A Gating Layer-Based Restorable Embedding Framework for Efficient Knowledge Representations

Anonymous ACL submission

Abstract

Large language models have achieved linguistic fluency and exhibited remarkable performances in various natural language tasks without gradient updates because more number of model parameters could retain more knowledge. However, large language models are not applicable to the domain-specific tasks requiring knowledge not included in the training corpus, due to the fact that knowledge in the model parameters is not controllable during generation and updating the model parameters is costly. This research introduces efficient embedding mechanisms to separate knowledge from language models. The method divides the previous end-to-end construction of the language models into three sub-parts: sentence-level knowledge encoding, sentence-embedding-based task processing, and restoring the processed knowledge embedding to token-level embedding. The experimental results verify that most knowledge consisting of 1 or 2 sentences can be restored and the performance in the passage retrieval task is significantly improved.

1 Introduction

Recently decoder (Radford et al., 2019; Wang and Komatsuzaki, 2021) and encoder-based language models (Raffel et al., 2020; Zhang et al., 2020; Lewis et al., 2020) have improved linguistic fluency by implicitly storing and using knowledge during language understanding and generation process. Moreover, large language models (LLMs) have achieved high performance in zero-shot and few-shot settings. However, the LLM-based approaches face several problems from the point of view of usability.

LLMs are too expensive to be updated because the number of the model parameters has reached 175B (Brown et al., 2020) and 530B (Narayanan et al., 2021). To attain the contextualized representation without updating the gradient or head layers, prompt inputs are given to LLMs. When domain-specific knowledge is needed, the prompts must include adequate domain knowledge because the portion of the specific domain knowledge in the LLM parameters is likely to be small. As more domain-specific knowledge is needed, the longer prompt sharply increases the computation cost due to the quadratic memory complexity according to the input sequence length in transformer (Vaswani et al., 2017). To mitigate the computational unfeasibility, research in the field of sparse attention (Beltagy et al., 2020; Zaheer et al., 2020; Roy et al., 2021) has been conducted. Although the input sequence length capacity in the transformer has increased about 8 to 10 times, it is still a serious limitation in knowledge processing on LLMs.

In addition, LLMs sometimes produce a contradiction or a plausible untruth, so-called hallucination (Maynez et al., 2020; Shuster et al., 2021; Roller et al., 2020). Since knowledge fragments are mixed and stored in the internal LLM parameters, it is unclear which knowledge fragments are chosen dynamically in the process of inferences. The hallucination is a critical issue for commercializing language processing technologies, such as ethics or persona representation in dialogue tasks, and logic consistency in reasoning tasks.

To resolve those limitations, this paper introduces a restorable embedding framework that isolates knowledge into the external memory from the internal LLM parameters. Separating knowledge into the external memory makes the knowledge input length irrelevant to the computation cost of LLMs, and allows the detection of which knowledge is utilized so that the hallucination can be avoided. This paper also suggests the mechanisms referring to the separated knowledge.

The key contributions of this paper are:

- This paper proposes a novel deep-layered neural model framework to restore the embedding vector to the original text sequence.

- This paper proves that the proposed mechanisms maintain the performances in various downstream tasks. In the passage retrieval task in which minimizing the loss rate of information is critical, the performance is considerably improved.

- This paper analyzes the optimal original conditional context length at which the hallucination occurrences are minimized.

2 Related Works

Research on constructing fine sentence and passage embeddings has been studied in various fields such as sentence embedding and passage retrieval. Since BERT (Devlin et al., 2019) was introduced, significant research effort has been spent on lowering the computational complexity in the process of scoring or classifying sentences. Sentence embedding studies have also been conducted in long document summarization and classification tasks, as a way to alleviate large memory consumption in long document processing.

2.1 Sentence-Level Embeddings

Various sentence embedding techniques such as Skip-thought (Kiros et al., 2015), InferSent (Conneau et al., 2017), and Universal Sentence Encoder (Cer et al., 2018) have been studied. Especially to alleviate the need to compute all combinations of sentence pairs, sentence-BERT (Reimers and Gurevych, 2019) utilizes sentence embeddings in the classification and similarity scoring tasks. Sentence-BERT was trained with the semantic textual similarity (STS) dataset (Jiang et al., 2020) for semantic embeddings and shows high performances and computational efficiencies in various sentence classification and regression tasks.

2.2 Embeddings in Natural Language Tasks

Passage retrieval aims to retrieve passages related to a query from a huge corpus. In the case of the open-domain question answering (QA) datasets such as Natural Question(Kwiatkowski et al., 2019) and TriviaQA(Joshi et al., 2017) and the document augmented conversation datasets such as WizInt(Komeili et al., 2022), the relevant passages must be found from the large-scaled texts like Wikipedia[1] and Common Crawl(Carlini et al., 2021). Because the number of passages is in millions, measuring the correlation with all documents for each query causes tremendous computation requirements. Therefore, recent studies represent queries and passages as embedding vectors and measure their correlations by cosine similarity or inner product between the vectors. Several methods (Karpukhin et al., 2020; Xiong et al., 2021; Zhang et al., 2021) propose to encode queries and passages with LLM encoders.

When the sequence to be summarized is lengthy in the long document summarization task, the quadratic memory complexity according to the sequence length makes the transformer intractable. To mitigate the quadratic memory complexity problem, research has been conducted on lowering memory complexity through sparse attention (Wang et al., 2020; Kitaev et al., 2020; Tay et al., 2020; Huang et al., 2021), and generating a summary with a hierarchical transformer based on embeddings (Rohde et al., 2021; Zhang et al., 2019; Liu and Lapata, 2019; Wu et al., 2021). The hierarchical transformer utilizes embedding vectors to generate a summary through an end-to-end encoder-decoder, but restoring the embeddings to lexical sentences has not been studied yet.

3 Restorable Embedding Framework

In the previous transformer structures, semantic embeddings and their corresponding lexical features are merged in the architectures. Those structures find a document involving an answer for the given tasks such as open-domain QA, and facilitate models to extract the answer from the document. Because the previous structures inevitably require large-scale modeling, our proposed restorable embedding framework aims to isolate knowledge into external memory by converting embedding vectors to their corresponding texts. With a certain range of knowledge input length, this framework successfully restores the embedding vectors to their original texts, resulting in enhancing memory and storage efficiency since mapping information of the original text and its corresponding embedding vector is not required.

