Pre-training Language Models with Deterministic Factual Knowledge

Shaobo Li[1], Xiaoguang Li[2], Lifeng Shang[2],
Chengjie Sun[1], Bingquan Liu[1], Zhenzhou Ji[1], Xin Jiang[2] and Qun Liu[2]
[1]Harbin Institute of Technology, [2]Huawei Noah's Ark Lab
shli@insun.hit.edu.cn, {sunchengjie, liubq, jizhenzhou}@hit.edu.cn
{lixiaoguang11, shang.lifeng, Jiang.Xin, qun.liu}@huawei.com

Abstract

Previous works show that Pre-trained Language Models (PLMs) can capture factual knowledge. However, some analyses reveal that PLMs fail to perform it robustly, e.g., being sensitive to the changes of prompts when extracting factual knowledge. To mitigate this issue, we propose to let PLMs learn the deterministic relationship between the remaining context and the masked content. The deterministic relationship ensures that the masked factual content can be deterministically inferable based on the existing clues in the context. That would provide more stable patterns for PLMs to capture factual knowledge than randomly masking. Two pre-training tasks are further introduced to motivate PLMs to rely on the deterministic relationship when filling masks. Specifically, we use an external Knowledge Base (KB) to identify deterministic relationships and continuously pre-train PLMs with the proposed methods. The factual knowledge probing experiments indicate that the continuously pre-trained PLMs achieve better robustness in factual knowledge capturing. Further experiments on question-answering datasets show that trying to learn a deterministic relationship with the proposed methods can also help other knowledge-intensive tasks.

1 Introduction

Petroni et al. (2019); Jiang et al. (2020); Shin et al. (2020); Zhong et al. (2021) show that we can successfully extract factual knowledge from Pre-trained Language Models (PLMs) using cloze-style prompts such as "The director of the film Saving Private Ryan is [MASK]." Some recent works (Cao et al., 2021; Pörner et al., 2020) find that the PLMs may rely on superficial cues to achieve that and can not respond robustly. Table 1 gives examples of inconsistent predictions exposed by changing the surface forms of prompts on the same fact.

This phenomenon questions whether PLMs can robustly capture factual knowledge through Masked Language Modeling (MLM) (Devlin et al., 2018) and further intensify us to inspect the masked contents in the pre-training samples. After reviewing several masking methods, we find that they focus on limiting the granularity of masked contents, e.g., restricting the masked content to be entities and then randomly masking the entities (Guu et al., 2020), and pay less attention to checking whether the obtained MLM samples are appropriate for factual knowledge capturing. For instance, when we want PLMs to capture the corresponding factual knowledge as recovering the masked entities, we should check whether the remaining context provides sufficient clues to recover the missing entity.

Cloze-style Prompt and Prediction	**Is Correct?**
War Horse is an American war film directed by Steven Spielberg.	✓
The director of the American war film War Horse is Keanu Reeves.	✗
Christopher Nolan is the director of the American war film War Horse.	✗

Table 1: A PLM could gives inconsistent results when probing the same fact with different prompts. The underlined words are the predictions.

Inspired by the above analysis, we can categorize MLM samples based on the relationship between the remaining context and masked content:

- **Non-deterministic samples** The clues in the remaining context are insufficient to constrain the value of the masked content. Multiple values are valid to fill in the masks.
- **Deterministic samples** The remaining context holds deterministic clues for the masked content. We can get one and only one valid value for the masked content.

For example, the first cloze in Table 1 masks the director of the film "*War Horse*." Since the film has only one director in the real world, we can get a unique answer deterministically. So it is a deterministic MLM sample. The crucial clues "*War Horse*" and "*directed by*" have a deterministic rela-

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 11118–11131
December 7-11, 2022

tionship with the missing entity "*Steven Spielberg*." For brevity, we refer to these clues as **deterministic clues** and the outcome "*Steven Spielberg*" as **deterministic span**. In contrast, if the sample becomes "[MASK]s *is an American war film directed by Steven Spielberg*," multiple names can fill the masks because Steven Spielberg produced more than one American war film. We cannot tell which one is better based on the existing clues, so it is a non-deterministic sample.

The non-deterministic samples establish a multi-label problem (Zhang and Zhou, 2006) for MLM, where more than one ground-truth value for outputs is associated with a single input. If we enforce the PLMs to promote one specified ground truth over others, the other ground truths become false negatives that could plague the training or cause a performance downgrade (Durand et al., 2019; Cole et al., 2021). The non-deterministic samples are competent for obtaining contextualized representations but become questionable for understanding the intrinsic relationship between factual entities. In contrast, the deterministic samples are less confusing since the answer is always unique, providing a stable relationship for PLMs to learn.

Therefore, we propose **deterministic masking** that always masks and predicts the deterministic spans in MLM pre-training to improve PLMs' ability to capture factual knowledge. The deterministic clues and spans are identified based on a KB. Two pre-training tasks, **clue contrastive learning** and **clue classification**, are introduced to make PLMs more aware of the deterministic clues when predicting the missing entities. The clue contrastive learning encourages PLMs to be more confident in prediction (Vu et al., 2019; Luo et al., 2021) when the deterministic clues are unmasked. The clue classification is to detect whether the remaining context contains deterministic clues. The experiments on the factual knowledge probing and question-answering tasks show the effectiveness of the proposed methods.

The contributions of this paper are: (1) We propose to model the deterministic relationship in MLM samples to improve the robustness (i.e., both consistency and accuracy) of factual knowledge capturing. (2) We design two pre-training tasks to enhance the deterministic relationship between entities to earn further improvement on robustness. (3) The experiment results show that learning the deterministic relationship is also helpful for other knowledge-intensive tasks, such as question answering.

2 Methods

Section 2.1 expatiates the deterministic masking, which includes how we align texts with triplets and identify deterministic clues and spans in texts. The clue contrastive learning and clue classification are described in Sections 2.2 and 2.3, respectively.

2.1 Deterministic Masking

In addition to masking only factual content, the deterministic masking also constrains the remaining context and the masked content to have a deterministic relationship: the remaining context should provide conclusive clues to predict the masked content, and the valid value to fill in the mask is unique.

