87.87	75.47	83.30	29.52	64.67
bge-en-icl (few-shot) [17]	<b>60.08</b>	56.67	46.55	<u>88.51</u>	77.31	83.69	30.68	66.08
Causal2Vec	57.28	59.46	48.89	88.43	76.41	<u>85.38</u>	30.57	<u>66.10</u>
Causal2Vec (w/ ICL)	57.48	59.36	<b>50.78</b>	<b>89.19</b>	<b>77.53</b>	<b>85.66</b>	30.82	<b>66.85</b>

#### 4.4 MTEB Results

We evaluate the proposed Causal2Vec against recent state-of-the-art text embedding methods on the full MTEB benchmark. It is worth noting that many existing approaches rely on non-public or synthetic data for training. Even among methods limited to disclosed data, many use substantial amounts of non-retrieval MTEB-related data, potentially leading to overfitting [17]. To ensure fair comparisons and better verify the model’s generalizability across diverse embedding tasks, we compare only against models trained on publicly available retrieval data. As shown in Table 1, Causal2Vec demonstrates consistently strong performance across various LLM backbones, with our best model, Causal2Vec-Mistral-7B, achieving state-of-the-art results on par with bge-en-icl [17]. Surprisingly, although LLaMA-2-7B has been shown to perform significantly worse than Mistral-7B on embedding tasks [12, 11], our Causal2Vec built upon LLaMA-2-7B surpasses most leading Mistral-7B-based methods, except for bge-en-icl (few-shot).

**Comparison to Bidirectional LLM-based Methods.** LLM2Vec [11] removes the causal attention mask to enable bidirectional attention, demonstrating competitive performance on the MTEB benchmark. NV-Embed [13] builds upon this approach by introducing a latent attention layer to obtain pooled embeddings, addressing the limitations of mean pooling. Notably, NV-Embed achieves significant improvements by incorporating additional non-retrieval MTEB-related and synthetic data. However, when trained solely on public retrieval data, NV-Embed shows lower performance compared to LLM2Vec. GRITLM [12] unifies embedding and generation training paradigms, yielding embedding performance on par with LLM2Vec. In contrast to these bidirectional LLM-based methods, our Causal2Vec requires no modifications to the model architecture, yet enables each token in the sequence to access contextual information through the introduced Contextual token. More importantly, our method consistently outperforms LLM2Vec across various base models under identical training data, achieving improvements of 0.78 points for S-LLaMA-1.3B, 0.80 for LLaMA-2-7B, and 1.30 for Mistral-7B. These results underscore that shifting from causal attention to bidirectional attention is not necessary for adopting LLMs to text embedding tasks—and may even compromise

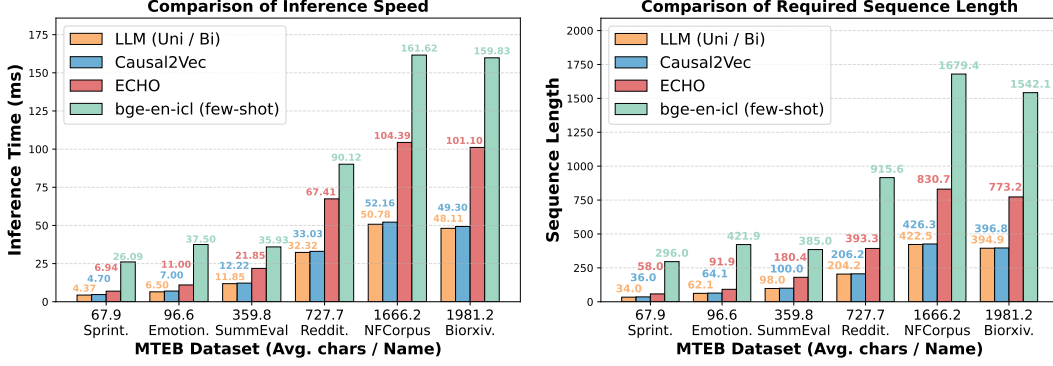


Figure 2: Average per-sample inference time (in milliseconds) and required sequence length of various Mistral-7B-based methods on selected MTEB subsets using a single A100 80GB GPU, including SprintDuplicateQuestions (Sprint.), RedditClusteringP2P (Reddit.), BiorxivClusteringP2P (Biorxiv.), etc. For the asymmetric dataset NFCorpus, we report the results per query-passage pair. LLM (Uni/Bi) denotes the standard Mistral-7B with causal or bidirectional attention.

the model’s ability to extract the well-learned semantic information. We argue that the effectiveness of bidirectional attention in capturing contextual information relies on maintaining consistency in attention mechanism throughout both pretraining and finetuning.

**Comparison to ECHO.** ECHO [16] repeats the input, allowing each token from the second occurrence to access the complete sequence content. However, this strategy comes with a significant drawback: it doubles the maximum sequence length, thus increasing computational cost during both training and inference. Conversely, our Causal2Vec enables contextual information access through a single Contextual token, without introducing additional overhead to the LLM. Furthermore, our proposed representation method, which concatenates the the last hidden states of Contextual and EOS tokens as final text embedding, effectively mitigates the recency bias inherent in last-token pooling used by ECHO and reduces reliance on the EOS token. As a result, Causal2Vec consistently outperforms ECHO on S-LLaMA-1.3B and Mistral-7B by 0.62 points and 1.42 points, confirming the effectiveness of our design in leveraging LLMs’ causal attention for text embedding.

