Improving Machine Reading Comprehension through A Simple Masked-Training Scheme

Xun Yao[†], **Junlong Ma**[†], **Xinrong Hu**[†], **Jie Yang**[◇*], **Yuan-Fang Li**[‡]
[†]School of Computer Science and Artificial Intelligence, Wuhan Textile University
[‡]School of Computing and Information Technology, University of Wollongong
[◇]Faculty of Information Technology, Monash University
{yaoxun, hxr}@wtu.edu.cn
jiey@uow.edu.au,
YuanFang.Li@monash.edu

Abstract

Extractive Question Answering (EQA) is a fundamental problem in Natural Language Understanding, aiming at answering given questions via extracting a contiguous sequence or span of words from a passage. Recent work on EQA has achieved promising performance with the help of pre-trained language models, for which Masked Language Modeling (MLM) is usually adopted as a pre-training task to predict masked tokens. This paper revisits MLM and proposes a simple yet effective method to improve the EQA performance, termed the `[Mask]`-for-Answering method (M4A). Specifically, three masking strategies are first introduced, which produce masked copies of the original passages. Instead of predicting masked tokens as in MLM, both original samples and masked copies are utilized simultaneously for training the EQA model. Importantly, a discrepancy loss is further incorporated to ensure that masked copies remain semantically close to the originals. As such, M4A is able to produce robust embeddings for both original and masked samples and infer correct answers even with masked context. Experimental study on several highly-competitive benchmarks consistently demonstrates the superiority of our proposed method over existing methods. M4A also achieves strong performance in low-resource settings and out-of-domain generalization.

1 Introduction

Extractive Question Answering (EQA), a fundamental task for Natural Language Understanding, refers to the process of identifying the answer span (a sequence of continuous words) over the given question and passage. Past years have witnessed a dramatically increasing interest in applications of Pre-trained Language Models (PLMs) for EQA, where PLMs are usually adopted as encoders to form contextual/semantic embeddings for the question-passage pair. Abundant evidences indicate that the strong encoding capability of PLMs has rapidly advanced the EQA performance, compared to traditional word embedding methods, such as GloVe and Word2Vec (Devlin et al., 2019; Joshi et al., 2020; Liu et al., 2019). Recently, how humans approaching reading comprehension becomes a major source of inspiration for enhancing EQA. There is a rich literature to incorporate the human-like reading comprehension strategy with PLMs and achieve remarkable success beyond the vanilla models, as later shown in Section 2.

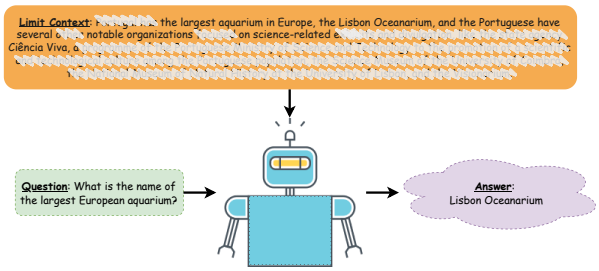

Figure 1: An illustration case from SQuAD (1.1) (Fisch et al., 2019), using the proposed `[Mask]`-for-Answering algorithm (M4A) to infer answer(s) from samples with limit (unmasked) context.

Notably, when experienced human readers perform question reasoning, they could infer the correct answer using only few sentences (even some parts of one sentence), instead of the entire passage (Paris et al., 1983; Yu et al., 2017). As illustrated in Fig. 1, even with the limit (unmasked) contexts, one could still predict the correct answer of "`Lisbon Oceanarium`" to the given question (of "`the name of the largest European aquarium`"). The observation of humans approaching reading comprehension with limit (unmasked) contexts is the major source of inspiration for this paper.

Accordingly, we propose a simple yet effective mask-training scheme, termed `[Mask]`-for-Answering (M4A). Specifically, M4A introduces

*corresponding author

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023, pages 222–232

three masking strategies to substitute non-answer tokens from the original training passage with [Mask] tokens. Additionally, semantic similarity is utilized to maintain the semantic closeness between masked samples and originals. The effectiveness of our training scheme can be intuitively explained from two perspectives: (1) providing strong training signals by strategically masking out potentially-irrelevant contents (non-answer tokens) and (2) data augmentation by simply perturbing the original training samples without additional annotations.

Our method differs from existing methods in the following perspectives. (1) The [Mask] token is traditionally utilized to hold out a portion of the input tokens in pre-training PLMs to predict missing tokens (Özkan Tan et al., 2023; Yang et al., 2023). Several studies leverage [Mask] to produce pseudo passage-question pairs (Ram et al., 2021; Bian et al., 2021), which is limited by parts of speech (POS) of masked tokens (usually nouns) and the objective is still for predicting masked tokens (the in-domain pre-training). In contrast, M4A directly employs samples with masked tokens in directly optimizing the downstream task objective. Additionally, in M4A non-answer tokens can be masked regardless their POS. (2) Masked samples also play a role of data augmentation, and existing augmentation methods either replace words with synonyms (Ng et al., 2020) or perturb input embeddings (Lee et al., 2021). The former is limited by the set of available synonyms, while our method is independent from synonyms. The latter adds noise on the embedding level (for all tokens) with a prior assumption of a multivariate Gaussian distribution. In contrast, M4A performs masking on the token level, and ensures (ground-truth) answer tokens unmasked.