The proposed framework to separate language models and knowledge is shown in Fig. 1. This framework consists of three stages: (1) creating knowledge embedding vectors for sentence-level knowledge to minimize the loss of information and to express what it stands for; (2) processing natural language tasks using the generated embeddings and knowledge embeddings stored in external memory,

[1] https://www.wikipedia.org/

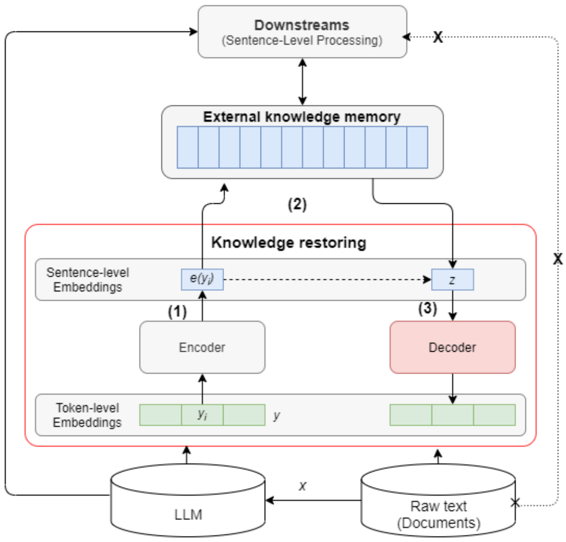

Figure 1: Conceptual diagram of the proposed restorable embedding framework.

and producing the result in the form of embedding; (3) converting the resulting embedding into natural language that humans can understand. If this framework is applied to natural language processing tasks, more contexts can be added with the same memory size. Moreover, contexts are converted into sentence-level knowledge embeddings so that looking up large contexts is avoidable.

To properly restore knowledge units in external memory, the proposed models must reconstruct their original sentences describing the corresponding knowledge semantics. Thus, this paper suggests the red box in Fig. 1, which represents our study to express the token-level embedding sequence as one embedding and to restore the expressed embedding into the original text. The proposed embedding techniques are also designed to improve the performances of various downstream tasks.

The notations in the paper are defined as follows.

- $\mathbf{x} = \{x_1, \cdots, x_T\}$: Token sequence to be expressed as embedding vector.
- $\mathbf{y} = \{y_1, \cdots, y_M\}$, $\mathbf{z} = \{z_1, \cdots, z_N\}$: Input token sequences to encoder and decoder respectively.
- d_{model}: Model dimensionality
- d_{repr}: Representation vector dimensionality
- $e(y_i)$: Embedding vector of i-th token in y
- $h(y_i)$: Contextualized embedding of y_i by encoder
- $\mathbf{e}_{repr}$: Encoded vector from encoder

3.1 Description of Conventional Embeddings

The encoder generating text embeddings utilizes the following methods: (a) employing the embedding vector whose CLS token is located at the start, and (b) exploiting the vector obtained through mean pooling. In the case of (a), the CLS token and text sequence are concatenated and then given to the encoder. The contextualized embedding value of the CLS token position is projected with a linear layer and creates an embedding vector. The $\mathbf{e}_{repr}$ of $\mathbf{x}$ is defined as Eq. 1 with the learnable projection matrix $\mathbf{W}$.

$$\begin{aligned} \mathbf{e}_{repr} &= \mathbf{W}h(y_1), \mathbf{W} \in \mathbb{R}^{d_{model} \times d_{repr}} \\ \text{where } \mathbf{y} &= \{[CLS], x_1, \cdots, x_T\} \end{aligned} \tag{1}$$

For (b), the embedding vector is achieved by projecting the vector obtained from mean pooling of all contextualized embedding values into a linear layer with the text sequence. The embedding vector $\mathbf{e}_{repr}$ of $\mathbf{x}$ is defined as Eq. 2.

$$\mathbf{e}_{repr} = \mathbf{W}\left(\sum_{i=1}^{T}(h(x_i)/\sqrt{T})\right) \tag{2}$$

For the decoding process, there are two vanilla methods to restore $\mathbf{e}_{repr}$ to the original $\mathbf{x}$ as shown in (a) and (b) of Fig. 2. (a) employs a decoder structure without cross-attention blocks like GPT. The decoder is trained to generate the original sentence with the concatenation of $\mathbf{e}_{repr}$ and the original text sequence $\mathbf{x}$. (b) utilizes $\mathbf{e}_{repr}$ as the key/value of the cross attention block in the decoder structure, concatenates the BOS token and $\mathbf{x}$ as the decoder input, and trains the model to output the original sentence. (a) is named as the input decoder whose input and target sequences are $e(\mathbf{z}) = \{\mathbf{e}_{repr}, e(x_1), \cdots, e(x_L)\}$ and $\{e(x_1), \cdots, e(x_L), e([EOS])\}$ in each. (b) is designated as the cross-attention-based decoder whose input and target sequences are $e(\mathbf{z}) = \{e([BOS]), e(x_1), \cdots, e(x_L)\}$ and $\{e(x_1), \cdots, e(x_L), e([EOS])\}$ in each. $\mathbf{e}_{repr}$ becomes the key and value in the cross-attention layer.

Cross-attention mechanisms calculate and sum semantic correlations with the key/value sequence dimension. An embedding in cross-attention illustrates multiplication for the inner product between the query vector and the scalar value of the embedding vector, and then addition to the query vector. In the embedding vector, not only the highly related elements to the current query vector but also

