Towards Fully Self-Supervised Knowledge Learning from Unstructured Text

Anonymous ACL submission

Abstract

Pre-trained language models (PLMs) like BERT have made significant progress in various downstream NLP tasks. However, by asking models to do cloze-style tests, recent work finds that PLMs are short in acquiring knowledge from the unstructured text. To understand the internal behavior of PLMs in retrieving knowledge, we first define knowledge-baring (K-B) tokens and knowledge-free (K-F) tokens for unstructured text and ask professional annotators to manually label some sample. Then, we find that PLMs are more likely to give wrong predictions on K-B tokens and attend less attention to those tokens inside the self-attention module. Based on these observations, we develop two solutions to help the model learn more knowledge from the unstructured text in a fully self-supervised manner. Experiments on knowledge probing tasks show the effectiveness of the proposed methods. To our best knowledge, we are the first to explore fully self-supervised learning of knowledge in continue pre-training. [1]

1 Introduction

Pre-trained language models (PLMs), such as BERT (Devlin et al., 2019) and GPT (Radford et al., 2018), have greatly improved the performance of many NLP tasks in the past few years. Pre-training has been regarded as a promising way for acquiring common knowledge from unstructured plain text. However, how to learn more knowledge for PLMs is still an unsolved problem (Petroni et al., 2019), especially in those tasks which need explicit usage of knowledge. There are mainly two common ways to enhance the PLMs with more knowledge. One is to introduce structured knowledge bases (Zhang et al., 2019; Wang et al., 2021b) while the other is using unstructured text. Compared with structured knowledge bases, unstructured text is easier to acquire and construct. In addition, with freer format, the unstructured text can express better complex knowledge.

[1]The codes and data will be released upon acceptance.

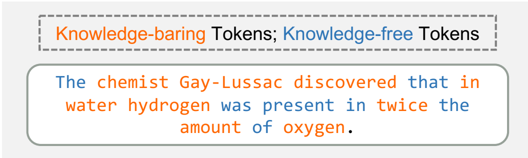

Figure 1: Examples of knowledge-baring (K-B) tokens and knowledge-free (K-F) tokens.

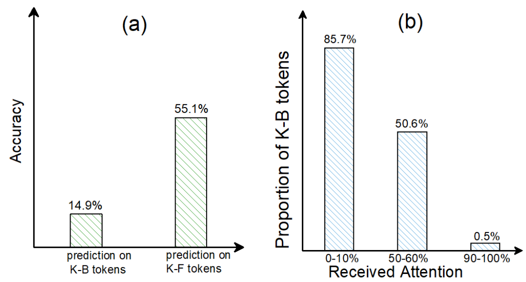

Figure 2: The RoBERTa's behavior on probing samples: (a) the model performs worse on knowledge-baring tokens than on knowledge-free tokens; (b) knowledge-baring tokens are likely to receive less attention in the self-attention process.

We focus on enhancing the ability of PLMs in acquiring knowledge from the unstructured text. First of all, we explore which tokens in the text embodies factual knowledge in a more fine-grained manner (i.e., token-level). This not only help us better understand the model's behavior of memorizing and utilizing knowledge, but also motivates us to design methods for better acquiring knowledge. In particular, for a piece of text, the tokens which are essential for human to understand the text's factual knowledge are considered as *knowledge-baring*; otherwise, they are *knowledge-free*. The example is presented at Figure 1.

We analyze the relationship between behaviors of PLMs and knowledge by manually annotating whether each token in samples is knowledge-baring. As shown in Figure 2 (a), we find that PLMs perform worse on knowledge-baring tokens in the Cloze-style test. In addition, shown in Figure 2 (b), the transformer-based model is likely to gain less attention on knowledge-baring tokens.

Intuitively, for better acquiring knowledge from unstructured text, the model is expected to mask-recover more knowledge-baring words when trained on unstructured text and get less influence from knowledge-free words. To this end, based on our observation, we propose two solutions at the mask policy and attention levels of the PLM: (1) In the mask policy, we have two methods. The first method is to perform random masking on the training corpus before each training iteration and find out which masks the model fails to predict correctly. These incorrectly predicted tokens are regarded as knowledge-baring tokens for masking in this training iteration. The second is that we feed-forward on the training data before each training iteration and use the attention to determine which tokens are more likely to be knowledge-baring for masking. (2) At the attention level, we adopt the visibility matrix to avoid knowledge-free tokens from affecting other tokens during self-attention.

Extensive experiments are conducted on three tasks. Specifically, to check whether the model has learned the knowledge from unstructured text, we let the model perform on the LAMA Probing task, a standard cloze-style test. To test whether the model can utilize the learned knowledge, we also introduce two probing task, namely QA-style Probing and Knowledge Graph Style (KG-style) Probing. Note that there is no labelled data for finetuning for the three tasks, they are only used for probing how much knowledge the model has learned from the unstructured text. Besides, the training corpus contains all needed knowledge of evaluation and testing. The test examples of three tasks are presented in Table 4. Experiments on the three tasks show the effectiveness of the proposed methods, achieving up to 6.1 and 5.5 points absolute improvement in LAMA Probing task on two datasets, up to 6.7 points absolute improvement in QA-style Probing task and 2.6 points absolute improvement in the KG-style probing task.