To this end, we align each text with a KB triplet and match the spans in the text with (*subject*, *predicate*, *object*) respectively. We select the spans aligned with *object*s as the candidates to be masked for pre-training. To further make the masked object deterministic, we query the KB with the aligned (*subject*, *predicate*) and check whether the valid object that exists in KB is unique.

If the KB emits this object exclusively, e.g., only the aligned object can compose a valid triplet with the aligned *subject* and *predicate*, the object is deterministic. The object is non-deterministic if multiple objects suit the aligned subject and predicate in the KB. The span aligned with the deterministic object is a deterministic span, and it would be masked to construct a deterministic MLM sample[1]. We pre-train PLMs on only the deterministic samples.

Figure 1 shows a deterministic sample aligned with the triplet ("*War Horse*," "*directed by*," "*Steven Spielberg*"). When querying KB with "*War Horse*" as the subject and "*directed by*" as the predicate, the result object "*Steven Spielberg*" is unique because there is only one director who produced this film, so the first sample is deterministic. In contrast, when using "*Steven Spielberg*" and "*directer of*" as the subject and the predicate, multiple valid objects exist in KB, so the second sample is non-deterministic and is filtered out.

By dropping the non-deterministic samples, we prevent PLMs from having a crush on one object but ignoring others that are also valid based on the

[1]We put the detailed procedure (includes entity linking and predicate string matching) in Appendix B.

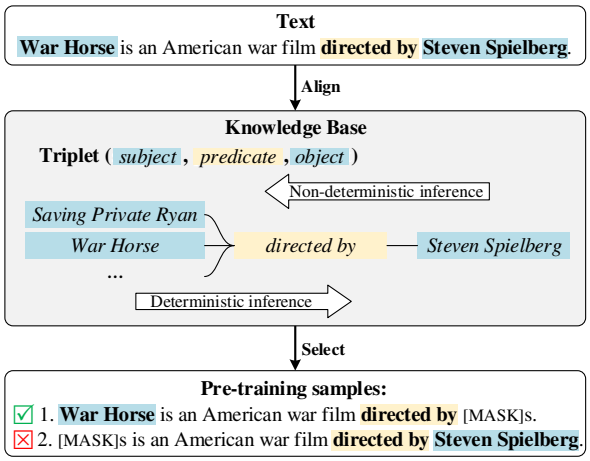
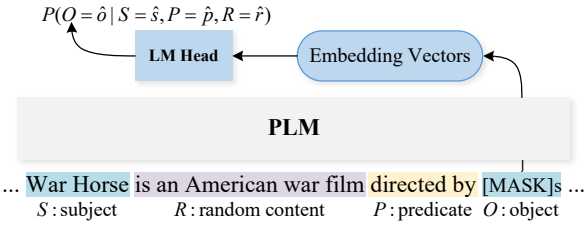
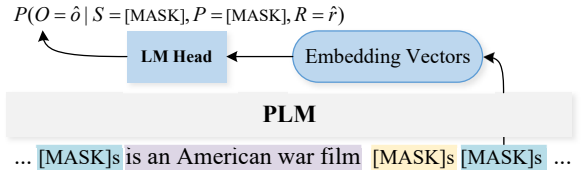


Figure 1: Construct a deterministic sample. The spans with blue background correspond to entities (subject or object), and the spans with yellow describe relations (predicate).



(a) Deterministic sample: masks the deterministic span (object) and keeps the deterministic clues (object and predicate).



(b) Contrastive sample without deterministic clues: masks both the deterministic clues and the deterministic span.

Figure 2: The two samples in clue contrastive learning. The first sample (a) has a more informative context, so PLM should be more confident when predicting the masked object O. The texts with purple background denote the spans other than entities and relations.

existing clues. While in the deterministic samples, the relationship between the remaining clues and the missing span is more stable and unambiguous. Training on the deterministic samples encourages PLM to infer the missing object based on its deterministic factual clues. It helps PLMs grasp a more substantial relationship between entities to model the factual contents and could aid in accomplishing some knowledge-intensive tasks.

2.2 Clue Contrastive Learning

To stimulate PLMs to catch the deterministic relationship between entities, we design the pre-training task clue contrastive learning following this intuition: PLMs should have more confidence to generate a masked span when its deterministic clues exist in the context, and introduce a contrastive objective accordingly. We explain it with a pair of samples in Figure 2. Figure 2a shows a deterministic MLM sample that masks the span "*Steven Spielberg*" and keeps its deterministic clues. Figure 2b masks both the deterministic clues and the deterministic span. The remaining context in Figure 2a contains fewer [MASK]s and provides more information, naturally reducing the uncertainty in prediction. So PLMs should assign a higher probability for the ground truth when giving the context in Figure 2a than Figure 2b.

Formally, we use S and P to denote the deterministic clues (subject and predicate) and O to denote the masked deterministic span (object). R represents the random spans in the context other than S, P, and O. The objective function that needs to be maximized is:

$$P(O=\hat{o} \mid S=\hat{s}, P=\hat{p}, R=\hat{r}) \\ -P(O=\hat{o} \mid S=[\text{MASK}], P=[\text{MASK}], R=\hat{r}), \quad (1)$$

$S=[\text{MASK}]$ and $P=[\text{MASK}]$ denote replacing the deterministic clues with [MASK]s. $\hat{s}$, $\hat{p}$, $\hat{o}$ and $\hat{r}$ are the ground-truth values of the S, P, O and R, respectively. $P(O=\hat{o} \mid \cdot)$ denotes the probability that the PLM correctly predicts the masked span O, i.e., the average probability that the PLM assigns to the ground-truth tokens. It is calculated by a Language Model Head (LM Head) based on the embedding of O from the PLMs.