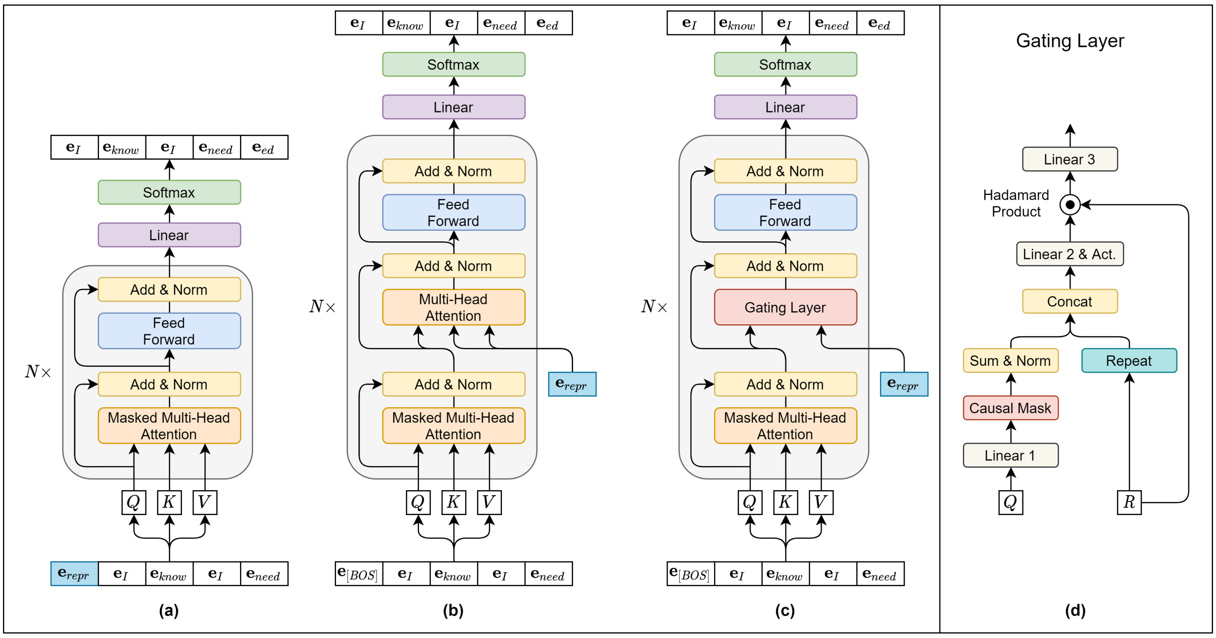

Figure 2: Decoder structures for restoring the embedding vector to the original text "I needed you.". (a) Input decoder utilizing the embedding vector as input. (b) Cross-attention-based decoder employing the embedding vector as key/value of the cross-attention layer. (c) The proposed gating layer-based decoder. (d) The proposed gating layer structure for (c).

all elements are reflected as much as the similarity between the embedding vector and the current query vector. The expected query vector $\hat{\mathbf{q}}_i$ updated by cross-attention is described in Eq. 3. In this equation, the query vector sequence input to the cross-attention layer is $\mathbf{q}_{1:N}$, and the i-th query vector is represented by $\mathbf{q}_i$. $\mathbf{e}_{repr}$ multiplied by a scalar c is added to the query vector. d_{model} and d_{repr} must be the same for the inner product between two vectors.

$$
\begin{aligned}
\hat{\mathbf{q}}_i =& \mathbf{q}_i + c \cdot \mathbf{e}_{repr}, \\
& \text{where} \quad c = \mathbf{q}_i \cdot \mathbf{e}_{repr} \\
& \text{s.t.} \quad d_{model} = d_{repr} \\
& \mathbf{q}_i \in \mathbb{R}^{d_{model}}, \mathbf{e}_{repr} \in \mathbb{R}^{d_{repr}}
\end{aligned} \tag{3}
$$

This paper concentrates on the case where $d_{model} = d_{repr}$. However, the constraint may be a disadvantage in constructing embeddings with minimizing the loss of information if increasing the size of d_{repr} to include more information in $\mathbf{e}_{repr}$ is necessary. Therefore, this paper proposes the addition of a gating layer that enables decoding even if d_{repr} and d_{repr} are different. The gating successfully extracts the semantically related elements to the current query vector from the embedding vector.

3.2 Gating Layer for Restorable Embeddings

The proposed mechanisms are described in (c) and (d) of Fig. 2. (c) shows the gating layer-based decoder instead of the cross-attention layer in (b), and (d) shows the proposed gating layer structure. The input of the gating layer is a query and $\mathbf{e}_{repr}$. When $\mathbf{q}_i$ inputs to the gating layer, $\mathbf{q}_i$ is projected to d_{repr} through the projection matrix $\mathbf{W}_1$, resulting in $\tilde{\mathbf{q}}_i$. $\bar{\mathbf{q}}_i$ is a normalized vector through causal maskings and add operations. As depicted in Eq. 4, $\tilde{\mathbf{q}}_i$ is added to the j-th vectors smaller than i and divided by i.

$$
\bar{\mathbf{q}}_i = \sum_{j=1}^{i} \tilde{\mathbf{q}}_j / \sqrt{i} \tag{4}
$$

In Eq. 5, each $\bar{\mathbf{q}}_i$ vector with $\mathbb{R}^{2d_{repr}}$ dimension is projected to d_{repr} through $\mathbf{W}_2 \in \mathbb{R}^{2d_{repr} \times d_{repr}}$, and then activation function is applied. The activated $\dot{\mathbf{q}}_i$ is gated through the hadamard product with $\mathbf{e}_{repr}$, and finally projected to d_{model} through $\mathbf{W}_3 \in \mathbb{R}^{d_{repr} \times d_{model}}$.

$$
\begin{aligned}
\ddot{\mathbf{q}}_i =& (\text{Act}(\dot{\mathbf{q}}_i \mathbf{W}_2) \odot \mathbf{e}_{repr}) \mathbf{W}_3 \\
& \text{where} \quad \dot{\mathbf{q}}_i = \text{Concat}(\bar{\mathbf{q}}_i ; \mathbf{e}_{repr})
\end{aligned} \tag{5}
$$

As shown in (c) of Fig. 2, $\ddot{\mathbf{q}}_i$ is added to $\mathbf{q}_i$ and then normalized by layer normalization. Therefore, $\mathbf{e}_{repr}$ gated by the hadamard product is added to

$\mathbf{q}_i$. (c) is called the gating decoder composed of the gating layer in the decoder, and the dimension and semantics of the input and target sequence of the gating decoder are the same as those of the cross-attention-based decoder.

The proposed learning objective follows the auto-regressive object function, as explained in Eq. 6.

$$\max_{\theta} \log p_{\theta}(\mathbf{x}) = \sum_{t=1}^{T} \log p_{\theta}(x_t | \mathbf{x}_{<t}, enc_{\hat{\theta}}(\mathbf{x})), \hat{\theta} \subset \theta \tag{6}$$

$enc_{\hat{\theta}}$ denotes an encoder function parameterized by $\hat{\theta}$, and p_{θ} denotes the entire encoder-decoder function parameterized by θ.

The gating layer described in (d) of Fig. 2 proposes a new structure containing causal making instead of the redundant multi-head attention shown in (b). The proposed gating layer excludes the duplicated computation of multi-head attention and includes a causal mask which is autoregressive training. The structure employs the advantages of multi-head attention and causal mask techniques. The multi-head attention analyzes the relevance in various perspectives, regardless of sequential and positional context. On the other hand, the causal mask successfully analyzes the correlations. Additionally, the gating layer attains higher computational efficiency by eliminating the repeated multi-head attention structure in (b).