**Comparison to bge-en-icl.** bge-en-icl [17] endows LLM-based embedding models with in-context learning (ICL) [47] capabilities by incorporating multiple task-related examples into the input. Our method significantly outperforms bge-en-icl (few-shot) on clustering, reranking, and STS tasks with an improvement of at least 1.69 points, while yielding comparable results on pair classification and summarization tasks. This indicates that our method achieves stronger overall performance on unseen non-retrieval tasks and generates universal text embeddings without relying on task-related demonstrations. More importantly, bge-en-icl relies on knowledge distillation from a reranker model [48] as the teacher—an effective strategy that is not adopted by other baselines. In addition, incorporating in-context examples substantially increases the computational burden of LLMs, especially given that the maximum sequence length of bge-en-icl can reach up to 2048 tokens, which is four times that of our method, significantly limiting its applicability in resource-constrained scenarios. Under the compute-matched setting, our model achieves an average score improvement of 1.43 over bge-en-icl (zero-shot). Moreover, bge-en-icl is sensitive to the selection strategy and number of in-context examples, hindering its applicability to different LLMs, especially for language models that lack in-context learning capabilities. Notably, our method is orthogonal to bge-en-icl. When equipped with the same ICL strategy, Causal2Vec (w/ ICL) achieves a state-of-the-art average score of **66.85**, outperforming bge-en-icl (few-shot) by 0.77 points.

**Efficiency.** In Figure 2, we report the approximate average inference time and required sequence length of different models on selected MTEB subsets. Specifically, ECHO and bge-en-icl inevitably increase inference time due to their extended sequence lengths. In contrast, since the additional bidirectional encoder contains only 110M parameters and a single Contextual token adds no computational burden to the LLM, Causal2Vec achieves inference speeds comparable to the standard Mistral-7B with causal or bidirectional attention, while reducing the required sequence length by up to **85%** (Sprint.: 34.0 vs. 269.0; Emotion.: 62.1 vs. 421.9) and inference time by up to **82%** (Sprint.: 4.37 vs. 26.09; Emotion.: 6.50 vs. 37.50) compared to bge-en-icl (few-shot).

Table 2: Performance comparison of different components on MTEB-MINI (41 datasets) using two base models: S-LLaMA-1.3B and Mistral-7B. CtxToken indicates adding the Contextual token to LLM’s input, while Concat denotes the proposed representation method that concatenates the last hidden states of LLM’s Contextual and EOS tokens as the text embedding.

CtxToken	Concat	Retr. (7) nDCG@10	Rerank. (3) MAP	Clust. (6) V-Meas.	PairClass. (3) AP	Class. (11) Acc.	STS (10) Spear.	Summ. (1) Spear.	Avg (41)
S-LLaMA-1.3B									
		46.19	62.99	36.32	85.68	69.91	83.70	30.54	64.00
✓		46.46	62.93	36.32	85.48	70.32	<b>84.13</b>	30.83	64.24
✓	✓	<b>47.18</b>	<b>64.48</b>	<b>36.61</b>	<b>86.18</b>	<b>71.14</b>	83.76	<b>31.45</b>	<b>64.72</b>
Mistral-7B									
		51.07	68.26	39.06	88.29	74.24	84.86	29.81	67.23
✓		51.15	68.32	<b>40.04</b>	88.14	<b>75.00</b>	84.92	29.82	67.60
✓	✓	<b>51.77</b>	<b>68.51</b>	39.99	<b>88.43</b>	74.72	<b>85.38</b>	<b>30.57</b>	<b>67.79</b>

Table 3: Average MTEB-MINI score (41 datasets) for Causal2Vec with and without LoRA applied to the bidirectional encoder (Bi-LoRA). Refer to Appendix C.3 for detailed results for each task category.

Method	S-LLaMA-1.3B	Mistral-7B
w/ Bi-LoRA	<b>64.72</b>	67.66
w/o Bi-LoRA	64.43	<b>67.79</b>

Table 4: Average MTEB-MINI score (41 datasets) for placing the Contextual token before and after the task-specific instruction in Causal2Vec. See Appendix C.4 for detailed results for each task category.

Method	S-LLaMA-1.3B	Mistral-7B
before instruction	64.63	67.63
after instruction	<b>64.72</b>	<b>67.79</b>

## 4.5 Ablation Studies

### 4.5.1 Effectiveness of Each Component

To evaluate the effectiveness of the proposed Contextual token and representation method, we conduct ablation studies on MTEB-MINI using two base models of different scales: S-LLaMA-1.3B and Mistral-7B. As shown in Table 2, incorporating the Contextual token into LLM’s causal attention mechanism yields average score improvements of 0.24 and 0.37 for S-LLaMA-1.3B and Mistral-7B, respectively. These results not only confirm the effectiveness of the Contextual token but also highlight its scalability and applicability across different LLMs. We attribute these performance improvements to the rich contextualized content encoded in the Contextual token, which allows preceding tokens in the sequence to access accurate sentence information even without attending to future tokens, thereby mitigating the inherent architectural limitation in causal attention.

By concatenating the last hidden states of Contextual and EOS tokens as the final vector representation, we observe consistent performance improvements across all seven tasks, with average score gains of 0.72 and 0.56 on S-LLaMA-1.3B and Mistral-7B compared to standard LLMs, respectively. These results further confirm the generalizability and robustness of our method in enhancing embedding quality. We attribute the effectiveness of the proposed representation method to the following reasons: (1) it effectively alleviates the recency bias, as the Contextual token is not influenced by tokens near the end of sequence; (2) it enables explicit supervision of the Contextual token during training; and (3) the concatenation of two context-aware tokens enriches the semantic information of the final text embedding. See Appendix C.2 for further discussion on different representation methods.