The main contributions of our proposed work are summarized as follows:

- A novel [Mask]-for-Answering method is proposed to produce robust features and incorporate human comprehension skills to infer answers from samples with masked tokens.
- Three masking strategies are introduced to produce masked samples, that are trained simultaneously with original inputs.
- The semantic similarity between original-and-masked pairs is applied to minimize the noise conveyed in masked samples.
- Empirically, our proposed M4A method outperforms recent strong baselines on six standard benchmarks. Intensive ablation studies are also conducted to understand the impact from masking strategies and ratios. Moreover, M4A also demonstrates a strong generalizability in the low-resource training and zero-shot domain adaptation setting.

2 Related work

The Extractive Question Answering (EQA) task requires a model to learn informative representation from the context passage, and return a span (continuous words from this passage) that matches the given question. Usually, Pre-trained Language Models (PLMs) are adopted as the encoder to estimate embeddings for the pair of question-passage, which is followed by a decision layer (*i.e.*, two binary classifiers to identify the start and end position of the answer span respectively). Due to the capability of forming semantic representation for input questions and passages, PLMs have significantly advanced the EQA frontier (Devlin et al., 2019; Joshi et al., 2020; Liu et al., 2022).

Inspired by the remarkable success of PLMs, a variety of improvement methods have been proposed to further enhance PLMs with human-reading strategies. A block based attention method is proposed in (Seonwoo et al., 2020), which predicts highly-relevant context about answers. Another similar work is found in (Guan, 2022), that introduces the Block-Skim strategy to identify and skim irrelevant context blocks by utilizing CNN to for EQA. In Sun et al. (2019), different-level attention mechanisms are implemented to simulate the process of back-and-forth reading, highlighting, and self-assessment, while Zhang et al. (2020) considers the reading strategy for multi-round reasoning phrases of reading-attending-excluding, that is, through initial scan *reading*, followed by *attended* intensive reading, and concluding with answer *exclusion*.

In addition to attention-based work, a few studies employed the data-augmentation strategy to fine-tune the EQA task, including synthetic question-answer generation, external knowledge, and input perturbation. As an example, the work from (Ram et al., 2021; Bian et al., 2021) produces synthetic pairs by masking specific words from the passage and training a model to answer questions related to those masked words. Yet, generation-based methods usually suffer from costly computational resource and have limitations on masked words (usually nouns) and question

types (mainly cloze-like). KALA (Kang et al., 2022) is proposed to integrate the contextual representation of intermediate PLM layers with related entity and relational representations (from the external Knowledge Graph). With knowledge-augmented representations, KALA improves the performance of the vanilla PLM on various EQA tasks. Additionally, input perturbation is also considered via the word deletion or synonym replacement (Wei and Zou, 2019). SSMBA (Ng et al., 2020) corrupts input sequences via substituting existing tokens with `[Mask]` and then reconstructing them. SWEP (Lee et al., 2021) directly perturbs input embeddings via adjustable Gaussian noises. Using both original and perturbed samples, models are trained to extract token representations to facilitate the subsequent answering task. By comparison, our proposed method neither recovers masked tokens nor perturbs input embeddings. Instead, augmented samples (with `[Mask]` tokens) are directly utilized for the model training while ensuring their semantic similarity with originals.

3 Proposed method

The proposed `[Mask]`-for-Answering (M4A) algorithm is detailed in this section. It consists of three main modules: source training, `[Mask]` training, and semantic alignment. Specifically, the second module enables the model to answer questions with incomplete (masked) contents, while the third module ensures semantic correlations between original and masked samples (shown in Fig 2). Notably, the `[Mask]` training module is not utilized during the inference stage, as our ultimate aim is to fine-tune a resilient encoder that can handle masked samples and generate robust features.

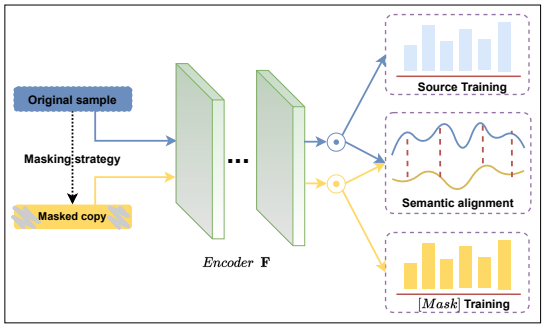

Figure 2: Illustration of the proposed M4A for EQA. Original inputs are masked to produce additional training samples, while their representation similarity is later maximized to remain the semantic alignment.

3.1 Source training

Given the input pair of question ($\boldsymbol{q}$) and passage ($\boldsymbol{p}$), the EQA task aims to identify the start and end positions of the correct answer span ($\boldsymbol{a}_{s/e}$) from $\boldsymbol{p}$. Specifically, the input of EQA is a tokenized sequence, $X = \texttt{[CLS]}\,\boldsymbol{q}_1\boldsymbol{q}_2\cdots\ \boldsymbol{q}_{|\boldsymbol{q}|}\,\texttt{[SEP]}\,\boldsymbol{p}_1\boldsymbol{p}_2\cdots\ \boldsymbol{p}_{|\boldsymbol{p}|}\,\texttt{[SEP]}$, where $\boldsymbol{q}_i$ and $\boldsymbol{p}_j$ represents the i-th and j-th token from $\boldsymbol{q}$ and $\boldsymbol{p}$, respectively. Then the encoder ($\mathbf{F}$, usually a PLM such as BERT (Devlin et al., 2019)) is applied to induce the following probability distribution:

$$\mathrm{p}(\boldsymbol{p}_i = \boldsymbol{a}_{s/e}) \triangleq \frac{\exp(\mathrm{MLP}^{s/e}(\mathbf{F}(\boldsymbol{p}_i)))}{\sum_j^{|\boldsymbol{p}|}\exp(\mathrm{MLP}^{s/e}(\mathbf{F}(\boldsymbol{p}_j)))}, \quad (1)$$

where $\mathbf{F}(\boldsymbol{p}_i)$ represents the extracted feature of $\boldsymbol{p}_i$, and $\mathrm{MLP}^{s/e}$ represents a multilayer perceptron network (usually with one-hidden layer) for predicting the *s*tart and *e*nd of the answer span, respectively. Accordingly, the loss function for EQA is defined as follows:

$$\mathcal{L}_{EQA} \triangleq -\sum_i^{|\boldsymbol{p}|} \mathbb{1}(\boldsymbol{p}_i = \boldsymbol{a}_{s/e}) \log \mathrm{p}(\boldsymbol{p}_i = \boldsymbol{a}_{s/e}), \quad (2)$$

where $\mathbb{1}(\cdot)$ is the indicator function that returns 1 if the condition is true and returns 0 otherwise. Overall, the source training module is to minimize the $\mathcal{L}_{EQA}$ loss using labeled original samples.

3.2 `[Mask]` training

The traditional mask-based training (or masked-language modeling (MLM)) aims to predict a set fraction of `[Mask]` tokens given the remaining unmasked text. This fraction is defined as the masking budget (b_M), and tokens for masking are chosen by a uniform (or random) sampling until b_M is met. This previous mask-and-predict task is different from ours. In contrast, samples with masked tokens are utilized for training the EQA model directly, while the mask-prediction task is less emphasized.

Given the tokenized input X, this module substitutes tokens from the passage $\boldsymbol{p}$ (the second part of X) with `[Mask]` to generate the masked copy X^M.* Specifically, three masking strategies, *Gaussian*, *U-shaped*, and *Uniform*, are adopted,

*Some masking techniques (such as removing continues words (Joshi et al., 2020) or words with dependency connections (Tian et al., 2022)) may be beneficial to produce more diverse masked copies, and we leave the investigation to future work.

which are differentiated by the distributions of masked tokens. A reference index r is introduced ($r \in [1, |\boldsymbol{p}|]$), where r is explicitly determined by the index of the *start* token from the ground-truth answer span. Then, *Gaussian* favors masking tokens around r, while *U-shaped* differs from *Gaussian* by masking more away from r (towards the begin or end of $\boldsymbol{p}$). *Uniform* masks tokens with a uniform probability (similar to the MLM). Note that for all three masking strategies, tokens belonging to the ground-truth answer span are assigned with zero masking probability (never masked), and only non-answer tokens can be masked.

To estimate the masking rate for each token within the passage, probability density functions (PDFs) under different masking strategies are firstly introduced. Assume that PDFs are defined within a range of $[-\Delta, \Delta]$ (where $\Delta > 0$ is a hyper-parameter). Followed by the standard Gaussian distribution $\mathcal{N}(0, 1)$, the PDF for the *Gaussian* masking is accordingly defined as

$$\mathrm{Gau}(x) = \frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}, \quad -\Delta \le x \le \Delta.$$

Similarly the PDF for the *U-shaped* distribution is defined as:

$$\mathrm{Ush}(x) = \begin{cases} \mathrm{Gau}(x-\Delta) & \text{if} \quad 0 \le x \le \Delta \\ \mathrm{Gau}(x+\Delta) & \text{if} \quad -\Delta \le x < 0 \end{cases}.$$

To be consistent with *Gaussian* and *U-shaped*, the PDF for *Uniform* is simply given by

$$\mathrm{Uni}(x) = \frac{1}{\Delta - (-\Delta)} = \frac{1}{2 \times \Delta}, \quad -\Delta \le x \le \Delta.$$

Next, given the reference index r, the following mapping function is adopted to project the t-th token into this range of $[-\Delta, \Delta]$ by:

$$\mathrm{Map}(t) = \frac{\Delta(t-r)}{\max(r, |X|-r)}. \tag{3}$$

At last, given the masking budget (the fraction of masked tokens) b_M, the masking probability for this t-th token is further given by:

$$\mathrm{p}(t) = b_m \times |X| \times \frac{f(\mathrm{Map}(t))}{\sum_i^{|X|} f(\mathrm{Map}(i))}, \tag{4}$$

where $f(\cdot)$ represents $\mathrm{Gau}(\cdot)$, $\mathrm{Ush}(\cdot)$, and $\mathrm{Uni}(\cdot)$ for the case of *Gaussian*, *U-shaped*, and *Uniform*, respectively. An illustrating example of the three masking strategies is provided in Fig. 3. At last, masked passages are combined with the given question to form X^M, and the resultant loss $\mathcal{L}_{EQA}^{\texttt{[Mask]}}$ is obtained by replacing X with X^M in Eq. (2).