To our knowledge, we are the first to explore the relationship between PLMs' behavior and knowledge in the token-level and the first to research on fully self-supervised learning of knowledge in continue pre-training.

2 Probing the Behavior of PLMs in Retrieving Knowledge

To better probing how the PLMs learn knowledge from the unstructured text, we start to identify the type and role of each word. Inspired by knowledge graphs as well as our observations, we find that knowledge in a sentence is largely embodied by a few keywords. For the remaining words, even if they are deleted, we can still receive the factual knowledge the sentence conveys.

- **knowledge-baring**: For a given text, if the deletion of one token will make it relatively hard for humans to obtain the factual knowledge contained in the text correctly, we take the token as knowledge-baring;
- **knowledge-free**: For a given text, if the deletion of one token still allows humans relatively easy to obtain the factual knowledge contained in the text correctly, we take the token as knowledge-free.

The example is shown in Figure 1. Note that, Knowledge-Free tokens are not totally free of knowledge, they certainly have some kind knowledge, such as linguistic knowledge and semantic knowledge. They are just relatively less important to the factual knowledge, which we put emphasis on in this work.

We randomly sample 100 cases from the LAMA SQuAD dataset and LAMA Google RE dataset (Petroni et al., 2019), respectively and then use the tokenizer of RoBERTa to tokenize each sentence. We ask three annotators, who are all Ph.D. students, manually label each token as **knowledge-baring** and **knowledge-free**. The inter-annotator agreement for samples of LAMA SQuAD/LAMA Google RE is 0.920/0.938, respectively. The statistic of labelled tokens is shown in Table 1.

We also use the Stanford CoreNLP toolkit (Manning et al., 2014) to conduct part-of-speech tagging analysis on those samples. We find that the most knowledge-baring tokens are nouns(64.2%), verbs(11.6%), numbers(9.2%) and adjective words(6.5%) while most knowledge-free tokens are preposition or subordinating conjunctions(25.1%), comma and punctuation(23.6%), determiners(15.2%) and verbs(11.7%) for the two set of samples. We also put the detailed results in the Appendix Table 10. From the results, we can see that we do not limit the scope of knowledge

	number of Knowledge-Baring Tokens	number of Knowledge-Free tokens
LAMA SQuAD	739	532
LAMA Google RE	1715	975

Table 1: The number of tokens that are knowledge-baring and knowledge-free we have labelled for the samples of the two dataset.

	Knowledge-Baring	Knowledge-Free
RoBERTa-Orig	14.9%	55.1%
RoBERTa-Cont	39.2%	82.8%

(a) On the LAMA SQuAD Samples.

	Knowledge-Baring	Knowledge-Frees
RoBERTa-Orig	38.6%	83.4%
RoBERTa-Cont	67.2%	93.5%

(b) On the LAMA Google RE samples.

Table 2: The probing accuracy on two types of tokens for original model (RoBERTa-Orig) and continued pre-trained model (RoBERTa-Cont) along with the original pre-training mask policy. Both models perform worse on knowledge-baring tokens.

to entities or nouns, we expand it to nouns, verbs, numbers, adjective words and so on.

To better understand the model's behavior on comprehending knowledge, we mainly explore two questions: (1) Does the model perform better on knowledge-baring contents or knowledge-free contents? (2) Can the model's attention scores reveal its association with knowledge?

2.1 Accuracy on Knowledge-Baring and Knowledge-Free Tokens

To investigate the first question, we first mask each token of the sentences in both datasets. For example, if one sentence contains 10 separate tokens, we derive 10 sentences with "<mask>" on each token after processing this sentence. If one word is tokenized to several tokens, we mask those tokens together. The detail is shown in the Table 8 (a) in Appendix. Then, we ask the model to predict the mask(s) of processed sentences.

To better understand the influence of pre-training on model learning knowledge, we use the original PLM as well as the continued pre-trained model to predict on the processed sentences. For continued pre-training, we first find the Wikipedia snippets where the sentences are from and then train the model using the pre-training objective with the snippets for 100 iterations.

The performances of RoBERTa and continued pre-trained RoBERTa on two types of tokens on two datasets are presented in Table 2. From the result, we find that the model performs much worse on knowledge-baring tokens than on knowledge-free tokens, which is 14.9% to 55.1% on SQuAD and 38.6% to 83.4% on Google RE. Even if the model is continued pre-trained, the accuracy of knowledge-baring tokens is still lower than that of K-F tokens, which is 39.2% to 82.8% on SQuAD and 67.2% to 93.5% on Google RE. The results show that it is more difficult for models to learn factual knowledge from the unstructured text than non-knowledge.