### 4.5.2 Impact of Freezing the Bidirectional Encoder

Since the BERT-style bidirectional encoder (E5-base-v2) we use is specifically trained for embedding tasks, this section investigates whether it should be frozen during finetuning. As shown in Table 3, we observe that finetuning the bidirectional encoder with LoRA leads to an average score improvement of 0.29 on the MTEB-MINI for S-LLaMA-1.3B, but degrades Mistral-7B’s performance by 0.13 points. We attribute this to two potential effects of making the bidirectional encoder trainable: (1) it may cause catastrophic forgetting in the bidirectional encoder, and (2) it may help the LLM better interpret the Contextual token through joint finetuning. Large-scale LLMs are more susceptible



Table 5: Performance comparison of Causal2Vec-S-LLaMA-1.3B using different numbers of Contextual tokens on the full MTEB (56 datasets). CtxToken denotes the Contextual token. Note: Causal2Vec uses a single Contextual token by default.

Task (# of datasets) Metric	Retr. (15) nDCG@10	Rerank. (4) MAP	Clust. (11) V-Meas.	PairClass. (3) AP	Class. (12) Acc.	STS (10) Spear.	Summ. (1) Spear.	Avg (56)
Causal2Vec	<b>52.69</b>	<b>56.54</b>	44.35	<b>86.18</b>	<b>72.94</b>	83.76	31.45	<b>62.63</b>
w/ 2 CtxTokens	52.48	56.27	44.37	86.06	72.43	<b>83.90</b>	31.60	62.47
w/ 4 CtxTokens	52.13	56.32	<b>44.46</b>	86.04	72.68	83.74	<b>31.73</b>	62.42
w/ 8 CtxTokens	52.46	56.28	44.44	85.99	72.56	83.75	30.67	62.46

Table 6: Performance comparison of Causal2Vec-S-LLaMA-1.3B using different bidirectional encoders (Bi-Encoders) on the full MTEB (56 datasets). Note: E5-base-v2 is used as the default bidirectional encoder in Causal2Vec.

Task (# of datasets) Metric	Retr. (15) nDCG@10	Rerank. (4) MAP	Clust. (11) V-Meas.	PairClass. (3) AP	Class. (12) Acc.	STS (10) Spear.	Summ. (1) Spear.	Avg (56)
BERT-base [8]	10.59	43.44	30.12	61.66	60.78	54.36	29.82	38.33
E5-small-v2 [23]	49.04	54.32	39.92	84.67	72.94	80.39	31.16	59.93
E5-base-v2 [23]	50.30	55.91	44.10	85.63	71.15	81.03	30.28	60.97
Causal2Vec	<b>52.69</b>	<b>56.54</b>	<b>44.35</b>	<b>86.18</b>	<b>72.94</b>	83.76	<b>31.45</b>	<b>62.63</b>
w/ E5-small-v2	52.45	55.82	44.15	86.08	72.65	<b>83.92</b>	31.24	62.43
w/ BERT-base	52.02	55.73	43.60	85.87	71.70	83.86	30.99	61.97
w/o Bi-Encoder	51.75	55.53	43.52	85.68	71.31	83.70	30.54	61.74

to the former, as they already have sufficient capacity to comprehend newly added tokens during finetuning [49, 50]. This suggests that whether the introduced bidirectional encoder should remain frozen may depend on the scale of the underlying LLM.

#### 4.5.3 The Position of Contextual Token

We investigate whether the Contextual token’s position affects embedding performance by comparing two placement settings: before vs. after the instruction. As shown in Table 4, "after instruction" consistently yields better results. We speculate that positioning the Contextual token before instruction tokens may hinder the LLM’s ability to accurately interpret and follow task-specific prompts.

#### 4.5.4 The Number of Contextual Tokens

To examine the impact of using multiple Contextual tokens, we adopt cross-attention with a set of learnable queries to extract a fixed number of Contextual tokens from the bidirectional encoder, following [51, 52]. As presented in Table 5, increasing the number of Contextual tokens leads to performance degradation. We hypothesize that the additional tokens fail to provide more distinctive semantic information and instead introduce redundancy. These findings suggest that a single Contextual token is sufficient to supply the missing contextual information under causal attention, while maintaining model simplicity and efficiency.

#### 4.5.5 Impact of Different Bidirectional Encoders

Finally, we explore the impact of different bidirectional encoders on MTEB-MINI. We conduct experiments using S-LLaMA-1.3B, with all bidirectional encoders being trainable to facilitate LLM’s adaptation to the Contextual token through joint finetuning. As shown in Table 6, incorporating even the standard BERT [8] enhances embedding performance over the baseline LLM using last-token pooling (w/o Bi-Encoder). Moreover, the better-performing bidirectional encoder (e.g., E5-base vs. E5-small) generates a Contextual token with richer semantic information, enabling LLMs to capture more contextual semantics and thereby improve text embedding quality. It is important to note that although the E5-series encoders are specifically trained for embedding tasks, their performance on MTEB remains limited. This indicates that the performance improvements achieved by our method primarily stem from addressing the inherent shortcomings of causal attention and last-token pooling.

## 5 Conclusion

This paper presents Causal2Vec, a simple yet powerful text embedding model built upon decoder-only LLMs. It requires no architectural modifications or additional input text, achieving consistently strong performance in general-purpose embedding tasks. By introducing the proposed Contextual token, we enable each token in the sequence to capture contextual information within the inherent autoregressive modeling paradigm. To address the limitations of last-token pooling commonly used in unidirectional models, we propose a specialized representation method that concatenates the last hidden states of Contextual and EOS tokens as the final text embedding. Experimental results demonstrate that Causal2Vec not only achieves state-of-the-art performance on the MTEB benchmark, but also significantly reduces the required sequence length and inference time compared to best-performing methods.

## References

- [1] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, volume 28, 2015.
- [2] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, 2017.
- [3] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, 2020.
- [4] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021.
- [5] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, 2023.
- [6] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.
- [7] Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. ChatQA: Surpassing GPT-4 on conversational QA and RAG. In *Advances in Neural Information Processing Systems*, 2024.
- [8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.
- [9] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [10] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
- [11] Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. LLM2vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*, 2024.