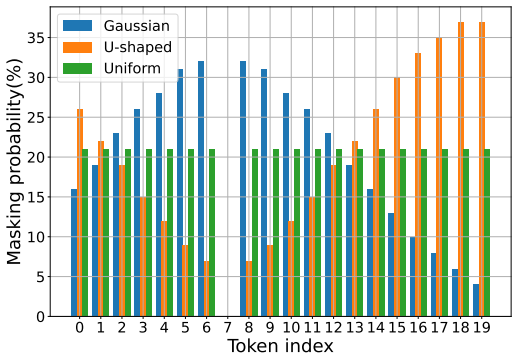

Figure 3: Comparison of proposed masking strategies, where b_m = 20%, the length of the passage is 20 and Δ=2. Presumably, the ground-truth answer token is with the 7^{th} index. Specifically, *Gaussian* prefers to mask tokens around the reference index of r=7, while more tokens from the begin and end are masked in the *U-shaped* strategy. By contrast, *Uniform* marks tokens with the equally distributed way. Notably, ground-truth answer tokens will never be masked across three cases.

3.3 Semantic alignment

The `[Mask]` training module enables the model to answer questions with masked samples X^M, which could introduce noise from masked tokens. To reduce the impact of noises, we design the semantic alignment module to suppress noisy signals conveyed in X^M, and enforce the semantics of original inputs is preserved even after masking parts of the passage. This is done by minimizing the difference between the distribution of individual-token score (*i.e.*, the probability of being correct-answer tokens) obtained from X and that of the corresponding X^M.

To begin with, let $\mathbf{C}$ and $\mathbf{C}^M$ represent feature embedding of original and masked samples, *i.e.*, $\mathbf{C} = \mathbf{F}(X)$ and $\mathbf{C}^M = \mathbf{F}(X^M)$, where $\mathbf{C}/\mathbf{C}^M \in \mathbb{R}^{|X| \times l}$ and l is the hidden dimension. Given two classifiers for identifying the start/end token ($\mathrm{MLP}^{s/e}$) from Eq. (1), the score distribution of individual tokens from $\mathbf{C}$ and $\mathbf{C}^M$ is estimated by:

$$\begin{aligned} d_{\mathbf{C}}^{s/e} &\triangleq \mathrm{softmax}(\mathrm{MLP}^{s/e}(\mathbf{C})) \\ d_{\mathbf{C}^M}^{s/e} &\triangleq \mathrm{softmax}(\mathrm{MLP}^{s/e}(\mathbf{C}^M)), \end{aligned} \tag{5}$$

The Jensen-Shannon divergence ($\mathbb{D}_{\mathrm{JS}}$) is then employed to measure the distribution similarity with the following objective:

$$\mathcal{L}_{ALI} \triangleq \mathbb{D}_{\mathrm{JS}}(d_{\mathbf{C}}^{s}, d_{\mathbf{C}^M}^{s}) + \mathbb{D}_{\mathrm{JS}}(d_{\mathbf{C}}^{e}, d_{\mathbf{C}^M}^{e}). \tag{6}$$

As such, this loss minimization encourages $\mathbf{F}$ to produce semantically similar representations between the original input X and its masked version

X^M. Note that this alignment loss is different from that of [Mask] training, as it does not require supervision signals, but only reduces the token-score difference obtained by X and X^M.

3.4 Overall objective function

In summary, in the proposed M4A, samples with masked tokens are trained simultaneously with originals. Additionally, the semantic alignment loss further ensures masked samples remaining semantically close to original ones. The following joint loss is utilized:

$$\mathcal{L} = \frac{1}{2}\mathcal{L}_{EQA} + \frac{1}{2}\mathcal{L}_{EQA}^{\texttt{[Mask]}} + \lambda\mathcal{L}_{ALI}, \qquad (7)$$

where λ is a penalty regularizer. During the inference, the [Mask] training and semantic alignment modules are discarded, and testing samples follow the traditional steps to extract latent representation via the trained encoder (**F**), before applying the classifiers ($\text{MLP}^{s/e}$) to identify the start and end position of answers.

4 Experiments

4.1 Setup

Experiments and analysis are carried on a collection of high-competitive EQA tasks. Specifically, six benchmarking datasets from MRQA 2019 (Fisch et al., 2019) are employed, including SQuAD (1.1), HotpotQA, NewsQA, NaturalQ, TriviaQA, and SearchQA. Their statistics are provided in Appendix A.1.

The RoBERTa-base model (Liu et al., 2019) is adopted as the encoder. In addition, with both *Gaussian* and *U-shaped* masking, the reference index r is set as the beginning location of the ground-truth answer token (from training samples), while ground-truth answer tokens are assigned with zero masking probability (never masked). The hyper-parameter is set as $\Delta = 2$ for Eq. (3). More training details are provided in Appendix A.2. The F1-evaluation metric, measured by the number of overlapping tokens between the predicted and ground-truth answers, is adopted.

4.2 Main results

Our proposed method is compared with the following models. We re-implement them using provided source codes and results are competing with the reported.

- Base (Liu et al., 2019) is implemented using the pre-trained vanilla model (RoBERTa-base) and fine-tuned as described in Section 3.1;
- BLANC (Seonwoo et al., 2020) applies a block-attention strategy to predict answers and supporting contexts (spans surrounding around answers) simultaneously;
- SSMBA (Ng et al., 2020) randomly substitutes tokens with [Mask] and recovers them to produce new samples for data augmentation;
- SWEP (Lee et al., 2021) augments the data by perturbing the input embedding with an adjustable Gaussian noise;
- KALA (Kang et al., 2022) augments the original contextual representation using related entity and relational representation from the external Knowledge Graph.