4 Experiments

If the proposed embeddings successfully restore the semantics, the performances of the relevant downstream tasks should be improved with the embeddings. For experimental evaluation, this paper applied the proposed methods to the text restoration and passage retrieval tasks with Natural question (Kwiatkowski et al., 2019) datasets. Perplexity(Sennrich, 2012), ROUGE (Recall-Oriented Understudy for Gisting Evaluation)(Lin and Hovy, 2003; Lin and Och, 2004) scores are measured for the experiments.

4.1 Experiments for Text Restoration Task

C4 RealNewsLike (Raffel et al., 2020) was utilized as a raw corpus for the text restoration task and pre-processed in the same way CommonCrawl(Carlini et al., 2021) was pre-processed in FakeNews (Zellers et al., 2019), such as bad word and deduplication filtering. The pre-processed

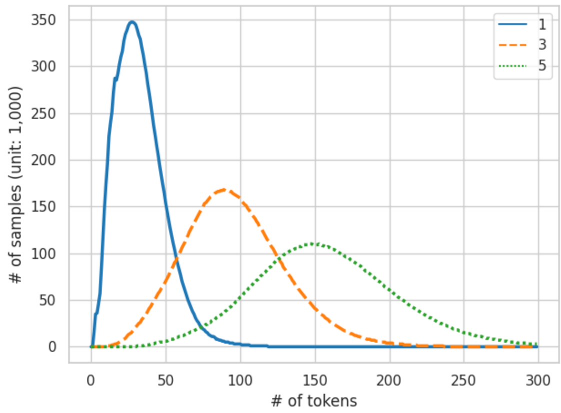

Figure 3: Token length distributions of 1, 3, and 5 sentences in C4 RealNewsLike for the text restoration task.

dataset consists of 13 million and 13,863 samples for training and validation respectively.

To examine the performances in downstream tasks according to the text sequence length, the dataset was divided at sentence level using the sentence tokenizer in NLTK (Bird and Loper, 2004). Figure 3 shows the token length distributions according to the number of sentences in C4 RealNewsLike. The average token length according to the number of sentences is about 33, 96, and 156 for 1, 3, and 5 sentences respectively.

4.1.1 Experiemtal Settings

The training was conducted for 1 epoch after initializing with the pre-trained weights of a small configuration of T5 (Raffel et al., 2020). For examining the performance difference in text restoration, both freezing and updating the weights transferred from T5 were evaluated. In the freezing layers, since only the last projection matrix of the encoder is learnable as a variable to make a restorable embedding, the text restoration with three additional transformer layers was considered and the parameters were randomly initialized. As shown in Table 1, the different configurations from (a) to (d) were evaluated with the randomly initialized parameters for each encoder and decoder variant.

Adam optimizer and linear learning rate scheduling were employed, and d_{model} and d_{repr} were set to 512 in all experiments. Gated ReLU (Dauphin

Configuration	(a)	(b)	(c)	(d)
Freezing the pre-trained weights	N	Y	N	Y
Number of additional layers	0	0	3	3

Table 1: Experimental configurations for the text restoration task.

Decoder	With CLS token				With mean pooling			
	PPL	R-1	R-2	R-L	PPL	R-1	R-2	R-L
(a) 6 layers from pre-trained model + no additional layers								
Without KE	6.178	9.87	0.79	8.09	1.16	93.37	82.93	89.72
Cross-attention KE	6.10	7.09	0.19	6.24	1.10	95.14	87.80	92.76
Gating layer KE (Ours)	6.04	11.21	0.55	8.21	**1.04**	**97.76**	**94.63**	**96.94**
(b) 6 layers from pre-trained model (Freeze) + no additional layers								
Without KE	1.79	13.33	0.75	9.53	2.24	65.99	34.45	50.96
Cross-attention KE	6.22	12.29	0.78	9.30	2.04	67.97	37.85	54.00
Gating layer KE (Ours)	6.16	11.13	0.29	8.47	**1.93**	**70.54**	**40.83**	**56.81**
(c) 6 layers from pre-trained model + 3 additional layers (Random initialization)								
Without KE	6.18	**13.32**	**0.75**	**9.53**	1.15	92.63	83.34	89.63
Cross-attention KE	6.10	9.95	0.21	8.31	1.12	94.13	86.26	91.62
Gating layer KE (Ours)	**6.04**	10.81	0.56	8.07	**1.03**	**98.32**	**96.30**	**97.91**
(d) 6 layers from pre-trained model (Freeze) + 3 additional layers (Random initialization)								
Without KE	6.30	**11.86**	**0.77**	**8.84**	1.34	84.77	69.68	81.12
Cross-attention KE	6.22	11.21	0.55	8.21	1.29	87.18	73.07	83.79
Gating layer KE (Ours)	**6.16**	9.88	0.58	7.57	**1.09**	**95.95**	**91.07**	**95.04**

Table 2: Text restoration performance on a single sentence according to the experimental configurations in Table1. The proposed embeddings were utilized to construct knowledge embedding (KE) vectors and the decoder type. PPL, R-1, R-2, and R-L denote perplexity, ROUGE-1, ROUGE-2, and ROUGE-L respectively.

et al., 2017) was used for the activation function in the gating layer, and the detailed hyperparameters for the model and optimizer in the experiments are described in Table 6 in Appendix A.

4.1.2 Experimental Results

For the single-sentence restoration task, as illustrated in Table 2, the CLS token-based approach underperforms the other methods in all configurations, even with three randomly initialized layers. From the perspective that Perplexity and ROUGE scores are not correlated, the global attention mechanisms help to make effective token-level contextualized embeddings, but there seems a limit to generating appropriate sentence-level embeddings.

The mean pooling approach overperforms the CLS-based method in all configurations. Because all tokens are directly involved in generating embeddings, the loss of information is minimized, and high restoration performances are achieved. By comparing the single-sentence restoration performance according to decoders in mean pooling, all performance metrics are improved with the proposed gating layer in all experimental configurations. Therefore, we evaluate that the proposed gating layer-based knowledge embedding model guarantees high and robust restoration.