2.2 Attention on Knowledge-Baring and Knowledge-Free Tokens

For the second question, we feed-forward the model on the sentences without masking them. For each token, we calculate the sum of all tokens' received attention weights and sum up for all layers and heads. The received attention (RcAtt) weight of token t in the model is

$$RcAtt_t = \sum_{i=1}^{L}\sum_{j=1}^{H}\sum_{k=1}^{N} att_{ijkt} \quad (1)$$

where L is the layer number, H is the head number and N is the token number; att_{ijkt} means in $layer_i$ $head_j$, the attention score $token_k$ to $token_t$.

We sort all the tokens by their RcAtt scores for each sentence and divided them into 10 percent segments. Next, we calculate the proportion of knowledge-baring tokens in each segment. Same as the previous question, we not only use the original PLM to predict, but also test the continued pre-trained model.

The results are presented in Table 3. We can see that the attention scores strongly correlate with whether the tokens are knowledge-baring. The K-B tokens are more likely to receive less attention, while the K-F tokens are more likely to receive more attention. When the model is continued pre-trained, this phenomenon still exists but at a slightly reduced level.

Conclusions. Based on the above two probing experiments, we can conclude that: (1) The PLMs perform worse on knowledge-baring words (i.e., with higher prediction error); (2) The knowledge-baring words are more likely to receive less attention than knowledge-free ones.

	0~10%	10~20%	20~30%	30~40%	40~50%	50~60%	60~70%	70~80%	80~90%	90~100%	Corr*
Original RoBERTa	85.7%	78.9%	72.3%	69.3%	58.1%	50.6%	46.4%	22.4%	5.5%	0.5%	-1.0
RoBERTa-Cont	75.1%	72.8%	64.9%	65.0%	57.4%	53.3%	53.9%	40.7%	9.4%	0.5%	-0.98

(a) On LAMA SQuAD samples

	0~10%	10~20%	20~30%	30~40%	40~50%	50~60%	60~70%	70~80%	80~90%	90~100%	Corr*
Original RoBERTa	97.6%	92.2%	84.7%	75.4%	70.9%	59.1%	53.2%	42.9%	10.6%	4.9%	-1.0
RoBERTa-Cont	81.7%	77.6%	77.3%	75.8%	68.9%	63.4%	61.9%	50.2%	38.1%	5.8%	-1.0

(b) On LAMA Google RE samples

Table 3: The relationship between knowledge-baring proportion (in red) and the level of receiving attention (the first row). The head X-Y% indicates those tokens rank in bottom X-Y% on attention receiving, for example, 0-10% means those tokens receive least attention. The cell with red color is the K-B proportion of those tokens. RoBERTa-Cont is the continued pre-trained RoBERTa. The last column is the the Spearman's rank correlation coefficient between the level of receiving attention and K-B proportion. We can see that tokens receiving more attention are less likely to be K-B.

3 Methods

In this section, we propose two methods based on the conclusion of the above probing experiments, making the PLMs learn more knowledge from unstructured text.

3.1 Backbone Model

We choose the RoBERTa (Liu et al., 2019) model as our baseline model. Moreover, we choose the original pre-training objective of RoBERTa as our baseline. The RoBERTa model is built on the encoder of the Transformer model(Vaswani et al., 2017). For each layer of RoBERTa, it consists of a multi-head self-attention layer and a position-wise feed-forward network. For i_{th} layer, the self-attention output of j_{th} head is

$$A_j = \mathrm{softmax}(\frac{Q_j K_j^T}{\sqrt{d_k}})V_j \quad (2)$$

where d_k is the dimension of Q, K, V vectors.

3.2 Mask Policy

Initially, RoBERTa randomly chooses tokens from the input text to mask. However, recent work (Wang et al., 2021a) shows that it is inefficient to memorize knowledge. Therefore, we aim to enable the model to focus on learning knowledge-baring content. Because we do not provide any label information to the model during training, the model needs to find the K-B tokens from the input text without any supervision.

From the Section 2, we find that the K-B tokens are related to whether the model can accurately predict the token and attention weight the token receive. Hence we provide two corresponding selective mask policies for the model to find and mask the K-B tokens. Note that, the two selective mask policies are mutually exclusive, so we compare their performance rather than combine them.

RoBERTa-Sel-I. Since the model performs much worse on knowledge-baring tokens than on knowledge-free tokens, we can use this feature to find out K-B tokens from the unstructured text. Before each training iteration, we randomly mask some tokens of the training text and predict on the masks, then we **Sel**ect out tokens that are **I**naccurately predicted and treat them as K-B tokens. Besides finding K-B tokens, this policy also helps the model to avoid learning those tokens which it has already learned previously.