For M4A we set $\lambda = 0.5$ (the ablation study of λ is provided later), the masking budget (or the the fraction of masked tokens) as b_m = 20% with the *U-shaped* masking strategy (the impact of different masking strategies are also offered in the ablation study).

Table 1: Comparison among M4A and existing methods. Specifically, M4A achieves $1.66e^{-16}$, $4.47e^{-7}$, $9.91e^{-19}$, $6.92e^{-19}$, $2.58e^{-9}$, and $5.14e^{-13}$ for SQuAD, HotpotQA, NewsQA, NaturalQ, TriviaQA and SearchQA, respectively, in terms of the p-values from the T-tests. This statistical significance testing confirms the stability of M4A.

Model	**SQuAD**	**HotpotQA**	**NewsQA**
Base	90.3±0.2	78.7±0.3	69.8±0.4
BLANC	91.1±0.2	77.8±0.1	70.7±0.5
SSMBA	90.1±0.3	77.3±0.4	69.2±0.2
SWEP	91.0±0.1	78.8±0.1	71.7±0.1
KALA	90.9±0.4	77.3±0.5	72.7±0.3
M4A	**92.2**±0.1	**79.2**±0.1	**72.8**±0.2
Model	**NaturalQ**	**TriviaQA**	**SearchQA**
Base	79.6±0.2	74.0±0.4	81.5±0.3
BLANC	80.3±0.1	75.1±0.1	82.6±0.1
SSMBA	79.8±0.2	74.8±0.2	82.1±0.1
SWEP	80.2±0.3	75.3±0.2	82.5±0.2
KALA	80.1±0.4	75.5±0.3	82.4±0.4
M4A	**81.4**±0.1	**76.3**±0.2	**83.5**±0.1

The comparison results in terms of the mean value and standard deviation over 10 runs are shown in Table 1, in which the best result for each dataset is **bolded**. M4A consistently improves

existing models across employed EQA tasks. For instance, the strongest baseline SWEP model achieves an approximately 0.8 absolute-point improvement compared to the vanilla model, while our M4A model enhance a further 1.1 absolute point over SWEP. Thus, it amounts to a comparable improvement and demonstrates the superiority of the proposed masking approach.

Notably, in terms of the computational complexity, the proposed algorithm has a same scale of model-training parameters as the vanilla model (RoBERTa-base). Specifically, M4A reuses two MLPs ($\text{MLP}^{s/e}$ for classifying the start/end answer token from Eq. (1)) in the `[Mask]` training and semantic alignment modules (Eq. (5)). As such, the proposed method does not require extra model parameters. Additionally, during the inference, both the `[Mask]` training and semantic alignment modules are discarded as M4A only requires the trained encoder. As an example, M4A needs only 0.02 seconds per question on the SQuAD testing set.

4.3 Ablation study

On the encoder flexibility. To start with, we evaluate the impact from the fundamental encoder towards the proposed M4A. Specifically, the BERT-base encoder (Devlin et al., 2019) is implemented as the `Base`, and the strongest baseline `SWEP` from Table 1 is also employed for comparison purposes. Most of the experimental settings, such as the batch size and the sequence length, are in line with the above RoBERTa-base model, except the learning rate is fixed as $3e^{-5}$ and the training epoch is 2 (consistent with `SWEP`). The comparison results over 10 runs are illustrated in Fig. 4, and our M4A method achieves the best F1 score across all datasets, outperforming SWEP. This experiment provides further evidence for M4A's stability on the underlying encoder: it outperforms the current best model SWEP on both RoBERTa and BERT as the encoder. Without explicitly mentioning, the following ablation studies are conducted using the RoBERTa-base encoder.

On the masking strategy. The following ablation experiments are performed using SQuAD to evaluate the proposed masking strategies, *i.e.*, *Gaussian*, *U-shaped*, and *Uniform*, whose main difference lies in the distribution of masked tokens. Given $\lambda = 0.5$, $\Delta = 2$, and $b_m = 20\%$, the reference index r, again, for *Gaussian* and *U-shaped* is set as the start location of the ground-truth answer token(s), respectively. As for *Uniform*, the token masking probability is set as b_m.

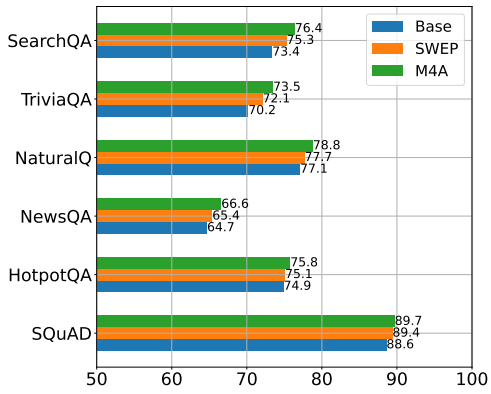

Figure 4: Impact analysis from the underlying encoder.

Table 2: Comparison of the answering performance achieved by different masking strategies.

Base	Gaussian	U-shaped	Uniform
90.3±0.2	91.7±0.1	92.2±0.1	92.1±0.1

Table 2 compares the three proposed masking strategies, all of which obtain the similar, and higher accuracy than the `Base` model. The result clearly demonstrates the effectiveness of M4A to utilize masked samples for solving EQA. In addition, the U-shaped masking strategy, which masks less tokens around the ground-truth answer(s) but more towards the begin/end of the entire passage, performs the best on the SQuAD dataset. This finding is consistent with BLANC (Seonwoo et al., 2020), from which supporting contexts (spans surrounding around answers) are highlighted via a predetermined soft label. In contrast, our method encourages to remove (via masking) tokens that are far away from answer tokens. Accordingly, the U-shaped masking strategy is adopted for all following experiments. However, we need to point out that the differences among these three masking distributions are relatively moderate (*e.g.*, ±0.25 F1). The contributor to this effectiveness is explored later on the module breakdown part.