With mean pooling-based embeddings, the model without freezing the weights from the pre-trained model draws higher performances whether additional layers are added or not. Those performance differences may be due to the gap in the number of adjustable model parameters. For example, the model configuration (a) in Table 2 depicts significantly higher performance than the model configuration (b). Whereas all weights of 6 layers in (a) can be updated during the embedding process, the last projection layer can be updated in (b). The experimental results illustrate that the number of adjustable parameters is an important factor for sentence-based knowledge embedding models.

Configuration	# S	PPL	R-1	R-2	R-L
Cross-attention-based decoder					
(c) + extra layers	1	1.12	94.13	86.26	91.62
	3	1.89	63.08	29.25	46.87
	5	2.80	52.35	15.09	31.28
(d) + freeze & extra layers	1	1.29	87.18	73.07	83.79
	3	2.48	59.00	24.39	44.09
	5	3.50	51.30	14.58	31.00
Gating layer-based decoder (Ours)					
(c) + extra layers	1	1.03	**98.32**	**96.30**	**97.91**
	3	1.37	**72.11**	**50.45**	**64.16**
	5	2.08	**52.82**	**18.91**	**36.77**
(d) + freeze & extra layers	1	1.09	**95.95**	**91.07**	**95.04**
	3	1.75	**67.14**	**39.97**	**58.43**
	5	2.76	**52.38**	**17.92**	**36.83**

Table 3: Text restoration performance according to the experimental configurations in Table1 and the number of original sentences denoted as # S. Mean pooling was employed to generate embeddings. PPL, R-1, R-2, and R-L denote perplexity, ROUGE-1, ROUGE-2, and ROUGE-L in each.

Table 3 shows the text restoration performances with either the cross-attention-based or the proposed gating layer-based decoders, according to the original text length. The experimental results indicate that the recovery performance decreases as the number of sentences increases, meaning that the amount of information accommodated in a vector of a certain dimension is limited. More experimen-

	# of sentences	R@20	R@100
No additional layers			
T5-small		49.58	67.12
(a)	1	**64.33**	**78.34**
	3	63.09	78.34
	5	63.09	77.88
(b) + freeze	1	63.61	78.39
	3	62.56	77.71
	5	62.18	77.67
Additional layers			
T5-small + additional layers		55.73	72.37
(c)	1	64.07	78.05
	3	63.13	77.82
	5	63.61	78.30
(d) + freeze	1	**70.30**	**83.32**
	3	68.70	82.29
	5	68.46	82.13

Table 4: Passage retrieval performance in Natural questions with the proposed embeddings, according to experimental configurations in Table 1.

tal results on other text lengths, configurations, and decoder types can be found in Appendix B.

4.2 Experiments for Passage Retrieval Task

4.2.1 Experiment Settings

For the passage retrieval task, the performances were measured to examine the effect of the proposed embedding mechanisms in downstream tasks. Dense passage retrieval (DPR) uses a bi-encoder including two encoders - a query encoder and a passage encoder. The evaluated models were trained with in-batch training (Karpukhin et al., 2020) by utilizing the positive passages of other samples in the batch as negative samples. The detailed hyperparameters are illustrated in Table 7 in Appendix A.

The Natural Question data and Wikipedia passages employed in the DPR downstream task were utilized for our experiments. For the evaluation, the recall of whether passages containing the correct answer for each question were retrieved in the top-K passages among the 21,015,324 passages was measured.

4.2.2 Experimental Results

For no additional layers, the performances with the proposed mechanisms were much superior to the direct transfer learning with T5-small. Even when randomly initialized additional layers were added, the passage retrieval with the proposed embedding models showed higher performance than the others. The performance gaps demonstrate that the proposed model is trained to construct efficient knowledge embeddings with minimizing the loss of information for each passage.

4.3 Analysis on Experimental Results

For no additional layers, the sentence restoration with freezing parameters recorded lower performances than that without freezing. Freezing parameters with additional layers showed performance improvements compared to freezing parameters without extra layers. Whereas the performances of the text restoration task represented superior without freezing pre-trained weights, the performances of the passage retrieval task showed better with freezing them. The reason might be that some representations for passage retrieval are damaged while the unfrozen model parameters learn knowledge restoration.

The proposed gating layer-based restorable embedding framework which possesses the external knowledge memory and employs the additional knowledge embeddings demonstrates high performances under all conditions - learning with a language modeling objective and learning the restoration while maintaining the pre-learned language model weights. Especially in (d) of Table 4, the proposed restorable embeddings performed an important role in the process of learning the semantic restoration of natural language, despite updating even fewer model parameters.

For the qualitative analysis, Table 5 exemplifies the original texts and samples restored by the gating layer-based or cross-attention-based decoders. With the proposed gating-layer-based docer, in the case of single-sentence input, complete text restoration was observed, meaning that the samples were restored with almost no loss of information. For three sentences, the first sentence was absolutely restored, but the second and third sentences omitted some words or generated different words from the original text. In particular, the wrong sentence restoration tends to appear more frequently in the latter sentences than in the former sentences.

For five sentences, more latter sequences such as the fourth and fifth sentences in Table 5 tend to be generated plausibly but semantically differently because the information from the original sentences is mixed in the restored sentences. The hallucination problem appears probably due to the loss of information during sentence encoding. As a result, under the condition of the sentence vector dimension and model size used in our experiments, converting only one or two sentences into embedding looks appropriate to prevent hallucination problems and minimize the loss of information.

Gating layer-based decoder		
Original	1	Was it a surprise to you that you were given the arts and culture position?
	2	No, there is no surprise when you are a cadre of the ANC because you are deployed anywhere.
	3	You are given a five-year contract to do a portfolio and when you are finished, you wait for another one.
	4	At no stage do you have a say.
	5	What qualities do you bring to the position?
1 sentence		
Restored	1	Was it a surprise to you that you were given the arts and culture position?
3 sentences		
Restored	1	Was it a surprise to you that you were given the arts and culture position?
	2	No, there is no surprise when you are a cadre of the ANC because you are deployed overseas.
	3	You are given a five-year contract to do a portfolio and when you (are) finish, you are waiting for another.
5 sentences		
Restored	1	Was it a surprise to you that you were given the arts and culture culture?
	2	No, there is no surprise when you are a candidate of the ANC because you are deployed anywhere.
	3	You are given a four-year contract to do a portfolio and when you (are) finish(ed), you are no longer looking for one.
	4	At one stage did you have a capabilities?
	5	What does the message bring to you?
Cross-attention-based decoder		
Original	1	Two bedrooms home on a corner lot.
	2	Two car detached garage.
	3	Nice covered front porch.
	4	Seller will not complete any repairs to the subject property, either lender or buyer requested.
	5	The property is sold in AS IS condition.
5 sentences		
Restored	1	Two car garage on a corner lot.
	2	Two covered covered porch.
	3	Sony front porch.
	4	Nice covered garage will not return any repairs to the seller, either buyer or seller.
	5	The property is listed in ASOLD condition.