- [12] Niklas Muennighoff, SU Hongjin, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. In *ICLR 2024 Workshop: How Far Are We From AGI*, 2024.
- [13] Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. NV-embed: Improved techniques for training LLMs as generalist embedding models. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [14] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. 2023.
- [15] MiniMax, Aonian Li, Bangwei Gong, Bo Yang, Boji Shan, Chang Liu, Cheng Zhu, Chunhao Zhang, Congchao Guo, Da Chen, Dong Li, Enwei Jiao, Gengxin Li, Guojun Zhang, Haohai Sun, Houze Dong, Jiadai Zhu, Jiaqi Zhuang, Jiayuan Song, Jin Zhu, Jingtao Han, Jingyang Li, Junbin Xie, Junhao Xu, Junjie Yan, Kaishun Zhang, Kecheng Xiao, Kexi Kang, Le Han, Leyang Wang, Lianfei Yu, Liheng Feng, Lin Zheng, Linbo Chai, Long Xing, Meizhi Ju, Mingyuan Chi, Mozhi Zhang, Peikai Huang, Pengcheng Niu, Pengfei Li, Pengyu Zhao, Qi Yang, Qidi Xu, Qixiang Wang, Qin Wang, Qiuhui Li, Ruitao Leng, Shengmin Shi, Shuqi Yu, Sichen Li, Songquan Zhu, Tao Huang, Tianrun Liang, Weigao Sun, Weixuan Sun, Weiyu Cheng, Wenkai Li, Xiangjun Song, Xiao Su, Xiaodong Han, Xinjie Zhang, Xinzhu Hou, Xu Min, Xun Zou, Xuyang Shen, Yan Gong, Yingjie Zhu, Yipeng Zhou, Yiran Zhong, Yongyi Hu, Yuanxiang Fan, Yue Yu, Yufeng Yang, Yuhao Li, Yunan Huang, Yunji Li, Yunpeng Huang, Yunzhi Xu, Yuxin Mao, Zehan Li, Zekang Li, Zewei Tao, Zewen Ying, Zhaoyang Cong, Zhen Qin, Zhenhua Fan, Zhihang Yu, Zhuo Jiang, and Zijia Wu. Minimax-01: Scaling foundation models with lightning attention. *arXiv preprint arXiv:2501.08313*, 2025.
- [16] Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. Repetition improves language model embeddings. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [17] Chaofan Li, Minghao Qin, Shitao Xiao, Jianlyu Chen, Kun Luo, Defu Lian, Yingxia Shao, and Zheng Liu. Making text embedders few-shot learners. In *The Thirteenth International Conference on Learning Representations*, 2025.
- [18] Xin Zhang, Zehan Li, Yanzhao Zhang, Dingkun Long, Pengjun Xie, Meishan Zhang, and Min Zhang. Language models are universal embedders. *arXiv preprint arXiv:2310.08232*, 2023.
- [19] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 809–819, 2018.
- [20] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.
- [21] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, 2021.
- [22] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 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, 2022.
- [23] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.
- [24] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

- [25] Akari Asai, Timo Schick, Patrick Lewis, Xilun Chen, Gautier Izacard, Sebastian Riedel, Hannaneh Hajishirzi, and Wen-tau Yih. Task-aware retrieval with instructions. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3650–3675, 2023.
- [26] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, 2023.
- [27] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, 2024.
- [28] Kun Luo, Minghao Qin, Zheng Liu, Shitao Xiao, Jun Zhao, and Kang Liu. Large language models as foundations for next-gen dense retrieval: A comprehensive empirical assessment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1354–1365, 2024.
- [29] Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. Fine-tuning llama for multi-stage text retrieval. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2421–2425, 2024.
- [30] Zheng Liu, Chaofan Li, Shitao Xiao, Yingxia Shao, and Defu Lian. Llama2Vec: Unsupervised adaptation of large language models for dense retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3490–3500, 2024.
- [31] Niklas Muennighoff. Sgpt: Gpt sentence embeddings for semantic search. *arXiv preprint arXiv:2202.08904*, 2022.
- [32] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 11897–11916, 2024.
- [33] Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. Scaling sentence embeddings with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3182–3196, 2024.
- [34] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in neural information processing systems*, volume 35, pages 27730–27744, 2022.
- [35] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in neural information processing systems*, volume 35, pages 24824–24837, 2022.
- [36] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in neural information processing systems*, volume 33, pages 1877–1901, 2020.
- [37] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.
- [38] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.
- [39] Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *The Twelfth International Conference on Learning Representations*, 2024.

- [40] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [41] 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. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.
- [42] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.
- [43] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, December 2022.
- [44] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, July 2023.
- [45] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649, 2024.
- [46] Xianming Li and Jing Li. AoE: Angle-optimized embeddings for semantic textual similarity. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1825–1839, August 2024.
- [47] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.
- [48] Zheng Liu, Chaofan Li, Shitao Xiao, Chaozhuo Li, Defu Lian, and Yingxia Shao. Matryoshka re-ranker: A flexible re-ranking architecture with configurable depth and width. *arXiv preprint arXiv:2501.16302*, 2025.
- [49] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023.
- [50] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pages 19730–19742. PMLR, 2023.
- [51] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229, 2020.
- [52] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pages 19730–19742. PMLR, 2023.
- [53] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations*, 2024.
- [54] Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, 2019.