On the masking budget. The impact from the masking budget (or the the fraction of masked tokens) b_m is validated hereafter. Obviously, more tokens are masked out with a higher value of b_m, which also leads to more noisy samples (due to the increase of masked or missing tokens). Specifically, experiments are conduced by varying b_m from 10%

to 100% on the SQuAD dataset. Notably, with b_M=100%, all tokens except ground-truth ones are removed.

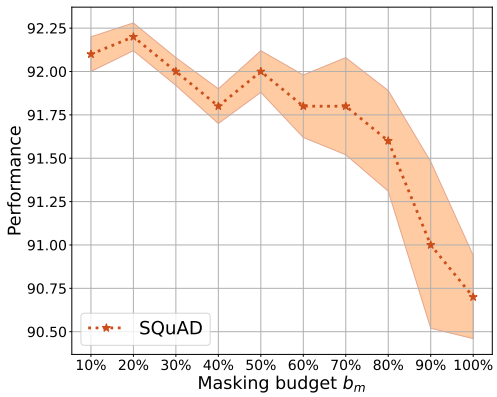

Figure 5: Performance comparison as a function of the masking budgets b_m.

The comparison is shown in Fig 5, and results show that the proposed method is relatively insensitive to b_m. For instance, with SQuAD the highest performance is observed as 92.2±0.1 (b_m = 20%), which decreases to 90.7±0.2 (b_m = 100%). Surprisingly, even with b_M=100%, M4A still obtains a slightly better performance than the `Base` model (90.3 from RoBERTa-base), which shows the lower bound of M4A is the vanilla model. That is, given the very incomplete (masked) passages, M4A is still capable of identifying correct answers, which empirically confirms the robustness and stability of our proposed masking strategy. Furthermore, we argue that such a stable performance (with high masking rate) is a result of the semantic alignment. In the extreme case of b_M=100%, for instance, only ground-truth tokens are maintained in the masked sample, and presumably the contribution from the `[Mask]` training loss becomes limited; yet, the semantic alignment loss enforces the encoder to produce robust features for sample with/out masking masking. We further investigate this hypothesis in the model breakdown study below.

On the module breakdown. The following experiment examines the effectiveness of M4A from aspects of masked training and semantic alignment. Specifically, the comparison is considered with the following variants: `Base` represents the model trained using only original samples (*w.r.t.* the first term of Eq. (7)); `Mask` only employs U-shaped masking samples for the model training (*w.r.t.* the second term of Eq. (7)); and `Alignment` maximizes the semantic similarity between the original-masked pairs (*w.r.t.* the last term of Eq. (7)).

Table 3: Examination on individual modules of training with masked copies and aligning original-and-masked semantics.

`Base`	`Mask`	`Base+Mask`
90.3	90.0	90.7
`Base+Alignment`	`Mask+Alignment`	`Full`
91.5	91.0	92.2

Table 3 shows contributions from individual modules using SQuAD with λ = 0.5 and b_m = 0.2. To begin with, both the proposed masked training and alignment modules stably improve the performance of the `Base` model. Training with only masked samples, as observed, the `Mask` variant achieves the worst result; the reason is mainly due to the noise brought by the incomplete (or masked) passage(s). Yet, `Base+Mask` achieves the accuracy of 90.7 for SQuAD, which shows the benefit of employing masked copies as the data augmentation. Furthermore, `Alignment` brings a bigger performance boost in comparison with `Mask`. The former achieves the averaged improvement of 1.2 points on top of `Base`, while the latter only obtains a boost of 0.4 points. More importantly, that is also evidenced by the second-place score from `Base+Alignment`, which trains the model without masked samples but only enforces the semantic similarity between the original and masked samples. In other words, with the presence of masked samples, the key contributor to M4A is to restrict their semantic alignment with the original ones (not the way of producing those masked samples, such as the masking budget and the masking strategy). The alignment loss regulates the encoder to generate robust features so to improve the model robustness and performance.

It is worth noting that, the utilization of the masking model proves essential not only for generating supplementary samples that complement the originals, but also for facilitating the Semantic Alignment module to align these masked samples with the original ones. In other words, masked samples form the foundational element on which the alignment depends, although the key contributor to M4A is the semantic alignment. Yet, the proposed masking strategy is also cost-effective, which stands in stark contrast to existing data augmentation techniques that often demand prior domain knowledge and/or resources.

On the penalty regularizer λ. The model

accuracy is also evaluated by varying values of λ from 0.1 to 1.0. The comparison is shown in Fig 6 with SQuAD. The result illustrates that the proposed method is robust to λ, as M4A achieves a stable performance for different scenarios (with an average F1 difference of ± 0.35 approximately).