This task encourages PLMs to give the ground truth $\hat{o}$ a higher probability when the deterministic clues exist in the context. It is somewhat conservative since we consider the noise in the data construction. The objective is still reasonable even when the S, P, and R are randomly labeled. Raw words are always more informative than the ordinary [MASK]s and can reduce the uncertainty of the context (Cover, 1999), so the uncertainty of prediction degrades accordingly (Vu et al., 2019; Luo et al., 2021). On the other hand, this objective trains PLMs to react to the changes in the context, i.e., learning how to tune the output as the input changes. We employ a large-scale KB as the approximation of real-world knowledge (Reiter, 1981) to get the pre-training samples.

2.3 Clue Classification

The clue classification asks PLMs to classify what kinds of clues exist in the remaining context. After masking the deterministic span O, we manipulate the remaining context to generate three samples that contain different kinds of contexts:

(a) **Keep deterministic clues:** we only mask the deterministic span O and leave its deterministic clues untorched. It is the same as the original deterministic MLM sample shown in Figure 2a.

(b) **Mask deterministic clues:** we mask O and its deterministic clues (S and P). It is the same as the constructive sample in Figure 2b.

(c) **Mask random spans:** we mask O and some random spans R other than the deterministic clues. An example is shown in Figure 3. The number of tokens in R is the same as the number of tokens in the deterministic clues.

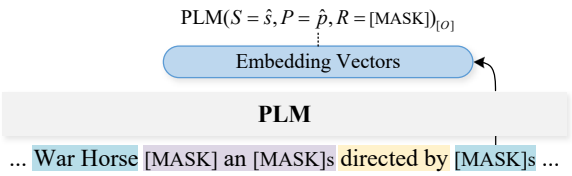


Figure 3: A sample that masks some random spans R (with purple background) in the context.

The PLMs that use the Transformer encoder (Vaswani et al., 2017) as the backbone emit a contextualized embedding for each input token. Every contextualized embedding can encode all the information in context since all the input tokens are involved when computing the embedding. So we encode the clues in the remaining context using the contextualized embedding of O. Formally, the clue representations for the above three samples are:

$$\begin{aligned} E_{(a)} &= \text{PLM}(S = \hat{s}, P = \hat{p}, R = \hat{r})_{[O]}, \\ E_{(b)} &= \text{PLM}(S = [\text{MASK}], P = [\text{MASK}], R = \hat{r})_{[O]}, \\ E_{(c)} &= \text{PLM}(S = \hat{s}, P = \hat{p}, R = [\text{MASK}])_{[O]}, \\ E_{(a)}, E_{(b)}, E_{(c)} &\in \mathbb{R}^{|O| \times d}. \end{aligned} \tag{2}$$

$E_{(a)}$, $E_{(b)}$, and $E_{(c)}$ represent the inputs that (a) keep deterministic clues S and P, (b) mask deterministic clues, and (c) mask random spans R, respectively. $\text{PLM}(\cdot)$ denotes the PLM that can output the contextualized embedding for each input token. $\text{PLM}(\cdot)_{[O]}$ denotes grabbing the embedding vectors corresponding to O from the PLM's output. Since the three input samples have the same O, the number of token-level embedding vectors is the same in $E_{(a)}$, $E_{(b)}$, and $E_{(c)}$.

Each token-level embedding vector $\mathbf{e}$ in $E_{(a)}$, $E_{(b)}$, and $E_{(c)}$ is fed into a three-way classifier:

$$y = \text{softmax}(\mathbf{W}^\top \mathbf{e}), \mathbf{e} \in \mathbb{R}^d, \mathbf{W} \in \mathbb{R}^{d \times 3}. \tag{3}$$

y is a three-element vector and shows the probabilities that $\mathbf{e}$ comes from $E_{(a)}$, $E_{(b)}$ and $E_{(c)}$. $\mathbf{W}^\top$ is the three-way classifier.

The number of masked tokens in samples (a) and (b) is different since the latter masks the clues additionally. It may become a shortcut for the proposed contrastive or classification tasks. So we introduce sample (c), which has the same number of masks as sample (b), to eliminate this shortcut. Some existing pre-training methods (Clark et al., 2019; Xiong et al., 2019; He et al., 2020) replace original tokens with fake tokens to build the pseudo samples and then classify between original and pseudo samples, while clue classification employs [MASK]s in the replacement. We wonder that fake tokens may make the intervened input tell fake facts conflict with the real world, leading the PLMs to capture wrong knowledge from the pseudo samples. [MASK] is a safer choice here.

3 Experiments

3.1 Pre-training Data

We use Wikipedia[2] as the source of texts and Wikidata[3] as the knowledge base. We split the Wikipedia texts into natural paragraphs and then align each paragraph with subject-predicate-object triplets in WikiData. Each aligned object that is deterministic based on WikiData is the deterministic span, and all the subjects and predicates that correspond to the deterministic span are deterministic clues. The paragraphs with identified deterministic spans and clues are then used for pre-training.

We first employ the TREX (Elsahar et al., 2018) that provides the alignments between texts and triplets to construct a preliminary dataset named **Partial data**. About 46.1% of triplets are non-deterministic and ignored in the partial data. TERX provides aligned triplets for only the abstract paragraphs (first paragraph) in Wikipedia. We enlarge the data size by processing all the paragraphs in Wikipedia. The detailed process is in Appendix B. The dataset that involves all the paragraphs is referred to as **Full data**. Table 2 shows the statistics

[2]https://www.wikipedia.org/
[3]https://www.wikidata.org/

of the two pre-training datasets. For efficiency reasons, we use the partial data to train the baselines in the ablation study. The full data is for our final model.

	Partial data	Full data
# Paragraphs	3,726,205	17,373,859
# Samples	12,134,717	39,289,518
Avg. tokens per paragraph	166.93	134.61
Avg. # tokens in per $S \cup P$	8.53	7.52
Avg. # tokens in per O	3.19	2.99

Table 2: Statistics of the pre-training data.

3.2 Evaluation Tasks and Datasets

We first evaluate the proposed method with *cloze-style QA*. Then We adopt two other knowledge-intensive tasks, *closed-book QA* and *extractive QA*, to evaluate the PLMs' ability to capture and understand factual knowledge.