Table 5: Original texts and samples restored by the gating layer-based or cross-attention-based decoders, according to the input text length. Blue texts represent parts different from the original text, and red texts indicates parts omitted from the original text.

5 Conclusions

This paper introduces a gating layer-based restorable embedding framework for constructing restorable embeddings of knowledge in the natural language process and proposes the gating layer structure to improve the restoration performance with the knowledge embeddings. The extracted knowledge embedding vectors from our mechanisms make information processing in natural language processing efficient. The experiments evaluate that the proposed gating layer-based embeddings successfully perform the downstream tasks such as the text restoration and passage retrieval tasks by showing superior performance qualitatively as well as quantitatively.

This paper focuses on how to restore the sentence-level embeddings to the original texts. The effective encoder structures and the way to construct effective embeddings are not considered in this work. Therefore, further research is to improve the efficiency of semantic representations in embeddings and to extend usability in a variety of natural language processing tasks under the consideration of effective mechanisms for storing and referencing knowledge.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

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.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. 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, Copenhagen, Denmark. Association for Computational Linguistics.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 933–941. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. 2021. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1419–1436, Online. Association for Computational Linguistics.

Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Tuo Zhao. 2020. SMART: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2177–2190, Online. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. 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, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. 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, Online. Association for Computational Linguistics.

Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2022. Internet-augmented dialogue generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8560–8578, Dublin, Irelan. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. volume 7, pages 452–466, Cambridge, MA. MIT Press.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 150–157.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612, Barcelona, Spain.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081, Florence, Italy. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. 2021. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '21, New York, NY, USA. Association for Computing Machinery.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Tobias Rohde, Xiaoxia Wu, and Yinhan Liu. 2021. Hierarchical learning for generation with long source sequences. *CoRR*, abs/2104.07545.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2020. Recipes for building an open-domain chatbot. *CoRR*, abs/2004.13637.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. 2021. Efficient Content-Based Sparse Attention with Routing Transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68.

Rico Sennrich. 2012. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 539–549, Avignon, France. Association for Computational Linguistics.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation.

Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. 2020. Sparse sinkhorn attention. *CoRR*, abs/2002.11296.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *CoRR*, abs/2006.04768.

Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Hi-transformer: Hierarchical interactive transformer for efficient and effective long document modeling. 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 848–853, Online. Association for Computational Linguistics.

Wenhan Xiong, Xiang Lorraine Li, Srinivasan Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen-tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oğuz. 2021. Answering complex open-domain questions with multi-hop dense retrieval. *International Conference on Learning Representations*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5059–5069, Florence, Italy. Association for Computational Linguistics.

Yuyu Zhang, Ping Nie, Arun Ramamurthy, and Le Song. 2021. Answering any-hop open-domain questions with iterative document reranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 481–490, New York, NY, USA. Association for Computing Machinery.

A Hyperparameter Settings in Experiments

Table 6 shows the hyperparameters of the model and optimizer when learning the text restoration task.

Encoder & Decoder		Optimizer & Generation	
Name	Value	Name	Value
d_{model}	512	Algorithm	AdamW
Number of attention heads	8	Learning rate	1e-3
Number of attention layers	6	Adam epsilon	1e-8
$d_{feedforward}$	2048	Weight decay	1e-2
Drop out rate	0.1	Scheduling	Linear
Activation for feed-forward	Relu	Warm up	Y
Epsilon for layer normalization	1e-6	Warm up rate	0.1
Max positional embedding size	512	Number of beams	4
Initialize factor	1.0	Early stopping	Y
Positional embedding type	Relative bucket embeddings	Top-k	50
Positional bucket size	32	Top-p	50

Table 6: Hyperparameters for training text restoration.

Table 7 illustrates the hyperparameters when learning the passage retrieval task.

Name	Value
Batch size	128
Epochs	40
Optimizer	AdamW
Learning rate	1e-3
Adam epsilon	1e-8
Weight decay	0
Scheduling	Linear
Warm up	Y
Warm up rate	0.2
Max length for query	70
Max length for context	350
Number of positive context per sample	1
Number of negative context per sample	1

Table 7: Hyperparameters for training passage retrieval

B More Experimental Results for Text Restoration Task

Table 8 shows all restoration performances according to the experimental configuration, the method used to create the embedding vector and the decoder type.