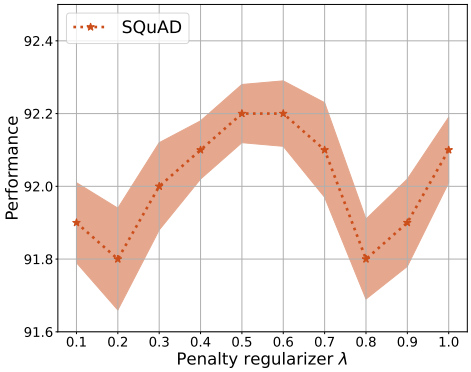

Figure 6: Performance comparison as a function of different λ.

On the data augmentation. This section validates the proposed model from the aspect of the low-resource training, as samples with masked tokens could play a role of the data augmentation. Accordingly, only a small amount (say m) of samples (randomly selected from the training set) are utilized for masking and further fine-tuning the model, where $m = [20\%, 40\%, 60\%, 80\%, 100\%]$ ($m = 100\%$ represents the full dataset).

Table 4: Averaged performance (measured by F1) obtained by SWEP/M4A under different percentages of training samples.

SQuAD	20%	40%	60%	80%	100%
SWEP	88.4	89.6	90.1	90.7	91.0
M4A	88.6	90.4	91.5	91.9	92.2

Table 4 shows the accuracy as a function of the sample size using the SQuAD dataset. Compared to the strongest baseline (`SWEP` from Table 1), M4A consistently improves the model performance under all percentages of the training samples. With 40% of labeled data, M4A has achieved even higher accuracy than `SWEP` with 60% samples. Empirically, the result clearly demonstrates the superiority of produced samples with `[Mask]` tokens as an effective data augmentation.

On out-of-domain generalizability. In this experiment, we measure M4A's ability via adapting to unseen domains (*i.e.*, datasets) in a zero-shot manner. Following `SSMBA` and `SWEP`, the model is first trained on a single source dataset (SQuAD in this case), and further evaluated on the other two target dataset (*i.e.*, HotpotQA and NaturalQ) without any fine-tuning[†].

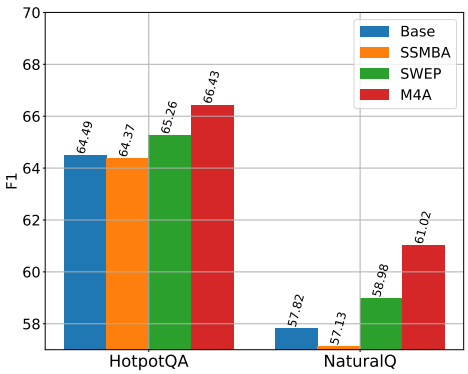

Figure 7: Transferring the EQA model trained by SQuAD to unseen datasets.

Averaged performance from our proposed and other methods is presented in Fig. 7. The `Base` method (directly applying the RoBERTa-base model trained from SQuAD) achieves the worst F1 (60.1 on average) on three subsequent target datasets, which reveals the different data distribution among diverse datasets. Therefore, a direct application of the `Base` model to downstream dataset(s) is ineffective as shown by its low generalizability. Surprisingly, `SSMBA` achieves even worse results than `Base` on HotpotQA and NaturalQ. Notably, `SSMBA` reconstructs masked tokens to produced augmented samples. Yet, new substituted tokens may still cause the semantic drift and lead to poor performance on test sets. In contrast, the average performance of M4A (62.58) is highest across three target datasets. Due to masked samples, M4A is encouraged to produce more robust features, which contributes to the model generalizability to unseen datasets. The comparison indicates that M4A can be regarded as a supplementary pre-training method, along with the traditional Masked Language Modeling (MLM), to offer a robust starting point for fine-tuning unseen datasets.

5 Conclusion

This paper proposes a `[Mask]`-for-Answering algorithm (M4A) to tackle the Extractive Question Answering (EQA) task, via simulating human-comprehension skills to infer answers with limit

[†]Notably these three datasets share the same Wikipedia domain background, as shown in Table 5 from Appendix A.1.

context. Specifically, three different masking strategies are introduced to produce masked copies of original samples. Those masked samples are directly utilized for the model training, and regularized by an alignment loss to ensure their semantic similarity with originals. Empirically, M4A achieves statistically better performance than current methods on several benchmark EQA datasets. In addition, M4A also demonstrates the strong capability in the settings of the low-resource training and zero-shot domain adaptation. At last, this masking-then-training strategy, explored by M4A, is also agnostic to downstream tasks, and we will incorporate it with other downstream applications.

Limitations

`[Mask]` tokens play a critical role, for our proposed M4A, in improving the model inference capability even with masked contents. Yet, the proposed masking strategies mainly focus on the distribution of masked tokens; intuitively, it still cannot be guaranteed that all question-irrelevant contents are masked (or removed). Therefore, masking strategies can be further refined, such as by incorporating external/prior knowledge about passages/questions, masking (stop) words with specific POS, or introducing dependency parsing for identifying trivial words.

Additionally, regarding the Jensen-Shannon divergence employed in this paper, we acknowledge its crucial role in semantic alignment. However, other methods exist for calculating semantic similarity. We defer exploring these alternatives to future work, as they hold potential to enhance the performance further

At last, while current Large Language Models (LLMs), such as GPT-3.5 and Vicuna, achieve impressive achievements across different tasks, including EQA, the proposed method is much more lightweight, based on models that are orders of magnitude smaller (*e.g.*, 110M parameters for BERT-base compared to 7B for Vicuna). Nevertheless, our ablation study highlights the adaptability of our approach as it can be applied to other encoder-decoder language models in a plug-and-play manner. We anticipate enhanced performance by adopting our proposed method using LLMs like Vicuna for EQA.