3.2.1 Cloze-style QA

Following Elazar et al. (2021); Cao et al. (2021); Petroni et al. (2019), we use cloze-style questions to probe the factual knowledge in PLMs. The PLMs need to recall the captured factual knowledge to fill in the masks. Cloze-style QA uses the same input-output format as MLM. So we do not need to fine-tune the PLMs and evaluate the factual knowledge capture performance directly.

Every cloze-style question is obtained by instantiating a artificial template on a fact. For example, the question "*Keanu Reeves is a citizen of* [MASK]" is constructed based on the template "[X] *is a citizen of* [Y]" and the triplet (*Keanu Reeves*, *citizen of*, *Canada*). Filling the mask with the correct token "*Canada*" is regarded as successfully capturing the corresponding fact.

We use the cloze-style questions and evaluation metrics from PARAREL (Elazar et al., 2021). PARAREL queries every fact with 8.64 different prompts on average. The prediction consistency is calculated by putting two different prompts into a prompt pair first (e.g., $n(n-1)/2$ pairs can be obtained from n different prompts). Then the percentage of the prompt pairs that can obtain the same result is used to indicate the consistency quantitatively. The overall factual knowledge capture performance is measured by jointly considering the prediction accuracy and consistency. Table 3 shows the statistics of the data from PARAREL.

We also employ four cloze-style datasets from

	Value
# Cloze-style questions	199,446
# Facts (Triplets)	23,097
Avg. # questions per fact	8.64

Table 3: Statistics of the cloze-style QA dataset in PARAREL (Elazar et al., 2021).

Dataset	# Facts
LAMA	29,522
LAMA w/o leakage	25,698
WIKI-UNI	69,761
WIKI-UNI w/o leakage	63,772

Table 4: Statistics of the four cloze-style QA datasets from (Cao et al., 2021).

(Cao et al., 2021). Table 4 shows the corresponding statistics. LAMA represents the cloze-style datasets that are similar to (Petroni et al., 2019) in (Cao et al., 2021). The distribution of the ground-truth answers in LAMA is imbalanced, providing a shortcut for PLMs to achieve good performance by selecting high-frequency entities as output. Therefore, Cao et al. (2021) proposes the WIKI-UNI dataset, where the distribution of the ground-truth answers follows a uniform distribution. The ground-truth answer has literal overlaps with the question sometimes. For example, in "*New York University is located in* [MASK] *city*," the right answer "*New York*" exactly leaked in the question. So Cao et al. (2021) filter out the questions that overlap with answers and obtain two more datasets based on LAMA and WIKI-UNI, which are indicated with the suffix "w/o leakage".

3.2.2 Closed-book QA

We also use the closed-book QA to test the ability of PLMs to capture factual knowledge. As proposed in (Wang et al., 2021; Roberts et al., 2020; Lewis et al., 2021), closed-book QA is similar to the way in which a student is taking a closed-book exam. The input of the model is the question only, and the model needs to generate the answer directly without seeing any other evidence. This task needs the model to generate arbitrary strings as answers, so we employ the BART, which has a decoder that can generate texts, as the base model for this task. We use the closed-book QA datasets from (Karpukhin et al., 2020), as shown in Table 5.

[4]The WebQuestions in (Karpukhin et al., 2020) does not have a development set, so we use the test set for both test and development.

Dataset	Train Set	Dev Set	Test Set
NaturalQuestions	79,168	8,757	3,610
TriviaQA	78,785	8,837	11,313
WebQuestions	3,778	-[4]	2,032

Table 5: Statistics of the closed-book QA datasets.

3.2.3 Extractive QA

The extractive QA is also known as machine reading comprehension (Liu et al., 2019a). The task is to search and extract the answer span from the input passage for the input question. It evaluates the ability of PLMs to understand the facts provided in passages. We employ the six extractive QA datasets from MRQA (Fisch et al., 2019), Table 6 presents the summary of the datasets. Following the setting in (Ram et al., 2021a), we use the development sets for testing.

Dataset	Train Set	Test Set
SQuAD	86,588	10,507
NewsQA	74,160	4,212
TriviaQA	61,688	7,785
SearchQA	117,384	16,980
HotpotQA	72,928	5,904
NatrualQuestions	104,071	12,836
Total	620,890	71,060

Table 6: Summary of the extractive QA datasets.

3.3 Results

3.3.1 Baselines

We continuously pre-train RoBERTa-*base* (Liu et al., 2019b) and BART-*base* (Lewis et al., 2020) from their official checkpoints with different masking methods:

- **Random token:** Mask random tokens in the tokenized text (Devlin et al., 2018).
- **Whole word:** Mask random words. All the tokens in the randomly selected words are masked at once (Cui et al., 2019).
- **Salient span:** Mask a span aligned with the subject or object, both deterministic and non-deterministic samples are included. (Guu et al., 2020).
- **Deterministic:** The proposed deterministic masking that masks a deterministic object, including only deterministic samples.

The above four models are trained with the mask-filling task only. The models pre-trained with clue contrastive learning and clue classification in company with deterministic masking are denoted as "**+ Con & Cls**." To further explore the potential of the proposed methods, we train the model on the full data with all the proposed methods, denoted as "**+ Full data**". We also introduce KEPLER-*base* as KB-enhanced baseline for comparison.

3.3.2 Cloze-style QA

Masking strategies Tables 7 and 8 present the results on cloze-style QA. We can see that random token masking can gain some improvements in performance, as well as the whole word masking. We think this is because the input texts, which come from Wikipedia, are formal descriptions of facts. Training on such texts helps shift the domain of PLMs for better generating factual words. The random token and whole word masking serve as solid baselines to focus the comparison between masking strategies, eliminating the confounders brought by extra pre-training on Wikipedia.

The salient span masking and deterministic masking both mask entity spans. The difference is that deterministic masking further limits the relationship between the remaining context and the masked span to be deterministic, driving PLMs to learn to infer based on the deterministic clues. The results show that the PLMs can achieve much better results with deterministic masking, indicating that the deterministic relationship is valuable for recovering factual spans robustly.