RoBERTa-Sel-A. As the knowledge-baring tokens are more likely to receive less attention, we can make use of the attention score each token receives. Before each training iteration, we let the model forward on the non-masked training text, and then we calculate each token's attention weights, which is the same as Eq 1. Next, we **Sel**ect out the tokens that get the least **A**ttention and treat them as K-B tokens.

After finding knowledge-baring tokens, we first randomly mask them and then randomly choose to mask all remaining tokens. For example, we set the first-phase mask language modelling (MLM) probability as 15%, and second-phase MLM probability as 10%, if the text has 100 tokens and we find 20 K-B tokens using one of our methods, we first mask 100×15%=15 tokens from the K-B tokens and then mask 100×10%=10 tokens from the left 85 tokens. The two-phase masks will be combined

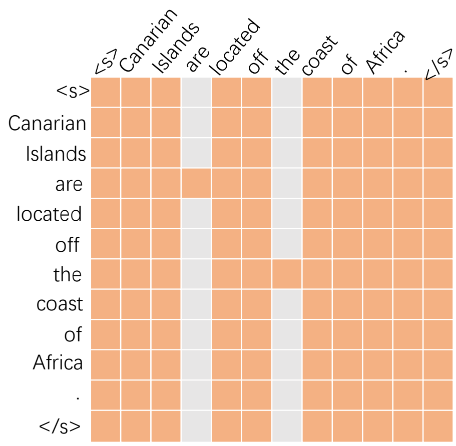

Figure 3: The illustration of the visibility matrix. The orange square means the left token can see the top token while the gray square means it cannot. In this example, the token "are" and "the" are invisible to other tokens.

Algorithm 1 Detecting "harmful" tokens.

Special Dataset Construction:
(1) Forward RoBERTa on the training data
(2) Select the tokens which receive the least 10% attentions

(3) ask the whole words which contain those tokens from the training corpus.
(4) The masked train set is served as the special dataset.
Initialization:
(1) Set a positive real number threshold τ.
(2) Tokenize the special validation data, collect all tokenized tokens that appear more than τ times to a set T.
(4) Add special tokens "<s>", "</s>", "<pad>" and "<mask>" to the set T.
(5) Initialize the set T_n as empty.
(5) Evaluate the model accuracy ACC_0 on the special dataset.
for token t in set T **do**
  (1) Make t invisible to other tokens
  (2) Evaluate the model accuracy ACC on the special dataset.
  if $ACC > ACC_0$ **then**
    Add t to T_n.
  end if
end for
return T_n

for pre-training.

Salient Span Mask (SSM) (Guu et al., 2020) uses a trained NER tagger and a regular expression to identify named entities and date from the raw corpus. These salient spans are selected and masked within a sentence for pre-training. We also conduct the SSM experiments on our dataset as a comparison. But note that the SSM policy is not fully self-supervised because it requires external labelled data to train a NER tagger and prior knowledge to design the expression while our methods are free of any external information and only relied on models themselves.

3.3 Visibility Matrix

In addition to making the model pay more attention to K-B tokens during the continue pre-training, we also consider making the model pay less attention to knowledge-free tokens. To achieve this goal, we adopt the concept of visibility matrix from Dong et al. (2019) and Bao et al. (2020). Using the visibility matrix, we expect those tokens that harm knowledge memorization cannot influence other tokens.

Figure 3 is the illustration of the visibility matrix. During the self-attention process, if token q can attend to token p, in other words, the hidden state of token q can be influenced by the hidden state of token p, we consider token q is visible to token p, otherwise, it is invisible. After adding visibility matrix mechanism to self-attention module, the self-attention output of i layer and j head in Eq 2 is changed to

$$A_j = \text{softmax}\left(\frac{Q_j K_j^T}{\sqrt{d_k}} + M^*\right) V_j \quad (3)$$

where $M^* \in \mathbb{R}^{n \times n}$, $M^*_{pq} = -\infty$ if token q is visible to token p and $M^*_{pq} = 0$ if token q is invisible to token p.

By conducting pilot experiments on making manually chosen irrelevant tokens invisible by other tokens, we find it effective to boost performance on the three tasks. So, we continue to design an algorithm to find the tokens that may harm the performance. Since the training data does not have any label, we construct a special dataset from the training data to find the "harmful" tokens. The algorithm is presented in Algorithm 1. For each time, we make one token invisible, and check whether it will help the evaluation performance on the special dataset.