- [55] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, 2018.
- [56] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, 2018.
- [57] Xinyu Zhang, Nandan Thakur, Odunayo Ogundepo, Ehsan Kamalloo, David Alfonso-Hermelo, Xiaoguang Li, Qun Liu, Mehdi Rezagholizadeh, and Jimmy Lin. MIRACL: A multilingual retrieval dataset covering 18 diverse languages. *Transactions of the Association for Computational Linguistics*, 11:1114–1131, 2023.
- [58] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human-generated MACHine reading COMprehension dataset, 2017.
- [59] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics, 2016.
- [60] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, 2017.
- [61] hilfialkaff DataCanary, Jiang Lili, Risdal Meg, Dandekar Nikhil, and tomtung. Quora question pairs. 2017.
- [62] Xinyu Zhang, Xueguang Ma, Peng Shi, and Jimmy Lin. Mr. TyDi: A multi-lingual benchmark for dense retrieval. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 127–137, 2021.
- [63] Wei He, Kai Liu, Jing Liu, Yajuan Lyu, Shiqi Zhao, Xinyan Xiao, Yuan Liu, Yizhong Wang, Hua Wu, Qiaoqiao She, Xuan Liu, Tian Wu, and Haifeng Wang. DuReader: a Chinese machine reading comprehension dataset from real-world applications. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 37–46. Association for Computational Linguistics, 2018.
- [64] Xiaohui Xie, Qian Dong, Bingning Wang, Feiyang Lv, Ting Yao, Weinan Gan, Zhijing Wu, Xiangsheng Li, Haitao Li, Yiqun Liu, et al. T2ranking: A large-scale chinese benchmark for passage ranking. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2681–2690, 2023.

## A Experimental Details for Training

### A.1 Hyperparameters

In this section, we provide additional training details based on Section 4.3. Specifically, we use the AdamW optimizer with an initial learning rate of  $1e-4$ , and a warm-up strategy for the first 300 steps followed by linear decay over the remaining steps. To reduce GPU memory usage, all models are trained with bfloat16 quantization, gradient checkpointing, and FlashAttention-2 [53], following [11]. We also apply gradient accumulation to process a large batch size of 512 while sampling the same dataset within each batch. The maximum sequence length is set to the standard 512 tokens by default.

Table 7 presents the hyperparameters that vary across different base models. We set the LoRA rank to 64 and LoRA alpha to 32 for 7B models following [17], while using a large LoRA alpha of 128 for the 1.3B model. For the linear scheduler, the maximum steps are set to twice the training steps. Moreover, since we adopt the instruction-tuned versions of LLaMA-2-7B and Mistral-7B, the special [INST] and [/INST] tokens are appended to the input sequence following the official instruction prompt template.

### A.2 Base Models

All base models and bidirectional encoders employed in this work are obtained from the Hugging Face platform, including: princeton-nlp/Sheared-LLaMA-1.3B, meta-llama/Llama-2-7b-chat-hf, mistralai/Mistral-7B-Instruct-v0.2, google-bert/bert-base-multilingual-cased, intfloat/e5-small-v2, and intfloat/e5-base-v2.

### A.3 Public Retrieval Datasets and Instructions

The collection of publicly available retrieval datasets used for training is curated by [16] and includes the following datasets: ELI5 (sample ratio 0.1) [54], HotpotQA [55], FEVER [56], MIRACL [57], MS-MARCO passage ranking (sample ratio 0.5) and document ranking (sample ratio 0.2) [58], NQ [3], NLI [21], SQuAD [59], TriviaQA [60], Quora Duplicate Questions (sample ratio 0.1) [61], Mr. TyDi [62], DuReader [63], and T2Ranking (sample ratio 0.5) [64].

Table 8 lists the instructions used for each dataset, which are manually written by [32]. Notably, in the query-passage pairs of retrieval datasets, task-specific instructions are appended only to the queries, without modifying the passages.

Table 7: Hyperparameters used in the experiments.

Hyperparameter	S-LLaMA-1.3B	Mistral-7B & LLaMA-2-7B
Batch Size	128	64
Gradient Accumulation Steps	4	8
Training Steps	2000	1000
Maximum Steps	4000	2000
LoRA Rank	64	64
LoRA Alpha	128	32
LoRA for Bidirectional Encoder	✓	✗

## B Experimental Details for Evaluation

### B.1 MTEB-MINI Details

Considering the substantial computational resources required for full evaluation on MTEB, we follow [11, 16] and select a subset of the MTEB for ablation and analysis. While prior studies utilize only a few datasets, our preliminary experiments suggest that evaluation on a limited subset may introduce significant bias and fail to effectively reflect the overall trends of the full MTEB. We empirically argue that a representative subset should cover as many MTEB datasets as possible to

Table 8: Instructions used for public retrieval datasets.

Dataset	Instruction (s)
ELI5	Provided a user question, retrieve the highest voted answers on Reddit ELI5 forum
HotpotQA	Given a multi-hop question, retrieve documents that can help answer the question
FEVER	Given a claim, retrieve documents that support or refute the claim
MIRACL	Given a question, retrieve Wikipedia passages that answer the question
MSMARCO Passage	Given a web search query, retrieve relevant passages that answer the query
MSMARCO Document	Given a web search query, retrieve relevant documents that answer the query
NQ	Given a question, retrieve Wikipedia passages that answer the question
NLI	Given a premise, retrieve a hypothesis that is entailed by the premise
	Retrieve semantically similar text
SQuAD	Retrieve Wikipedia passages that answer the question
TriviaQA	Retrieve Wikipedia passages that answer the question
QuoraDuplicates	Given a question, retrieve questions that are semantically equivalent to the given question
	Find questions that have the same meaning as the input question
Mr. TyDi	Given a question, retrieve Wikipedia passages that answer the question
DuReader	Given a Chinese search query, retrieve web passages that answer the question
T2Ranking	Given a Chinese search query, retrieve web passages that answer the question

Table 9: Composition of the MTEB-MINI benchmark.