The proposed pre-training tasks The clue contrastive learning and the clue classification, which aim to strengthen the deterministic relationship, also provide further performance improvements (denoted as **+ Con & Cls**). Finally, the full data with all the proposed methods brings the most significant improvement. The proposed pre-training models also outperform the KEPLER-*base*.

Out-of-domain evaluation To analyze the improvement in-depth, we split the probing questions into *in-domain* and *out-of-domain* according to whether the pre-training corpus covers the corresponding triplets in questions. As Table 7 shows, the three random-based masking methods (Random token, Whole word, and Salient Span) boost performance on in-domain questions but get stuck on the out-of-domain questions. It is natural that the PLMs can answer the questions that are involved in pre-training. Surprisingly, although the out-of-domain questions are inaccessible in the pre-training corpus, the deterministic masking also gains performance improvement (3-4%), indicating

	Total			Out-of-domain			In-domain		
	Acc.	**Consis.**	**Joint**	**Acc.**	**Consis.**	**Joint**	**Acc.**	**Consis.**	**Joint**
RoBERTa-*base*	39.48	52.05	16.40	33.09	55.06	15.60	42.97	54.19	18.55
Random token	43.44	58.76	24.72	35.47	59.57	22.64	47.38	61.04	27.57
Whole word	44.04	58.96	25.88	36.67	60.48	23.01	47.91	61.57	29.40
Salient span	43.87	60.48	26.53	37.48	61.19	22.90	47.72	63.08	29.93
KEPLER-*base*	39.63	50.96	17.81	-	-	-	-	-	-
Deterministic	45.29	64.65	29.37	38.59	65.39	25.88	49.13	66.42	32.17
+ Con & Cls	46.01	64.53	29.62	38.05	65.35	26.24	50.26	65.71	32.32
+ Full data	**49.40**	**67.09**	**33.44**	**40.35**	**67.69**	**30.30**	**54.20**	**69.24**	**37.17**
BART-*base*	40.43	52.60	17.83	34.26	55.88	15.86	44.10	54.85	20.42
Random token	42.53	57.03	23.34	34.98	59.59	21.15	46.75	59.02	25.78
Whole word	41.68	58.96	24.38	34.53	61.24	21.32	45.11	60.94	26.42
Salient span	43.16	60.09	25.30	35.78	60.68	20.83	47.33	61.78	27.66
Deterministic	44.13	64.67	28.77	36.70	65.15	24.45	48.58	66.25	31.98
+ Con & Cls	46.65	65.64	28.89	39.51	66.31	25.53	50.72	67.86	32.45
+ Full data	**49.21**	**68.41**	**33.11**	**41.02**	**69.51**	**29.33**	**52.95**	**69.70**	**35.84**

Table 7: The factual knowledge capturing performance, evaluated by the cloze-style QA dataset PARAREL. **Acc.** is the accuracy, **Consis.** denotes the prediction consistency when changing the prompts, and **Joint** denotes the metric that jointly measures accuracy and consistency. **Out-of-domain** represent the set of questions whose triplets do not appear in the pre-training.

Dataset	**LAMA**	**w/o Leakage**	**WIKI-UNI**	**w/o Leakage**
RoBERTa-*base*	19.94	15.10	10.48	7.11
Random token	25.01	18.18	14.18	9.00
Whole word	25.66	18.94	14.42	9.27
Salient span	29.13	21.43	14.99	9.09
KEPLER-*base*	15.04	10.66	8.44	5.89
Deterministic	32.96	25.46	16.28	10.33
+ Con & Cls	32.16	24.88	16.24	11.03
+ Full data	**35.35**	**28.86**	**19.36**	**14.23**
BART-*base*	11.77	7.08	6.03	3.47
Random token	25.39	17.75	14.01	8.24
Whole word	25.06	17.21	14.21	8.75
Salient span	30.07	22.20	15.56	9.60
Deterministic	31.26	23.69	15.81	10.22
+ Con & Cls	32.03	24.57	15.58	10.23
+ Full data	**35.49**	**28.98**	**18.64**	**13.21**

Table 8: The results on cloze-style QA datasets from (Cao et al., 2021). The performance is measured by accuracy[5].

that the deterministic relationship could help PLMs to better recollect the facts learned implicitly.

3.3.3 Closed-book QA

We fine-tune the continuously pre-trained BART-*base* on the Closed-book QA task. The metrics are EM(Exact Match) and F1 from (Rajpurkar et al., 2016). Table 9 shows the comparison results of different strategies.

[5]The detailed metrics grouped by the relation types (N-1,N-M relations) are in Appendix A.

Closed-book QA is more difficult than cloze-style QA since the models need to generate answers without any extra hints, e.g., the answer length is indicated by the number of [MASK]s in the cloze-style QA, while the models need to predict the answer length in closed-book QA. The input-output format of closed-book QA differs from per-training, so we need to fine-tune PLMs to recall facts based on natural questions to fit this format. Table 9 shows the evaluation results. Generally, the proposed methods outperform the baselines, demonstrating that the proposed methods can help the PLM that uses encoder-decoder architecture to capture and recall factual knowledge.

3.3.4 Extractive QA

We fine-tune the models that based on RoBERTa-*base* for extractive QA. Following (He et al., 2020; Joshi et al., 2020), we employ the MRQA data with two different settings: (a) **Separate:** the models are trained and tested on every QA dataset separately, (b) **Combine:** all the training samples from the six datasets are merged in training. Then the fine-tuned models are evaluated on each dataset respectively. Table 10 shows the evaluation results, the metrics are averaged over the six development sets.

In extractive QA, the input includes a question and the supporting evidence to answer it. So the models do not have to recollect the essential evidence but should put more effort into understanding the evidence. Table 10 shows the evaluation results.