4 Tasks

Note that, there are three main differences between the proposed visibility matrix and the mask matrix used in recent works (Dong et al., 2019; Bao et al., 2020): 1) The visibility matrix is independent on input masks while mask matrix only make the masked tokens invisible; 2) We have designed an automated algorithm to search invisible tokens rather than by random masking; 3) The invisible

LAMA Probing	
Train Text: ... Kenya ranks low on Transparency International's Corruption Perception Index (CPI), a metric which attempts to gauge the prevalence of public sector corruption in various countries. ...	
Test Query: On the CPI scale, Kenya ranks <mask>.	**Test Answer**: low
QA-style Probing	
Train Text: ... The capital of the Ottoman empire was Istanbul....	
Test Query: What was the capital of the Ottoman empire? <mask>	**Test Answer**: Istanbul
KG-style Probing	
Train Text: Shlomo Shriki, Israeli painter and artist, born in Morocco (1949), grew up and was educated in Kibbutz Yifat.	
Test Query: Shlomo Shriki, place of birth, <mask>	**Test Answer**: Morocco

Table 4: Examples of three tasks. The training text are all unstructured text and label-free. In validation/test, the model need to predict on the <mask> token.

	Training Passages	Validation Queries	Testing Queries
LAMA Probing (LAMA SQuAD)	271	152	152
LAMA Probing (LAMA Google RE)	5516	2758	2758
QA-style Probing	271	152	152
KG-style Probing	5516	2206	2205

Table 5: The statistics of three tasks (four datasets).

tokens can still see themselves while the tokens in mask matrix cannot.

We adopt three tasks to evaluate the usage of knowledge from the unstructured text in this work: LAMA probing, QA-style Probing, and Knowledge Graph (KG) Reasoning. The examples of the three tasks are presented in Table 4. These tasks are slightly different from the ordinal machine learning tasks, as the training data and evaluation/test data have different formats.

We use the **LA**nguage **M**odel **A**nalysis Probing (Petroni et al., 2019) task to *directly* evaluate how much knowledge can PLM obtain from unstructured text. For each example, the training case contains a passage and the validation/test case contains a cloze-style query and answer pair. The model needs to learn knowledge from training passages and use the knowledge to fill the "<mask>" tokens in the validation/test cloze-style sentences.

We use the QA-style Probing task and the Knowledge Graph style (KG-style) Probing task to testify whether the PLM can utilize its learned knowledge in downstream tasks.

For each sample in the **QA-style Probing** task, the training case contains a sentence, while the validation/test case contains a cloze-style QA pair, whose question has one or several "<mask>" tokens after the "?". The needed knowledge of validation/test questions is in the training sentences. The model needs to learn knowledge from training sentences and use the knowledge to fill the "<mask>" tokens in the validation/test cloze-style questions. For each sample in the **KG-style Probing** task, the training case contains a sentence, while the validation/test case contains a cloze-style triple, whose object is replaced with one or several "<mask>" tokens. The needed knowledge of validation/test triples is in the training sentences. The model needs to learn knowledge from training sentences and use the knowledge to fill the "<mask>" tokens in the validation/test cloze-style triples. To make the model adapt to the cloze-style triples answer, for 20% training sentences, we add the corresponding triple at the end of each sentence and remove the triple from the validation/test set.

Data. The task data originate from public released datasets. For the LAMA SQuAD dataset, we link the probes to SQuAD1.1 dataset (Rajpurkar et al., 2016) and find the related questions and passages of each case. Then we use the passages as training data and probes as the validation/test data to construct the dataset for LAMA Probing task. Moreover, we use the recovered probing sentences as the training data and the questions concatenated with "<mask>" as the validation/testing data for the QA-style Probing task. For the LAMA Google RE dataset, we use the snippet of each case as training data and probe sentences as the validation/test data for the LAMA Probing task. Furthermore, we use passages as the training data and use the <subject, relation, object> triples as the validation/test data for the KG-style Probing task.

Note that, for the three tasks, all needed knowledge of validation and test questions can be directly extracted from the training set.

For each task and dataset, we use Algorithm 1 to find "harmful" tokens automatically. In practice, we use the original RoBERTa-large model or the continued pre-trained RoBERTa-large model to evaluate. After finding those tokens, we make them invisible to all other tokens during training, validation and testing periods. An example of the processed visibility matrix is shown in Figure 3.

5 Experiments

Settings. We adopt the RoBERTa-large model as our base model, and conduct continued pre-training on it. We follow most of the traditional pre-training hyper-parameters of RoBERTa (Liu et al., 2019), such as training batch size, optimization method and model configurations. However, some specific parameters are modified when applying our methods. We present needed hyper-parameters at Section A in the Appendix.

5.1 Overall Results

Table 6 shows the results on three tasks. Specifically, the LAMA probing task is used to *explicitly* evaluate how much knowledge is stored from the unstructured text. Moreover, the QA-style Probing and the KG-style Probing tasks are used to *explicitly* validate the model's ability in making use of knowledge on the other formats.

Firstly, we investigate the masking policy (Section 3.2) in continued pre-training. It can be found that our proposed two selective mask policies (RoBERTa-Sel-I and RoBERTa-Sel-A) outperform the original random mask policy (RoBERTa-Cont), obtaining up to 6.1/5.1, 6.5 and 1.4 absolute improvement on three tasks, respectively. It indicates that our methods can enhance the RoBERTa with more domain specific knowledge in the continued pre-training process.