Category	Dataset
Retrieval (7)	ArguAna, SciFact, NFCorpus, FiQA2018, SCIDOCS, TRECCOVID, Touche2020
Reranking (3)	AskUbuntuDupQuestions, SciDocsRR, StackOverflowDupQuestions
Clustering (6)	BiorxivClusteringS2S, BiorxivClusteringP2P, MedrxivClusteringS2S, MedrxivClusteringP2P, MedrxivClusteringP2P, TwentyNewsgroupsClustering, StackExchangeClusteringP2P
Pair Classification (3)	SprintDuplicateQuestions, TwitterSemEval2015, TwitterURLCorpus
Classification (11)	AmazonCounterfactualClassification, AmazonReviewsClassification, Banking77Classification, EmotionClassification, MassiveIntentClassification, MassiveScenarioClassification, MTOPODomainClassification, MTOPIIntentClassification, ToxicConversationsClassification, TweetSentimentExtractionClassification, ImdbClassification
STS (10)	BIOSES, SICK-R, STS12, STS13, STS14, STS15, STS16, STS17, STS22, STSBenchmark
SummEval (1)	SummEval
Overall	41 datasets

ensure consistency with evaluation results of the complete MTEB. To this end, as shown in Table 9, we introduce the MTEB-MINI by selecting 41 datasets spanning all task categories in MTEB.

## B.2 Instructions for MTEB Evaluation

To enable a fair comparison with prior leading embedding methods [32, 16, 11, 13, 17], we use the same instruction prompts for evaluation on both MTEB and MTEB-MINI. The instructions applied to each dataset are listed in Table 13.

## C Additional Results

### C.1 The L2 Norms of Contextual and EOS Tokens

To examine the respective contributions of Contextual and EOS tokens to the final text embedding, we compare their L2 norms on selected MTEB datasets. As depicted in Figure 3, we observe that the EOS token consistently shows higher L2 norms across various task categories, indicating its greater influence on the concatenated representation.



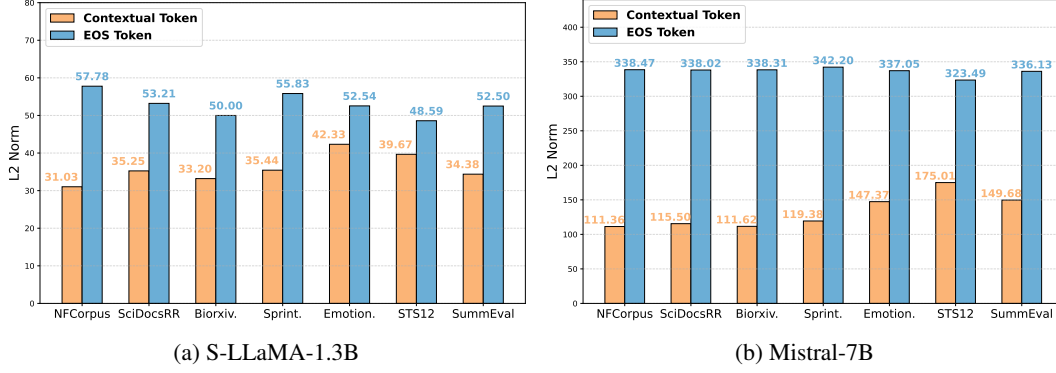


Figure 3: L2 norms of Contextual and EOS tokens on selected MTEB subsets for two base models: S-LLaMA-1.3B and Mistral-7B. The evaluated datasets span seven tasks, including NFCorpus, SciDocsRR, BiorxivClusteringP2P (Biorxiv.), SprintDuplicateQuestions (Sprint.), EmotionClassification (Emotion.), STS12, and SummEval.

Table 10: Performance comparison on MTEB-MINI (41 datasets) using different representation method with two base models: S-LLaMA-1.3B and Mistral-7B. By default, Causal2Vec generates text embedding by concatenating the output hidden states of LLM’s Contextual and EOS tokens, while Bi-EOS denotes concatenating LLM’s output EOS token with the bidirectional encoder’s output that has been processed by an MLP layer. Note: all experiments incorporate the Contextual token into LLM’s input sequence.

Task (# of datasets)	Retr. (7)	Rerank. (3)	Clust. (6)	PairClass. (3)	Class. (11)	STS (10)	Summ. (1)	Avg (41)
Metric	nDCG@10	MAP	V-Meas.	AP	Acc.	Spear.	Spear.	
S-LLaMA-1.3B								
Causal2Vec	<b>47.18</b>	<b>64.48</b>	36.61	<b>86.18</b>	<b>71.14</b>	83.76	<b>31.45</b>	<b>64.72</b>
w/ Bi-EOS	46.57	62.92	<b>36.70</b>	85.19	70.76	84.05	30.44	64.39
w/ last-token	46.46	62.93	36.32	85.48	70.32	<b>84.13</b>	30.83	64.24
Mistral-7B								
Causal2Vec	<b>51.77</b>	<b>68.51</b>	39.99	<b>88.43</b>	74.72	<b>85.38</b>	<b>30.57</b>	<b>67.79</b>
w/ Bi-EOS	51.50	68.38	39.73	88.37	74.90	85.17	30.03	67.68
w/ last-token	51.15	68.32	<b>40.04</b>	88.14	<b>75.00</b>	84.92	29.82	67.60

## C.2 Different Representation Methods

We also explore different representation methods tailored to our embedding framework. As shown in Table 10, both concatenation strategies consistently outperform last-token pooling for S-LLaMA-1.3B and Mistral-7B. This suggests that incorporating an additional context-aware token with the EOS token leads to richer semantic information while reducing the model’s reliance on the single EOS token alone. Additionally, we observe further performance improvements when the concatenated Contextual token is derived from the LLM, rather than from the bidirectional encoder followed by an MLP layer. We speculate that this helps the LLM better capture the semantic content encoded in the Contextual token.

## C.3 Impact of Freezing the Bidirectional Encoder

To investigate the impact of freezing the bidirectional encoder during Causal2Vec finetuning, we compare the performance of applying "w/ Bi-LoRA" and "w/o Bi-LoRA" with two base models: S-LLaMA-1.3B, and Mistral-7B. Table 11 presents the detailed results across all seven tasks on MTEB-MINI.