Dataset	Model	F1	EM
TriviaQA	BART-*base*	23.91	17.52
	Random token	23.67	18.04
	Whole word	24.64	18.88
	Salient span	24.98	19.21
	Deterministic	24.94	19.22
	+ Con & Cls	25.28	19.58
	+ Full Data	**26.35**	**20.57**
NaturalQuestions	BART-*base*	26.89	21.27
	Random token	27.34	21.55
	Whole word	27.53	22.13
	Salient span	27.31	22.07
	Deterministic	27.83	22.60
	+ Con & Cls	28.14	22.69
	+ Full Data	**29.17**	**23.91**
WebQuestions	BART-*base*	33.62	26.62
	Random token	32.58	26.38
	Whole word	32.45	26.03
	Salient span	32.70	26.08
	Deterministic	32.73	26.38
	+ Con & Cls	32.63	25.59
	+ Full Data	**33.91**	**27.26**

Table 9: The performance on the closed-book QA datasets.

Model	Separate		Combine	
	F1	EM	F1	EM
RoBERTa-*base*	80.78	69.51	81.78	70.57
Random token	80.53	69.22	81.79	70.52
Whole word	80.86	69.71	81.77	70.60
Salient span	80.85	69.61	81.72	70.50
KEPLER-*base*	80.28	69.02	81.41	70.32
Deterministic	80.83	69.63	81.78	70.59
+ Con & Cls	80.94	69.75	81.79	70.67
+ Full data	**80.96**	**69.67**	**81.86**	**70.71**

Table 10: The performance on the extractive QA task.

Due to the difference in the input-output format between the MLM and span extraction task, the change in the masking methods has somewhat limited effects on the performance here. The averaging on six different MRC datasets and the hyperparameter search (in Appendix C) could further diminish the performance difference between the models. However, the proposed methods still show slight advantages in the comparison, demonstrating that learning the deterministic relationship could also help to comprehend factual knowledge.

4 Related Work

Pre-training on large-scale unlabeled text can help PLMs capture meaningful knowledge and benefits the downstream tasks accordingly (Brown et al., 2020; Radford et al., 2018). BERT (Devlin et al., 2018) proposes a Mask Language Model (MLM) in which the model needs to recover some masked tokens based on the remaining context. The effectiveness of the MLM makes BERT become the starting point for fitting many downstream tasks (Chen et al., 2020). Afterward, several different masking methods (Cui et al., 2019; Joshi et al., 2020; Levine et al., 2020; Sun et al., 2019) have explored how masking methods affect performance and have obtained further performance improvement. These works push the limit of MLM and show the importance of designing better masking strategies.

On the other hand, some pre-training tasks other than MLM have been proposed. Clark et al. (2019) trains the model to distinguish the replaced words from the original words in the context. (Xiong et al., 2019; He et al., 2020) let factual spans be the replacement candidates. Qin et al. (2020) contrasts the representations between different entities and relations. This paper views another perspective of the masking methods: whether the remaining context can uniquely determine the masked span. Accordingly, we propose a deterministic masking strategy that masks deterministic spans in MLM samples. Moreover, we design clue contrastive learning and clue classification as pre-training tasks to help PLMs identify the deterministic clues for the masked span and contrast them with the non-deterministic ones. Moreover, we evaluate the performance of the proposed model with various downstream tasks.

5 Conclusion

This paper proposes to train PLMs to learn a deterministic input-output relationship in MLM to improve PLMs on capturing factual knowledge. The deterministic relationship ensures the masked content in MLM samples is deterministically predictable based on the remaining context. To further enhance the deterministic relationship, we design a pre-training task clue contrastive learning that encourages PLMs to give more confident predictions when the input keeps deterministic clues, and the clue classification to train PLMs to predict whether the deterministic clues exist. Extensive experiments show that the proposed methods can improve the accuracy and consistency of factual knowledge capturing and boost the performance of the other two knowledge-intensive tasks.

6 Limitations

We summarize this paper's main limitations as follows: First, this study focuses on enhancing the deterministic relationship but does not explore the non-deterministic relationships. The other non-deterministic relationships also play essential roles in tasks such as semantic role labeling and emotion recognition, where the proposed methods may not be helpful. Second, due to the diversity and richness of natural language, we cannot perfectly recognize the deterministic clues and spans from texts. We have to consider the noises in recognization when designing the pre-training tasks. Finally, we continuously pre-train PLMs on only Wikipedia text, somewhat narrowing down their domain. Constructing more pre-training samples by the proposed procedure (Procedure 1 in the Appendix) could be better. Moreover, we can use the current pre-training samples (based on Wikipedia) to train an "interpolation model" that can tag the deterministic clues and spans in the input texts. The interpolation model can also be used to enlarge the pre-training data.

Acknowledgements

We would like to thank the anonymous reviewers for providing valuable reviews throughout the multi-turn rolling review progress. Thanks to Benyou Wang for the helpful discussions, suggestions, and encouragement. The research in this article is supported by the National Key Research and Development Project (2021YFF0901600) and the Interdisciplinary Development Program of Harbin Institute of Technology (No. SYL-JC-202203).

References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. Knowledgeable or educated guess? revisiting language models as knowledge bases. 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 1860–1874, Online. Association for Computational Linguistics.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020. The lottery ticket hypothesis for pre-trained bert networks. *arXiv preprint arXiv:2007.12223*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Elijah Cole, Oisin Mac Aodha, Titouan Lorieul, Pietro Perona, Dan Morris, and Nebojsa Jojic. 2021. Multi-label learning from single positive labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 933–942.

Thomas M Cover. 1999. *Elements of information theory*. John Wiley & Sons.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Ziqing Yang, Shijin Wang, and Guoping Hu. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Thibaut Durand, Nazanin Mehrasa, and Greg Mori. 2019. Learning a deep convnet for multi-label classification with partial labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 647–657.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models.

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)*.

Adam Fisch, Alon Talmor, Robin Jia, Minjoon Seo, Eunsol Choi, and Danqi Chen. 2019. MRQA 2019 shared task: Evaluating generalization in reading comprehension. In *Proceedings of 2nd Machine Reading for Reading Comprehension (MRQA) Workshop at EMNLP*.

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*.