Furthermore, we find that model trained with Visibility Matrix (VM) mechanism (Section 3.3) can substantially achieve better accuracy. For example, RoBERTa-Cont-VM outperforms RoBERTa-Cont by 4.9/4.4, 5.5 and 1.7 absolute gains on three tasks, respectively. Since RoBERTa-Sel-I is superior to RoBERTa-Sel-A on two tasks and three datasets, we further only present the results of RoBERTa-Sel-I combined with the Visibility Matrix mechanism. The combination of selective mask policy Sel-I and visibility matrix (RoBERTa-Sel-I-VM) performs best in the LAMA Google RE, QA-style Probing and KG-style Probing.

Finally, we observe that at the same continued pre-training iterations, our models generally give higher accuracy than RoBERTa-Cont on all tasks, showing that our methods can also benefit in the efficiency of learning knowledge. In addition, though SSM introduces external tools (a trained NER tagger) and prior knowledge (expression to identify dates), our methods performs better than it. It is mainly because SSM only mask entities while leaves other kinds of tokens, which are also important for knowledge probing in the two task. SSM outperforms our methods on KG-style Probing, it is natural since KG-style Probing queries contain only entities and relations.

5.2 On Knowledge-Baring Tokens

We also evaluate the continued pre-training on K-B tokens to see whether the improvement comes from the model's better understanding of K-B tokens. The evaluation data statistic is shown in Table 8 (a) in Appendix.

The results are presented in Table 7. From this table, we can see that our methods can help model better comprehend K-B tokens, showing that the overall better results in Table 6 comes models' comprehension of K-B tokens.

5.3 Discovery on Invisible Tokens

We find that the three tokens “<s>”, “</s>” and “.” receiving much attention, consistently ranking on the top 20% in one piece of text. However, if we make one or more of them invisible to other tokens, the performance on the three tasks will decrease by at least 5 points. Though they cannot be viewed as knowledge-baring tokens, they are still crucial for knowledge learning. We hypothesize they can store the general knowledge information of the text.

6 Related Work

Continue Pre-training of PLMs. Gururangan et al. (2020) reveals that continued pre-training on specific domains will contribute to the performance in downstream tasks within the same domains, and continued pre-training on some task's input data will also boost the performance on those datasets. Guu et al. (2020) proposed Salient span masking (SSM) which is using a NER tagger and rules to detect named entities and date, then they mask at least one salient span each time when pretraining. On the contrary, we do not introduce any external information or prior knowledge to determine masks. Gu et al. (2020) first uses the training pairs of downstream tasks to help continue-pretrain a PLM. They find which tokens deleted from the

	LAMA SQuAD	LAMA Google RE	QA-style Probing	KG-style Probing
RoBERTa-Orig	16.4	24.6	0.0	2.6
RoBERTa-Cont	33.6 (+0.0)	58.4 (+0.0)	37.9 (+0.0)	28.1 (+0.0)
RoBERTa-SSM	37.5 (+3.9)	62.6 (+4.2)	42.7 (+4.8)	**31.2 (+3.1)**
RoBERTa-Sel-A	35.9 (+2.3)	62.4 (+4.0)	44.4 (+6.5)	27.7 (-0.4)
RoBERTa-Sel-I	**39.7 (+6.1)**	63.5 (+5.1)	43.6 (+5.7)	29.5 (+1.4)
RoBERTa-Cont-VM	38.5 (+4.9)	62.8 (+4.4)	43.4 (+5.5)	29.6 (+1.7)
RoBERTa-Sel-I-VM	37.2 (+3.6)	**63.9 (+5.5)**	**44.8 (+6.7)**	30.7 (+2.6)

Table 6: The accuracy on three knowledge intensive tasks. The first block denotes the results of original and continued pre-trained RoBERTa. The second and third blocks show the performance of improved models in terms of **Sel**ective mask policy (Section 3.2) and **V**isibility **M**atrix (Section 3.3). The numbers in brackets show the absolute improvements compared to the continued pre-trained RoBERTa.

	LAMA-SQuAD	LAMA-Google RE
RoBERTa-Orig	13.9%	38.6%
RoBERTa-Cont	38.4%	67.2%
RoBERTa-Sel-A	41.8%	71.4%
RoBERTa-Sel-I	42.6%	71.6%
RoBERTa-Cont-VM	41.9%	71.0%

Table 7: The probing results on the annotated knowledge-baring tokens.

input of task's training data will influence the confidence of prediction of the finetuned model, and they focus on masking those tokens when continue pre-training. Ye et al. (2021) proposed a two-loop meta-learned policy in continue pre-training BART for Closed-book QA Tasks, Knowledge-Intensive Tasks (Petroni et al., 2021) and abstractive summarization. They first continue pretrain the BART with a passage and second train it with a (q,a) pair, then they use the validation loss on the pair to update the parameters of mask policies. There are two main differences between our work and the above two works. First, their works use labelled datasets to help continue pre-training, while our work does not use any labelled data. Second, they conduct continue pre-training for specific tasks, such as Closed-book QA and sentiment analysis, we aims for general knowledge learning and directly test how much knowledge the model has learned, not for specific downstream tasks.