Table 11: Performance comparison on MTEB-MINI (41 datasets) with and without applying LoRA to the bidirectional encoder (Bi-LoRA), using two base models: S-LLaMA-1.3B and Mistral-7B.

Task (# of datasets) Metric	Retr. (7) nDCG@10	Rerank. (3) MAP	Clust. (6) V-Meas.	PairClass. (3) AP	Class. (11) Acc.	STS (10) Spear.	Summ. (1) Spear.	Avg (41)
S-LLaMA-1.3B								
w/ Bi-LoRA	<b>47.18</b>	<b>64.48</b>	<b>36.61</b>	<b>86.18</b>	<b>71.14</b>	83.76	<b>31.45</b>	<b>64.72</b>
w/o Bi-LoRA	46.80	63.97	36.26	86.15	70.57	<b>83.92</b>	30.76	64.43
Mistral-7B								
w/ Bi-LoRA	51.50	68.23	39.64	88.43	<b>74.86</b>	85.20	30.41	67.66
w/o Bi-LoRA	<b>51.77</b>	<b>68.51</b>	<b>39.99</b>	<b>88.43</b>	74.72	<b>85.38</b>	<b>30.57</b>	<b>67.79</b>

Table 12: Performance comparison between placing the Contextual token before and after instruction on MTEB-MINI (41 datasets) using two base models: S-LLaMA-1.3B and Mistral-7B.

Task (# of datasets) Metric	Retr. (7) nDCG@10	Rerank. (3) MAP	Clust. (6) V-Meas.	PairClass. (3) AP	Class. (11) Acc.	STS (10) Spear.	Summ. (1) Spear.	Avg (41)
S-LLaMA-1.3B								
before instruction	47.17	64.18	<b>36.78</b>	86.10	70.89	83.76	30.70	64.63
after instruction	<b>47.18</b>	<b>64.48</b>	36.61	<b>86.18</b>	<b>71.14</b>	<b>83.76</b>	<b>31.45</b>	<b>64.72</b>
Mistral-7B								
before instruction	51.65	68.16	39.26	88.21	<b>74.96</b>	85.23	29.55	67.63
after instruction	<b>51.77</b>	<b>68.51</b>	<b>39.99</b>	<b>88.43</b>	74.72	<b>85.38</b>	<b>30.57</b>	<b>67.79</b>

#### C.4 The Position of Contextual Token

In Table 12, we report the detailed MTEB-MINI results for different positions of the Contextual token, including before and after the instruction tokens.

#### C.5 Full MTEB Results

We present detailed results on all 56 MTEB datasets for the proposed Causal2Vec in Table 14, including three base models: S-LLaMA-1.3B, LLaMA-2-7B, and Mistral-7B.

## D Limitations

Despite the effectiveness of Causal2Vec, several limitations should be acknowledged: (1) Our findings suggest that a single Contextual token is sufficient to provide the missing contextual information for decoder-only LLMs. Future work could explore generating additional Contextual tokens using different bidirectional encoders or utilizing multiple task-related examples. (2) Our experiments are limited to three popular LLMs with fewer than 7B parameters, while further validation on more diverse and larger-scale LLMs could better demonstrate the scalability and robustness of our proposed mechanism.

## E Broader Impacts

Our proposed Causal2Vec can be applied to a wide range of real-world applications, including information retrieval and LLM-based retrieval-augmented generation systems. However, LLMs suffer from biases and hallucinations, which could potentially lead to negative societal impacts.

Table 13: Instructions used for evaluation on the MTEB. “STS\*” denotes that the corresponding instruction is applied to all STS datasets.

Dataset	Instruction Template
AmazonCounterfactualClassification	Classify a given Amazon customer review text as either counterfactual or not-counterfactual.
AmazonPolarityClassification	Classify Amazon reviews into positive or negative sentiment
AmazonReviewsClassification	Classify the given Amazon review into its appropriate rating category
Banking77Classification	Given a online banking query, find the corresponding intents
EmotionClassification	Classify the emotion expressed in the given Twitter message into one of the six emotions: anger, fear, joy, love, sadness, and surprise.
ImdbClassification	Classify the sentiment expressed in the given movie review text from the IMDB dataset.
MassiveIntentClassification	Given a user utterance as query, find the user intents
MassiveScenarioClassification	Given a user utterance as query, find the user scenarios
MTOPDomainClassification	Classify the intent domain of the given utterance in task-oriented conversation
MTOPIntentClassification	Classify the intent of the given utterance in task-oriented conversation
ToxicConversationsClassif.	Classify the given comments as either toxic or not toxic
TweetSentimentClassification	Classify the sentiment of a given tweet as either positive, negative, or neutral
ArxivClusteringP2P	Identify the main and secondary category of Arxiv papers based on the titles and abstracts.
ArxivClusteringS2S	Identify the main and secondary category of Arxiv papers based on the titles
BiorxivClusteringP2P	Identify the main category of Biorxiv papers based on the titles and abstracts
BiorxivClusteringS2S	Identify the main category of Biorxiv papers based on the titles
MedrxivClusteringP2P	Identify the main category of Medrxiv papers based on the titles and abstracts
MedrxivClusteringS2S	Identify the main category of Medrxiv papers based on the titles
RedditClustering	Identify the topic or theme of Reddit posts based on the titles
RedditClusteringP2P	Identify the topic or theme of Reddit posts based on the titles and posts
StackExchangeClustering	Identify the topic or theme of StackExchange posts based on the titles
StackExchangeClusteringP2P	Identify the topic or theme of StackExchange posts based on the given paragraphs
TwentyNewsgroupsClustering	Identify the topic or theme of the given news articles
SprintDuplicateQuestions	Retrieve duplicate questions from Sprint forum
TwitterSemEval2015	Retrieve tweets that are semantically similar to the given tweet
TwitterURLCorpus	Retrieve tweets that are semantically similar to the given tweet
AskUbuntuDupQuestions	Retrieve duplicate questions from AskUbuntu forum
MindSmallReranking	Retrieve relevant news articles based on user browsing history
SciDocsRR	Given a title of a scientific paper, retrieve the titles of other relevant papers
StackOverflowDupQuestions	Retrieve duplicate questions from StackOverflow forum
ArguAna	Given a claim, find documents that refute the claim
ClimateFEVER	Given a claim about climate change, retrieve documents that support or refute the claim.
CQADupstackRetrieval	Given a question, retrieve detailed question descriptions from Stackexchange that are duplicates to the given question.
DBPedia	Given a query, retrieve relevant entity descriptions from DBPedia
FEVER	Given a claim, retrieve documents that support or refute the claim
FiQA2018	Given a financial question, retrieve user replies that best answer the question
HotpotQA	Given a multi-hop question, retrieve documents that can help answer the question
MSMARCO	Given a web search query, retrieve relevant passages that answer the query
NFCorpus	Given a question, retrieve relevant documents that best answer the question
NQ	Given a question, retrieve Wikipedia passages that answer the question
QuoraRetrieval	Given a question, retrieve questions that are semantically equivalent to the given question.
SCIDOCs	Given a scientific paper title, retrieve paper abstracts that are cited by the given paper
SciFact	Given a scientific claim, retrieve documents that support or refute the claim
Touche2020	Given a question, retrieve detailed and persuasive arguments that answer the question
TRECCOVID	Given a query on COVID-19, retrieve documents that answer the query
STS*	Retrieve semantically similar text.
BUCC/Tatoeba	Retrieve parallel sentences.
SummEval	Given a news summary, retrieve other semantically similar summaries