Bin He, Xin Jiang, Jinghui Xiao, and Qun Liu. 2020. Kgplm: Knowledge-guided language model pre-training via generative and discriminative learning. *arXiv preprint arXiv:2012.03551*.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know. *Trans. Assoc. Comput. Linguistics*, 8:423–438.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

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.

Yoav Levine, Barak Lenz, Opher Lieber, Omri Abend, Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. 2020. Pmi-masking: Principled masking of correlated spans. In *International Conference on Learning Representations*.

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.

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.

Shanshan Liu, Xin Zhang, Sheng Zhang, Hui Wang, and Weiming Zhang. 2019a. Neural machine reading comprehension: Methods and trends. *Applied Sciences*, 9(18):3698.

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

Xin Luo, Wei Chen, Yusong Tan, Chen Li, Yulin He, and Xiaogang Jia. 2021. Exploiting negative learning for implicit pseudo label rectification in source-free domain adaptive semantic segmentation. *arXiv preprint arXiv:2106.12123*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473.

Nina Pörner, Ulli Waltinger, and Hinrich Schütze. 2020. E-BERT: efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020.

Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2020. Erica: improving entity and relation understanding for pre-trained language models via contrastive learning. *arXiv preprint arXiv:2012.15022*.

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.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021a. Few-shot question answering by pretraining span selection. *arXiv preprint arXiv:2101.00438*.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021b. Few-shot question answering by pretraining span selection. 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 3066–3079, Online. Association for Computational Linguistics.

Raymond Reiter. 1981. On closed world data bases. In *Readings in artificial intelligence*, pages 119–140. Elsevier.

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.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Automatic prompt construction for masked language models. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Johannes M. van Hulst, Faegheh Hasibi, Koen Dercksen, Krisztian Balog, and Arjen P. de Vries. 2020. Rel: An entity linker standing on the shoulders of giants. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. 2019. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2517–2526.

Cunxiang Wang, Pai Liu, and Yue Zhang. 2021. Can generative pre-trained language models serve as knowledge bases for closed-book qa? *arXiv preprint arXiv:2106.01561*.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *arXiv preprint arXiv:1912.09637*.

Min-Ling Zhang and Zhi-Hua Zhou. 2006. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering*, 18(10):1338–1351.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: learning vs. learning to recall. *CoRR*, abs/2104.05240.

		Total			Out-of-domain			In-domain		
		Acc.	Consis.	Joint	Acc.	Consis.	Joint	Acc.	Consis.	Joint
RoBERTa-*base*	Salient span	43.87	60.48	26.53	37.48	61.19	22.90	47.72	63.08	29.93
	Object	44.41	63.68	28.00	38.48	64.40	24.94	47.82	65.25	31.27
	Deterministic	45.29	64.65	29.37	38.59	65.39	25.88	49.13	66.42	32.17
	+ Con	45.93	65.50	29.52	39.33	66.34	26.26	49.50	67.26	32.58
	+ Con & Cls	46.01	64.53	29.62	38.05	65.35	26.24	50.26	65.71	32.32
BART-*base*	Salient span	43.16	60.09	25.30	35.78	60.68	20.83	47.33	61.78	27.66
	Object	42.61	62.67	27.24	35.76	62.22	22.39	46.91	64.89	30.73
	Deterministic	44.13	64.67	28.77	36.70	65.15	24.45	48.58	66.25	31.98
	+ Con	44.96	64.11	28.17	37.66	64.50	24.51	48.97	66.16	31.71
	+ Con & Cls	46.65	65.64	28.89	39.51	66.31	25.53	50.72	67.86	32.45

Table 11: The evaluation results on PARAREL. "Object" denotes the baseline that masks and predicts the objects. "Deterministic" denotes the MLM baseline that uses deterministic masking. "+ Con" is the baseline that uses the clue contrastive learning with deterministic masking.

Appendix

A Ablation Study

How does masking objects help in factual knowledge capturing? As described in Section 3.2.1, the cloze-style questions that we use to probe the factual knowledge in PLMs, are constructed by integrating subject-predicate-object triples with artificial templates. Due to the conventions in the template construction, the objects could have more opportunities to be the answer than the predicates and subjects. So focusing on recovering objects in pre-training may also benefit cloze-style QA. The proposed deterministic masking naturally masks more objects in pre-training because of the rules we designed to identify the deterministic span. Both masking objects and the deterministic relationship could bring improvements in the deterministic masking.

To investigate and clarify their contributions, we introduce a baseline "Object" that masks and predicts only object spans in pre-training. Table 11 shows the evaluation result. We can see that the Object baseline performs better than the Salient span baseline on factual knowledge capture. It reveals that masking objects indeed improve performance. Nevertheless, the deterministic masking (denoted as "Deterministic") achieves better results, denoting that both masking objects and learning the deterministic relationship contribute positively to factual knowledge capture.

The effectiveness of the clue contrastive learning and clue classification To reveal the contribution of the proposed pre-training tasks separately, we introduce a baseline that only uses the clue contrastive learning. We refer to it as "+ Con" in Table 11. "+ Con & Cls" denotes the PLM that uses the clue classification in addition to the clue contrastive learning. We can see that the performance increases as we apply the proposed methods incrementally.

The improvements on deterministic and non-deterministic cloze-style questions The dataset from PARAREL includes only the N-1 relations[6]. While the LAMA dataset from (Petroni et al., 2019) (Tables 4 and 8) includes both the N-1/1-1 and N-M relations. To reveal the improvements in terms of relation types, we separate the samples into N-1/1-1 and N-M based on the relation types and report the results separately, as shown in Table 12. Similar to the deterministic relationship we used, the ground-truth object is unique when the relation type is N-1 or 1-1. The results show that the improvement for the N-1/1-1 relations is more significant than the N-M relations when using the proposed methods.