Knowledge Probing in PLMs. LAMA (LAnguage Model Analysis) probe (Petroni et al., 2019) first uses the Cloze-style test to evaluate how much knowledge in the PLMs, they manually transfer some questions of SQuAD (Rajpurkar et al., 2016) and some triples of Google RE, T-REx (Elsahar et al., 2018) and ConceptNet (Liu and Singh, 2004) to Cloze-style prompts. In this work, we create two variants x on LAMA probing and use the LAMA probing test and the variants to evaluate how much knowledge the model has learned from the unstructured text.

Despite increasing research in knowledge and PLMs, relatively less work associate knowledge from text with testing questions. Roberts et al. (2020) and Fedus et al. (2021) use a set of query&answer (QA) pairs to finetune the model and use another set of QA pairs to test it, which have no explicit correlation with pre-training data. We cannot exactly know whether the model learn from the training data or just solve questions by overlap between the finetuning data and test data (Lewis et al., 2021) or simply by spurious cues (Niven and Kao, 2019). In contrast, we impose restrictions on the continued pre-training data and the test questions as well as get rid of finetuning process to ensure the model can only acquire needed knowledge from the training data.

7 Conclusion

We probe the behavior of the pre-trained language models on the unstructured text about the knowledge-baring and knowledge-free tokens, by asking those models to do the cloze-style test on our annotated data. We find that: (1) The model performs worse on K-B tokens; (2) The model gathers less attention on K-B tokens. To enable the model to better acquire knowledge from the unstructured text, we consider two selective mask policies and adopt the visibility matrix mechanism to help the model focus on K-B tokens when learning from the unstructured text. To our knowledge, we are the first to explore fully self-supervised learning of knowledge in continued pre-training.

8 *Ethics / Impact Statement

Our used data is processed from open source datasets, including LAMA SQuAD / LAMA Google RE [2] and SQuAD 1.1 [3].

References

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, and Hsiao-Wuen Hon. 2020. UniLMv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 642–652. 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.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

W. Fedus, Barret Zoph, and Noam M. Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *ArXiv*, abs/2101.03961.

Yuxian Gu, Zhengyan Zhang, Xiaozhi Wang, Zhiyuan Liu, and Maosong Sun. 2020. Train no evil: Selective masking for task-guided pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6966–6974, Online. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Patrick Lewis, Pontus Stenetorp, and Sebastian Riedel. 2021. Question and answer test-train overlap in open-domain question answering datasets. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1000–1008, Online. Association for Computational Linguistics.

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. *CoRR*, abs/1907.07355.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

[2] `https://dl.fbaipublicfiles.com/LAMA/data.zip`

[3] `https://rajpurkar.github.io/SQuAD-explorer/`

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. 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.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

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, pages 5998–6008. Curran Associates, Inc.

Cunxiang Wang, Pai Liu, and Yue Zhang. 2021a. Can generative pre-trained language models serve as knowledge bases for closed-book QA? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3241–3251, Online. Association for Computational Linguistics.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2021b. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1405–1418, Online. Association for Computational Linguistics.

Qinyuan Ye, Belinda Z. Li, Sinong Wang, Benjamin Bolte, Hao Ma, Wen-tau Yih, Xiang Ren, and Madian Khabsa. 2021. On the influence of masking policies in intermediate pre-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7190–7202, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of ACL 2019*.

	#Sentences	#Masked Tokens
LAMA SQuAD (Knowledge-Baring)	609	739
LAMA SQuAD (Knowledge-Free)	524	532
LAMA Google RE (Knowledge -baring)	1268	1715
LAMA Google RE (Knowledge -free)	865	975

(a) Data after every token of the 200 samples is masked separately, which is used for accuracy analysis in Section 2.1.

	#Sentences	# Tokens
LAMA SQuAD	100	1471
LAMA Google RE	100	2903

(b) Data used for attention analysis in Section 2.2.

Table 8: Data statistics after the 200 samples are processed for analysis in Section 2.1 and Section 2.2.

Hyperparam	
Learning Rate	1e-4
Train Batch Size	256 (passages)
MLM propability	0.15
Max Tokens Length	512
Optimizer	Adam
Adam ϵ	1e-6
Adam β_1	0.9
Adam β_2	0.98
Weight Decay	0.01
Learning Rate Decay	Linear

Table 9: The hyper-parameters for continue pre-training RoBERTa in this work.