Table 14: Results of Causal2Vec on all 56 MTEB datasets across three base models: S-LLaMA-1.3B, LLaMA-2-7B, and Mistral-7B.

Dataset	S-LLaMA-1.3B	LLaMA-2-7B	Mistral-7B	Mistral-7B (w/ ICL)
AmazonCounterfactualClassification	74.49	76.79	76.22	75.99
AmazonPolarityClassification	92.75	94.80	95.02	95.80
AmazonReviewsClassification	46.48	51.75	51.40	53.78
ArguAna	54.73	57.35	57.55	59.11
ArxivClusteringP2P	46.25	48.37	48.99	50.64
ArxivClusteringS2S	39.52	42.88	45.51	47.09
AskUbuntuDupQuestions	61.59	63.54	65.96	65.71
BIOSES	83.96	84.06	86.42	87.28
Banking77Classification	85.96	88.14	88.62	88.85
BiorxivClusteringP2P	38.13	39.05	39.24	40.82
BiorxivClusteringS2S	35.13	36.42	38.32	39.09
CQADupstackRetrieval	39.53	43.42	45.59	46.82
ClimateFEVER	32.62	32.46	35.55	34.12
DBPedia	44.85	49.85	51.65	52.09
EmotionClassification	46.82	49.74	50.56	51.40
FEVER	88.11	90.53	91.53	91.68
FiQA2018	44.52	51.29	54.96	56.03
HotpotQA	67.13	71.45	74.25	73.11
ImdbClassification	83.76	88.33	91.24	92.45
MSMARCO	40.88	41.22	42.22	42.83
MTOPDomainClassification	94.05	95.53	95.79	96.36
MTOPIntentClassification	73.45	82.38	83.12	86.23
MassiveIntentClassification	73.36	77.63	78.14	79.53
MassiveScenarioClassification	77.58	79.88	81.27	82.40
MedrxivClusteringP2P	33.38	33.13	34.33	36.32
MedrxivClusteringS2S	31.58	32.14	34.28	34.73
MindSmallReranking	32.71	32.46	32.32	32.31
NFCorpus	37.43	40.21	41.63	41.41
NQ	58.02	64.10	66.65	66.50
QuoraRetrieval	88.91	88.80	89.35	89.24
RedditClustering	56.91	63.07	64.73	64.99
RedditClusteringP2P	61.26	64.31	66.43	68.21
SCIDOCS	19.56	21.28	22.40	22.76
SICK-R	81.99	82.78	83.49	83.33
STS12	77.04	78.77	79.37	79.71
STS13	87.47	88.89	88.69	89.75
STS14	83.21	85.29	85.43	85.80
STS15	88.93	89.86	90.76	90.92
STS16	86.83	87.72	88.26	88.38
STS17	91.13	92.19	92.47	92.31
STS22	69.21	70.67	69.44	69.20
STSBenchmark	87.85	88.77	89.47	89.96
SciDocsRR	81.68	84.11	84.40	84.61
SciFact	73.04	75.77	77.52	77.92
SprintDuplicateQuestions	96.26	97.00	96.70	97.16
StackExchangeClustering	64.27	69.14	72.23	76.40
StackExchangeClusteringP2P	32.41	36.74	37.73	40.63
StackOverflowDupQuestions	50.16	52.61	55.16	54.81
SummEval	31.45	31.09	30.57	30.82
TRECCOVID	76.24	78.65	83.48	83.09
Touche2020	24.74	22.83	24.86	25.51
ToxicConversationsClassification	65.03	65.02	63.05	65.14
TweetSentimentExtractionClassification	61.53	61.44	62.52	62.46
TwentyNewsgroupsClustering	49.01	54.33	56.01	59.62
TwitterSemEval2015	75.24	79.78	81.35	82.91
TwitterURLCorpus	87.03	86.77	87.23	87.50
<b>MTEB Average (56)</b>	62.63	64.94	66.10	66.85