Towards robust token embeddings for Extractive Question Answering

Xun Yao[1], Junlong Ma[1], Xinrong Hu[1], Jie Yang[2], Yi Guo[3], and Junping Liu[1]

[1] School of Computer Science and Artificial Intelligence, Wuhan Textile University
{yaoxun,2015363059,hxr,jpliu}@wtu.edu.cn

[2] School of Computing and Information Technology, University of Wollongong, Australia
jiey@uow.edu.au

[3] School of Computer, Data and Mathematical Sciences, Western Sydney University, Australia
y.guo@westernsydney.edu.au

Abstract. Extractive Question Answering (EQA) tasks have gained intensive attention in recent years, while Pre-trained Language Models (PLMs) have been widely adopted for encoding purposes. Yet, PLMs typically take as initial input token embeddings and rely on attention mechanisms to extract contextual representations. In this paper, a simple yet comprehensive framework, termed perturbation for alignment (PFA), is proposed to investigate variations towards token embeddings. A robust encoder is further formed being tolerant against the embedding variation and hence beneficial to subsequent EQA tasks. Specifically, PFA consists of two general modules, including the embedding perturbation (a transformation to produce embedding variations) and the semantic alignment (to ensure the representation similarity from original and perturbed embeddings). Furthermore, the framework is flexible to allow several alignment strategies with different interpretations. Our framework is evaluated on four highly-competitive EQA benchmarks, and PFA consistently improves state-of-the-art models.

Keywords: Extractive Question Answering · Token embedding · Contextual representation · Wasserstein distances · Divergence

1 Introduction

Extractive Question Answering (EQA) is a fundamental task for Machine Reading Comprehension (MRC), that aims to identify the answer span (a sequence of continuous words) over the given question and passage. Recent years have witnessed a remarkable success in utilizing Pre-trained Language Models (PLMs) to address EQA [2, 6, 9]. Approaches usually consist of a three-step process. In the first step, each input token is linked with an **embedding** vector via a pre-determined lookup table. The second step is to further utilize those token embeddings and estimate their contextual **representation**, followed by a decision layer (*i.e.*, a binary classifier to identify the start and end position of the answer span) as the last step[4].

The majority research on EQA has focused on extracting contextual-aware representation (the second step) using a variety of attention mechanisms, ranging from the standard self-attention [2], block based [10, 12], gated adapter [16], and hierarchical flow [14, 18], *etc.* While the contextual representation has been tremendously critical, token embeddings (from the first step) still remain fundamentally significant: (1) as the input to the second step, embeddings have a direct impact on forming the subsequent contextual representation; (2) each token has one initial and fixed embedding (regardless of surrounding context), while the variation of token embeddings towards the downstream task is underexplored.

To alleviate the aforementioned gap, a simple yet comprehensive framework, termed Perturbation for Alignment (PFA), is proposed in this paper (illustrated in Fig 1). Our framework consists of two main modules, including embeddings perturbation and semantic alignment. The former takes as input the original token embedding to produce its perturbed version. With the presence of perturbed embeddings, the latter module is then enforced to form perturbation-invariant representation via flexible alignment strategies of the Wasserstein distances or other divergence.

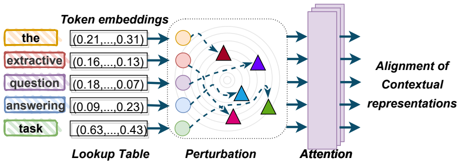


Fig. 1. PFA perturbs initial token embeddings (circles) to form their variations (triangles), and later align their semantic to increase the model robustness and generalizability.

Our work differs from existing methods in the following perspectives: (1) PFA strengthens the model generalizability via tolerating variations of token embeddings, which is neglected by the majority of existing work (focusing on subsequent contextual representations). (2) a few studies apply the noise addition ([8], requires the prior knowledge) and dropout mask ([13], depends on the hyperparameter setting) to distort token embedding. Yet, our work offers a more general framework for the embedding perturbation and the representation alignment, embracing previous methods as special cases. The main contributions of our proposed work are summarized as follows[5]:

- Our study introduces a unified framework (Perturbation for Alignment, PFA) to investigate the variation of token embeddings and its influence towards the subsequent Extractive Question Answering task.
- PFA is characterized by embeddings perturbation and semantic alignment modules, while the former perturbs original embeddings and the latter en-

[4] Without explicitly mentioned, we refer token **embeddings** as vectors obtained directly from the lookup table, while token **representations** as those obtained using attention mechanisms after the lookup.