A Hyper-parameters

The traditional hyper-parameters for continue pre-training RoBERTa can be seen at Table 9.

Moreover, for RoBERTa-Sel-I and RoBERTa-Sel-A, we set the first-phase MLM probability as 15% and the second-phase MLM probability as 10%. For the RoBERTa-SSM, we adopt a publicly released NER model, which is based on RoBERTa-base and trained on the conll2003 dataset, [4] and a regular expression to identify named entities and date, respectively. In the LAMA Probing task, all models are trained for 100 iterations. For the visible mechanism, we use the original RoBERTa-large to find the knowledge-free tokens. In the QA-style Probing task, models are trained for 500 iterations. For the visible matrix mechanism, we set τ as 3 for the two datasets.

[4] huggingface.co/andi611/roberta-base-ner-conll2003

B Details of POS analysis on Samples

We present a detailed results of part-of-speech tagging analysis of annotated samples in Table 10.

C Mask Analysis

To compare three different mask policies, namely RoBERTa-Cont, RoBERTa-Sel-I and RoBERTa-Sel-A we conduct 10-iteration continue pre-training on the 200 samples in Section 2 and record their masked tokens.

Then We take part-of-speech analysis on the mask tokens for the three mask policies, which is presented in Table 11. From the result, we can see that our two selective mask policies choose more nouns, numbers, verbs and adjective words to mask than the random mask policies.

We also calculate the K-B / K-F ratio of masked tokens for the three mask policies and list the result in Table 12. From the table, it can be seen that our two selective mask policies can significantly increase the proportion of K-B tokens in the masked tokens.

Knowledge-Baring Tokens	NN 243 0.329	NNP 179 0.242	JJ 68 0.092	NNS 49 0.066	VBN 46 0.062	VBD 21 0.028	CD 20 0.027	VBZ 19 0.026	VB 15 0.02	IN 14 0.019	POS 8 0.011	RB 7 0.009	NNPS 6 0.008	VBP 5 0.007	VBG 5 0.007
Knowledge-Free Tokens	IN 149 0.28	DT 109 0.205	. 100 0.188	VBZ 39 0.073	VBD 31 0.058	TO 14 0.026	, 13 0.024	CC 9 0.017	VB 8 0.015	RB 8 0.015	WDT 7 0.013	PRP$ 6 0.011	VBP 5 0.009	WRB 4 0.008	MD 4 0.008

(a) In LAMA SQuAD samples

Knowledge-Baring Tokens	NNP 657 0.383	NN 419 0.244	CD 205 0.12	VBN 102 0.059	JJ 91 0.053	IN 38 0.022	VBD 36 0.021	NNS 23 0.013	FW 22 0.013	PRP 19 0.011	VBG 14 0.008	DT 12 0.007	VBZ 11 0.006	VBP 10 0.006	RB 9 0.005
Knowledge-Free Tokens	IN 230 0.236	, 143 0.147	DT 113 0.116	. 110 0.113	CC 58 0.059	-RRB- 58 0.059	-LRB- 58 0.059	VBD 53 0.054	VBZ 40 0.041	HYPH 26 0.027	: 18 0.018	RB 16 0.016	WP 11 0.011	PRP$ 11 0.011	WRB 4 0.004

(b) In LAMA Google RE samples

Table 10: Part-of-speech Results on our annotated samples. For each cell, the tag name is at the top, the number of this tag is in the middle, the proportion of this tag is in the bottom. For each type of token in each data set, we only display the top-15 tags.

RoBERTa-Cont (Random)	NNP 0.206	NN 0.151	IN 0.117	DT 0.067	CD 0.054	JJ 0.048	, 0.047	VBN 0.045	VBD 0.043	. 0.042	VBZ 0.031	NNS 0.021	CC 0.019	-RRB- 0.015	RB 0.013
RoBERTa-Sel-I	NNP 0.24	NN 0.181	IN 0.097	CD 0.073	DT 0.051	JJ 0.047	VBN 0.038	VBD 0.037	, 0.034	. 0.033	VBZ 0.024	NNS 0.023	CC 0.016	-RRB- 0.012	RB 0.012
RoBERTa-Sel-A	NNP 0.265	NN 0.167	IN 0.112	CD 0.086	JJ 0.049	DT 0.048	VBN 0.033	, 0.026	VBD 0.024	. 0.021	VBZ 0.021	NNS 0.018	CC 0.013	-LRB- 0.012	RB 0.012

Table 11: The result of part-of-speech analysis for three mask policies. For each cell, the tag name is at the top and the proportion of this tag is in the bottom.

Method	K-B / K-F ratio
RoBERTa-Cont (Random)	1.47:1
RoBERTa-Sel-I	2.16:1
RoBERTa-Sel-A	2.33:1

Table 12: The K-B to K-F ratios for three mask policies. The experiment is conducted on the samples which are annotated with K-B and K-F.