Ethical Statement

In terms of reproducibility, we have made the experimental source code available anonymously. Additionally, the benchmark datasets used in our study are publicly accessible. However, it is important to acknowledge the potential presence of hidden biases from Pre-trained Language Models, stemming from biased data they were trained on. Despite utilizing these pre-trained language models for encoding, we did not encounter any biased outcomes. Nevertheless, we carefully considered the low-risk nature of our specific domain during the study.

Acknowledgments

We express our sincere gratitude to the anonymous reviewers for their invaluable feedback, which has significantly enhanced the depth and presentation of our research. This work was partially supported by the Australian Research Council Discovery Project (DP210101426), the Australian Research Council Linkage Project (LP200201035), and AEGiS Advance Grant(888/008/268), University of Wollongong.

References

Ning Bian, Xianpei Han, Bo Chen, Hongyu Lin, Ben He, and Le Sun. 2021. Bridging the Gap between Language Model and Reading Comprehension: Unsupervised MRC via Self-Supervision. volume abs/2107.08582. ArXiv preprint.

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.

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 the 2nd Workshop on Machine Reading for Question Answering*, pages 1–13, Hong Kong, China. Association for Computational Linguistics.

Li Z. Lin Z. Zhu Y. Leng J. Guo M. Guan, Y. 2022. Block-skim: Efficient question answering for transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10710–10719.

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.

Minki Kang, Jinheon Baek, and Sung Ju Hwang. 2022. KALA: knowledge-augmented language model adaptation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5144–5167, Seattle, United States. Association for Computational Linguistics.

Seanie Lee, Minki Kang, Juho Lee, and Sung Ju Hwang. 2021. Learning to perturb word embeddings for out-of-distribution 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 5583–5595, Online. Association for Computational Linguistics.

Junping Liu, Shijie Mei, Xinrong Hu, Xun Yao, Jack Yang, and Yi Guo. 2022. Seeing the wood for the trees: a contrastive regularization method for the low-resource knowledge base question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1085–1094, Seattle, United States. Association for Computational Linguistics.

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. volume abs/1907.11692. ArXiv preprint.

Nathan Ng, Kyunghyun Cho, and Marzyeh Ghassemi. 2020. SSMBA: Self-supervised manifold based data augmentation for improving out-of-domain robustness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1268–1283, Online. Association for Computational Linguistics.

Scott G. Paris, Marjorie Y. Lipson, and Karen K. Wixson. 1983. Becoming a strategic reader. *Contemporary Educational Psychology*, 8(3):293–316.

Ori Ram, Yuval Kirstain, Jonathan Berant, Amir Globerson, and Omer Levy. 2021. 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.

Yeon Seonwoo, Ji-Hoon Kim, Jung-Woo Ha, and Alice Oh. 2020. Context-aware answer extraction in question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2418–2428, Online. Association for Computational Linguistics.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2019. Improving machine reading comprehension with general reading strategies. 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 2633–2643, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuanhe Tian, Yan Song, and Fei Xia. 2022. Improving relation extraction through syntax-induced pre-training with dependency masking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1875–1886, Dublin, Ireland. Association for Computational Linguistics.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Dongjie Yang, Zhuosheng Zhang, and Hai Zhao. 2023. Learning better masking for better language model pre-training. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7255–7267. Association for Computational Linguistics.

Adams Wei Yu, Hongrae Lee, and Quoc Le. 2017. Learning to skim text. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1880–1890, Vancouver, Canada. Association for Computational Linguistics.

Chenbin Zhang, Congjian Luo, Junyu Lu, Ao Liu, Bing Bai, Kun Bai, and Zenglin Xu. 2020. Read, attend, and exclude: Multi-choice reading comprehension by mimicking human reasoning process. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1945–1948, New York, NY, USA. Association for Computing Machinery.

Neşet Özkan Tan, Alex Yuxuan Peng, Joshua Bensemann, Qiming Bao, Tim Hartill, Mark Gahegan, and Michael Witbrock. 2023. Input-length-shortening and text generation via attention values. *arXiv, 2303.07585*.

A Appendix

A.1 Employed benchmarks

The statistics for employed datasets are provided in Table 5.

A.2 Training Details

The RoBERTa-base model (Liu et al., 2019) is adopted as the encoder. The dropout rate across all layers is set as 0.1. The Adam optimizer with a dynamic learning rate is adopted, for which the learning rate is warmed up for 10 thousand steps to a maximum value of $1e^{-4}$ before decaying linearly

Table 5: Employed datasets for the EQA task, where **Domain** represents the passage resource, and **#Train** and **#Test** is the number of training and test samples, respectively.

Dataset	**Domain**	**#Train**	**#Test**
SQuAD(1.1)	Wikipedia	86,588	10,507
HotpotQA	Wikipedia	72,928	5,904
NewsQA	News articles	74,160	4,212
NaturalQ	Wikipedia	104,071	12,836
TriviaQA	Web snippets	61,688	7,785
SearchQA	Web snippets	117,384	16,980

to a minimum value of $2e^{-5}$. The training is performed with batches of 8 sequences of length 512. The maximal number of training epoch is 10. For each dataset, 1,000 samples are randomly selected from the training set to form the validation set, and training stops when the validation accuracy fails to improve for one epoch. At last, the proposed model is trained on a machine with four Tesla K80 GPUs.