B Pre-training Data Construction

Procedure 1 shows how we construct the pre-training data, including entity linking, predicate matching, triplet aligning, and deterministic relationship checking. Each text piece t is a paragraph in Wikipedia. We use the entity linker provided in (van Hulst et al., 2020), represented as ENTITYLINKER, to identify all the entities in the paragraph[7]. The WikiData defines 12,043 aliases for 8,529 predicates. Function PREDICATEMATCHER searches the substring corresponding to a predicate by comparing the predicate's aliases with all the substrings in the text. The identified predicate span is the nearest match whose edit distance is less than

[6]Defined in https://github.com/yanaiela/pararel/wiki/31-N1-Relations

Procedure 1 Deterministic Sample Construction

Require: Text collection T, Knowledge base K, EntityLinker
Output: Deterministic sample collection D_{d}
Output: Salient span masking sample collection D_{ssm}

1: $D_{\mathrm{d}} \leftarrow \{\}$, $D_{\mathrm{ssm}} \leftarrow \{\}$
2: **for all** text piece t **in** T **do**
3: $E \leftarrow \textsc{EntityLinker}(t)$ ▷ Identify all the entities in t
4: $D_{\mathrm{ssm}} = D_{\mathrm{ssm}} \cup \{(t, E)\}$ ▷ Save the entities for the salient span masking
5: **for all** entity pair (e_i, e_j) **in** $E \times E$ **do**
6: **for all** predicate r that can connects (e_i, e_j) **do** ▷ Triplet (e_i, r, e_j) exists in K
7: **if** e_j has only one match when querying K with e_i and r as the subject and predicate **then**
8: $p \leftarrow \textsc{PredicateMatcher}(t, r)$ ▷ Find the spans in t that correspond to r
9: $s \leftarrow e_i$ ▷ Use e_i as subject s
10: $o \leftarrow e_j$ ▷ Use e_j as object o
11: $d \leftarrow (s, p, o, E, t)$ ▷ Group the alignments into d
12: $D = D \cup \{d\}$ ▷ Record the sample d
return D_{d}, D_{ssm}

13: **function** PredicateMatcher(t, r)
14: **for all** alias string a for r in K **do** ▷ WikiData holds a alias string collection for every predicate
15: **for all** substring s in t **do**
16: **if** edit distance between a and s < 2 **then**
17: **Return** s

Model	Total	N-1/1-1	N-M
RoBERTa-*base*	19.94	22.18	16.46
Random token	25.01	28.98	18.81
Whole word	25.66	29.26	20.05
Salient span	29.13	30.89	26.38
Deterministic	32.96	37.84	25.35
+ Con & Cls	32.16	36.62	25.19
+ Full data	**35.35**	**41.86**	**25.18**
BART-*base*	11.77	13.96	8.35
Random token	25.39	27.66	21.84
Whole word	25.06	28.69	19.39
Salient span	30.07	31.83	27.31
Deterministic	31.26	35.77	24.21
+ Con & Cls	32.03	36.38	25.25
+ Full data	**35.49**	**41.99**	**25.33**

Table 12: The detailed results on the LAMA dataset in (Cao et al., 2021), reported separately with respect to relation types: N-1/1-1 or N-M.

two.

After recognizing the entities and predicates, we combine every entity pair with every predicate as a triplet (entity, predicate, entity), enumerate all the possible combinations, and keep the ones that existed in KB as the triplets aligned with the paragraph. Line 7 checks if the object is deterministic by querying KB. We record the obtained deterministic sample in the format of (subject s, predicate p, object o, text t, and entities E).

The baselines use the pre-training sample as the following:

- **Deterministic** (mask deterministic object to train MLM): Get a sample (s, p, o, E, t) from D_{d}, mask the span corresponding to o and train PLMs to predict o based on the remaining context.
- **Random** (mask tokens randomly): Get a sample from D_{d}, tokenize t and calculate the number of tokens in o, denoted as $\textsc{TokenCount}(o)$, randomly sample $\textsc{TokenCount}(o)$ tokens to be masked in the MLM training.
- **Whole word** (mask whole words randomly): Get a sample from D_{d}, calculate the num-

[7] https://github.com/informagi/REL

ber of words in o (separated by space), denoted as WORDCOUNT(o), randomly mask WORDCOUNT(o) words in t.

- **Salient Span** (mask entities randomly): Get a sample (s, p, o, E, t) from D_{ssm}, randomly mask an aligned entity in E.

Although the masking granularity is different in the baslines, we keep the length of the masked content as similar as possible for a fair comparison.

Then we introduce how the two proposed pre-training tasks use the data. In the clue contrastive learning, the s and p are the deterministic clues and masked in the contrastive sample. If the same o have more than one deterministic clue in t, e.g., multiple deterministic clues for the same o are given by different triplets, all the deterministic s and p are considered as determined clues and masked in the contrastive sample (b). In the clue classification, the number of the randomly masked token in the sample (c) is the same as the constrastive sample.

Procedure 1 is used to generate the **full data** (summarized in Table 2). We obtain the **partial data** similarly, except that we do not need the ENTITYLINKER (at Line 3 in Procedure 1) and directly use the entity-text alignments provided in TREX.

C Hyperparameters

Pre-training For the baselines trained on the partial data, the batch size is set to 512, the learning rate is 3×10^{-5}, and the number of total training steps is 50,000. There are 200,000 training steps for the final model on the full data.

Extractive QA In the experiments on extractive QA, we find that the model's performance is sensitive to hyperparameters. We conduct a grid search over the learning rate and batch size. In the separate setting, the learning rate is searched over $\{1 \times 10^{-5}, 2 \times 10^{-5}, 3 \times 10^{-5}\}$, the batch size is searched over $\{16, 32, 64\}$ when fine-tuning every PLM on every dataset, and the epoch is set to 4. In the combine setting, the learning rate $\in \{2 \times 10^{-5}, 3 \times 10^{-5}\}$, the batch size $\in \{32, 64\}$, and the epoch is set to 2. We save the model checkpoint per 5,000 steps. The best model is selected from evaluating all the checkpoints. We use the code released by (Ram et al., 2021b)[8].

Closed-book QA The learning rate is set to 5×10^{-5}. The training steps is set to 100,000 for NaturalQuestions and TriviaQA, 40,000 for WebQuestions. The model is evaluated per 10,000 training steps to select the best checkpoint.

[8] https://github.com/oriram/splinter