# Sentences	Decoder	Classification token PPL	R-1	R-2	R-L	Mean pooling PPL	R-1	R-2	R-L
1	(a) 6 layers from pre-trained model + 0 additional layers								
	Input	6.178	9.87	0.79	8.09	1.16	93.37	82.93	89.72
	Cross	6.10	7.09	0.19	6.24	1.10	95.14	87.80	92.76
	Gating	6.04	11.21	0.55	8.21	1.04	97.76	94.63	96.94
	(b) 6 layers from pre-trained model (freeze) + 0 additional layers								
	Input	1.79	13.33	0.75	9.53	2.24	65.99	34.45	50.96
	Cross	6.22	12.29	0.78	9.30	2.04	67.97	37.85	54.00
	Gating	6.16	11.13	0.29	8.47	1.93	70.54	40.83	56.81
	(c) 6 layers from pre-trained model + 3 additional layers (random initialization)								
	Input	6.18	13.32	0.75	9.53	1.15	92.63	83.34	89.63
	Cross	6.10	9.95	0.21	8.31	1.12	94.13	86.26	91.62
	Gating	6.04	10.81	0.56	8.07	1.03	**98.32**	**96.30**	**97.91**
	(d) 6 layers from pre-trained model (freeze) + 3 additional layers (random initialization)								
	Input	6.30	11.86	0.77	8.84	1.34	84.77	69.68	81.12
	Cross	6.22	11.21	0.55	8.21	1.29	87.18	73.07	83.79
	Gating	6.16	9.88	0.58	7.57	1.09	**95.95**	**91.07**	**95.04**
3	(a) 6 layers from pre-trained model + 0 additional layers								
	Input	8.13	13.33	0.48	11.08	2.33	58.98	23.10	40.36
	Cross	8.04	13.14	0.26	9.55	1.83	64.86	30.42	47.79
	Gating	7.90	18.41	1.14	12.72	1.49	70.79	43.06	58.97
	(b) 6 layers from pre-trained model (freeze) + 0 additional layers								
	Input	8.33	12.70	0.07	10.45	4.88	43.60	12.08	24.60
	Cross	8.21	14.17	0.34	10.85	4.44	45.37	12.87	25.07
	Gating	8.08	14.80	0.79	10.86	4.09	47.52	13.81	25.99
	(c) 6 layers from pre-trained model + 3 additional layers (random initialization)								
	Input	8.14	14.32	0.32	11.36	2.31	54.43	21.22	39.01
	Cross	8.04	14.48	0.79	10.88	1.89	63.08	29.25	46.87
	Gating	7.91	14.67	0.42	11.10	1.37	**72.11**	**50.45**	**64.16**
	(d) 6 layers from pre-trained model (freeze) + 3 additional layers (random initialization)								
	Input	8.34	11.20	0.13	9.70	2.96	51.82	18.70	38.18
	Cross	8.22	15.07	0.23	11.76	2.48	59.00	24.39	44.09
	Gating	8.09	16.81	1.11	11.98	1.75	**67.14**	**39.97**	**58.43**
5	(a) 6 layers from pre-trained model + 0 additional layers								
	Input	8.80	11.98	0.24	10.69	3.60	49.63	13.45	28.19
	Cross	8.67	15.14	0.87	12.53	2.75	49.63	13.45	28.19
	Gating	8.53	11.19	0.21	8.85	2.25	55.36	18.54	35.98
	(b) 6 layers from pre-trained model (freeze) + 0 additional layers								
	Input	9.02	13.98	0.09	12.43	6.30	38.24	8.87	20.48
	Cross	8.87	13.26	0.21	11.46	5.80	41.25	9.63	21.00
	Gating	8.74	11.46	0.12	10.12	5.39	43.66	10.60	21.79
	(c) 6 layers from pre-trained model + 3 additional layers (random initialization)								
	Input	8.80	4.71	0.09	4.42	3.36	46.57	12.34	28.54
	Cross	8.66	16.96	0.80	12.30	2.80	52.35	15.09	31.28
	Gating	8.54	7.42	0.29	6.15	2.08	**52.82**	**18.91**	**36.77**
	(d) 6 layers from pre-trained model (freeze) + 3 additional layers (random initialization)								
	Input	9.02	8.02	0.30	7.38	4.19	45.31	11.46	27.65
	Cross	8.87	12.02	0.34	10.80	3.50	51.30	14.58	31.00
	Gating	8.75	17.16	1.25	11.79	2.76	**52.38**	**17.92**	**36.83**

Table 8: Restoration performance according to the experimental configuration, the method used to create the embedding vector, and the decoder type.

C Passage Retrieval Performance of Proposed Model

Table 9 shows the passage retrieval performance of the proposed model according to the configuration.

	# Sentences	# Additional layers	R@1	R@5	R@20	R@100
Random initialize		0	14.77	32.68	49.58	67.12
W/ freeze	1	0	21.50	44.11	63.61	78.39
W/ freeze	3	0	21.43	43.96	62.56	77.71
W/ freeze	5	0	21.18	43.61	62.18	77.67
WO/ freeze	1	0	**24.34**	**47.49**	**64.33**	**78.34**
WO/ freeze	3	0	22.29	45.05	63.09	78.34
WO/ freeze	5	0	22.18	45.08	63.09	77.88
Random initialize		3	16.88	37.90	55.73	72.37
W/ freeze	1	3	**26.92**	**52.54**	**70.30**	**83.32**
W/ freeze	3	3	24.97	50.02	68.70	82.29
W/ freeze	5	3	25.05	49.56	68.46	82.13
WO/ freeze	1	3	21.53	45.97	64.07	78.05
WO/ freeze	3	3	20.97	44.83	63.13	77.82
WO/ freeze	5	3	22.41	45.13	63.61	78.30

Table 9: Passage retrieval performance in Natural Questions according to experimental configuration and sentence length.

D Performance on Various Sentence-Level NLP Tasks

Table 10 shows the performance on the various sentence-level downstream tasks with the sentence embeddings of the proposed model.

					GLUE		
			MNLI	QNLI	WNLI	MRPC	QQP
	# Sentences	# Additional layers	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy
Random initialize		0	74.91	80.82	58.33	75.00	**88.81**
W/ freeze	1	0	75.58	81.68	52.78	74.51	88.43
W/ freeze	3	0	75.48	81.66	37.50	**77.21**	88.47
W/ freeze	5	0	**75.58**	**81.92**	55.56	74.26	88.32
WO/ freeze	1	0	72.38	80.33	56.94	71.81	88.69
WO/ freeze	3	0	72.34	80.56	**58.33**	74.26	88.69
WO/ freeze	5	0	72.41	81.28	56.94	73.04	88.50
Random initialize		0	74.93	78.53	52.78	74.26	89.89
W/ freeze	1	3	**75.74**	81.97	50.00	71.57	89.96
W/ freeze	3	3	75.73	82.27	55.56	72.79	**90.01**
W/ freeze	5	3	75.69	**82.65**	45.83	73.53	89.96
WO/ freeze	1	3	72.47	79.83	56.94	72.79	89.04
WO/ freeze	3	3	72.26	80.38	52.78	**75.25**	89.12
WO/ freeze	5	3	72.10	80.22	**56.94**	74.26	89.11

			GLUE	SSTDataset	TREC	
			SST2	SSTDataset	Coarse	Fine
	# Sentences	# Additional layers	Accuracy	Accuracy	Accuracy	Accuracy
Random initialize		0	91.28	85.42	**97.02**	**85.91**
W/ freeze	1	0	**91.74**	**86.05**	96.83	85.32
W/ freeze	3	0	91.17	85.96	96.03	85.71
W/ freeze	5	0	91.63	85.96	96.23	83.93
WO/ freeze	1	0	86.93	77.90	93.85	78.17
WO/ freeze	3	0	87.84	78.08	94.25	80.16
WO/ freeze	5	0	87.96	79.17	94.84	81.15
Random initialize		0	92.09	85.78	97.02	**92.46**
W/ freeze	1	3	92.55	85.69	96.83	89.48
W/ freeze	3	3	**92.55**	85.33	**97.22**	91.47
W/ freeze	5	3	91.97	**86.50**	96.43	91.67
WO/ freeze	1	3	87.16	76.54	92.66	83.13
WO/ freeze	3	3	88.19	77.45	94.84	84.13
WO/ freeze	5	3	88.76	78.17	94.84	84.72

Table 10: Performance on various sentence-level downstream tasks with the sentence embeddings of the proposed model.

E Restored Samples

This section shows samples restored by the models trained on sentence restoration (No cherry-picking). For five sentences, in the sentences generated by the cross-attention-based decoder, parts of the sentence such as subjects and objects are mixed. For the sentences generated by the gating layer-based decoder, almost no parts are mixed. In five sentences of Table 13, the restored texts by the cross-attention-based decoder are a jumble of information.

Original	1	Was it a surprise to you that you were given the arts and culture position?
	2	No, there is no surprise when you are a cadre of the ANC because you are deployed anywhere.
	3	You are given a five-year contract to do a portfolio and when you are finished, you wait for another one.
	4	At no stage do you have a say.
	5	What qualities do you bring to the position?
Gating layer-based decoder		
1 sentence		
Restored	1	Was it a surprise to you that you were given the arts and culture position?
3 sentences		
Restored	1	Was it a surprise to you that you were given the arts and culture position?
	2	No, there is no surprise when you are a cadre of the ANC because you are deployed overseas.
	3	You are given a five-year contract to do a portfolio and when you (are) finish, you are waiting for another.
5 sentences		
Restored	1	Was it a surprise to you that you were given the arts and culture culture?
	2	No, there is no surprise when you are a candidate of the ANC because you are deployed anywhere.
	3	You are given a four-year contract to do a portfolio and when you (are) finish(ed), you are no longer looking for one.
	4	At one stage did you have a capabilities?
	5	What does the message bring to you?
Cross-attention-based decoder		
1 sentence		
Restored	1	Was it a surprise to you that you were given the arts and culture position?
3 sentences		
Restored	1	Was it a surprise to you when you were given the arts and culture culture?
	2	No, there is no surprise that you are a part of the ANC because you are deployed there.
	3	You are paid a five-year contract when you are ready to do a portfolio and finish another, for five years.
5 sentences		
Restored	1	Was it a surprise to you that there was no talent or culture when you were awarded the ANC?
	2	No, you are a part of the arts department.
	3	You are given that you are ready to finish a five-year contract when you are awarded a position and do not finish until a year.
	4	At one stage, do you have another role?
	5	What do you do for the ANC?

Table 11: Original texts and samples restored by the gating layer-based or cross-attention-based decoders, according to the input text length. Blue texts represent parts different from the original text, and red texts indicates parts omitted from the original text.

Original	1	Occasional diarrhea is a common occurrence.
	2	Most people will experience an episode of diarrhea at least once or twice a year that will disappear in a couple of days.
	3	Luckily, there are many foods to eat that may help a person reduce the symptoms of diarrhea.
	4	There are also some foods to avoid when dealing with a bout of diarrhea, and some additional home care tips to consider.
	5	Anyone who is experiencing persistent diarrhea should see a doctor, as a person may become dehydrated over time.
Gating layer-based decoder		
1 sentence		
Restored	1	Occasional diarrhea is a common occurrence.
3 sentences		
Restored	1	Occasional diarrhea is a common occurrence.
	2	Most people will experience an episode of diarrhea at least twice or twice a year that will disappear in a couple of days.
	3	Luckily, there are many foods to eat that may help a person reduce the symptoms of diarrhea.
5 sentences		
Restored	1	Occupy diarrhea is a common occurrence.
	2	Most people will experience an episode of diarrhea at least once a month or two that will disappear in a week.
	3	Fortunately, there are plenty of ways to eat a food that may help eliminate the symptoms.
	4	There are also some symptoms of diarrhea to avoid eating with a side dish, and some regular food tips that you should consider.
	5	Anyone experiencing chronic diarrhea will be referred to as a woman, but you have a medical problem before.
Cross-attention-based decoder		
1 sentence		
Restored	1	Occasional diarrhea is a common occurrence
3 sentences		
Restored	1	Otago occurrences is an uncommon problem.
	2	Most people will experience (an episode of) a diarrhea of at least one day or two during a month that will disappear in less than a month.
	3	Fortunately, there are many ways to eat foods that can help (a person reduce) the symptoms of a person.
5 sentences		
Restored	1	Occupied diarrhea is a frequent issue.
	2	Many people will experience a severe diarrhea at least once a week 2014 and that may occur in some cases of diarrhea.
	3	Here are a few things that will stop you to consume more of the food to avoid.
	4	There are also a few cases of diarrhea, while people can experience a side effect to avoid experiencing chronic diarrhea.
	5	If an individual is experiencing chronic diarrhea or diarrhea, some people are able to do a handover after that.

Table 12: Original texts and samples restored by the gating layer-based or cross-attention-based decoders, according to the input text length. Blue texts represent parts different from the original text, and red texts indicates parts omitted from the original text.

Original	1	Two bedrooms home on a corner lot.
	2	Two car detached garage.
	3	Nice covered front porch.
	4	Seller will not complete any repairs to the subject property, either lender or buyer requested.
	5	The property is sold in AS IS condition.
Gating layer-based decoder		
1 sentence		
Restored	1	Two bedrooms home on a corner lot.
3 sentences		
Restored	1	Two bedrooms home on a corner lot.
	2	Two car detached garage.
	3	Nice covered front porch.
5 sentences		
Restored	1	Two bedroom home on a corner lot.
	2	Two detached car garage.
	3	Nice covered front porch.
	4	Seller will not complete any repairs to the (subject) property, either insured buyer or seller.
	5	The property is listed in ASOLD condition.
Cross-attention-based decoder		
1 sentence		
Restored	1	Two bedrooms home on a corner lot.
3 sentences		
Restored	1	Two bedroom homes on a corner lot.
	2	Two car detached garage.
	3	Nice covered front porch.
5 sentences		
Restored	1	Two car garage on a corner lot.
	2	Two covered covered porch.
	3	Sony front porch.
	4	Nice covered garage will not return any repairs to the seller, either buyer or seller.
	5	The property is listed in ASOLD condition.

Table 13: Original texts and samples restored by the gating layer-based or cross-attention-based decoders, according to the input text length. Blue texts represent parts different from the original text, and red texts indicates parts omitted from the original text.