[5] The source code is available at `https://anonymous.4open.science/r/Perturbation4EQA-1BC5`

sures contextual representations (from perturbed embeddings) remain semantically close to original ones.
- The framework is flexible to accommodate many metrics. Specifically, exploiting the Wasserstein distance used in the optimal-transport theory, PFA can be also cast as a general min-max optimization.
- Empirically, PFA outperforms recent-strong baselines on four standard EQA benchmarks, advancing the best state-of-the-arts by on average 1.43 absolute point in accuracy. Moreover, PFA also demonstrates a strong capability in the setting of low-resource fine-tuning and out-of-domain generalizability.

2 Related work

Given the input pair of question ($\boldsymbol{q}$) and passage ($\boldsymbol{p}$), Extractive Question Answering (EQA) aims to identify the start and end positions of the answer span ($\boldsymbol{a}_{s/e}$) from $\boldsymbol{p}$. Specifically, the input of EQA is a tokenized sequence, *i.e.*, $\texttt{[CLS]}\,\boldsymbol{p}_1\boldsymbol{p}_2\cdots\boldsymbol{p}_{|\boldsymbol{p}|}\,\texttt{[SEP]}\,\boldsymbol{q}_1\boldsymbol{q}_2\cdots\boldsymbol{q}_{|\boldsymbol{q}|}\,\texttt{[SEP]}$, where $\boldsymbol{p}_i$ and $\boldsymbol{q}_j$ represents the i-th and j-th token from $\boldsymbol{p}$ and $\boldsymbol{q}$, respectively. At first, individual tokens (say $\boldsymbol{p}_i$) are represented by their own static embeddings (say $\mathcal{S}(\boldsymbol{p}_i)$) from a preset lookup table. Then the encoder ($\mathcal{F}$, a Pre-trained Language Model (PLM) such as BERT [2] or RoBERTa [9]) is applied to induce the following probability distribution:

$$\mathrm{p}(\boldsymbol{p}_i = \boldsymbol{a}_{s/e}) \triangleq \frac{\exp(\mathcal{F}(\mathcal{S}(\boldsymbol{p}_i))^T \boldsymbol{w}_{s/e})}{\sum_j^{|\boldsymbol{p}|} \exp(\mathcal{F}(\mathcal{S}(\boldsymbol{p}_j))^T \boldsymbol{w}_{s/e})}, \tag{1}$$

where $\boldsymbol{w}_{s/e}$ is the learnable parameter from a decision layer (usually performed as a multilayer perceptron (MLP)). Accordingly, the loss function is defined as follows:

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

where $\mathbb{1}(\cdot)$ is the indicator function that returns 1 if the condition is true and returns 0 otherwise. Notably, the majority of existing models focus on utilizing PLMs to form contextual-aware representations via different enhancement strategies. For instance, a block-based attention method is proposed in [12], which predicts answers and supporting words (*i.e.*, contexts highly-relevant to answers). Another similar work is found in [10]. In [14], different-level attention mechanisms are implemented to simulate the process of back-and-forth reading, highlighting, and self-assessment, while [18] simulates the human-reading strategy of reading-attending-excluding to train PLMs. In addition, the work [16] integrates a gated-attention adapter with PLMs to identify answers. More recently, KALA [7] 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.

On the other hand, another line of work considers the embedding-focused strategy. Typically, input tokens are manipulated to produce crafted examples (including token deletion [17] or replacement [11]), which is equivalent to modifying input embeddings. Additionally, Srivastava *et al.* consider to apply the dropout mask on embeddings [13], and SWEP [8] augments them with adjustable noises. The subsequent EQA model is further trained using both original and perturbed embeddings simultaneously. Yet, token manipulation methods could decrease the model accuracy due to over-fitting crafted examples, while other methods require the prior knowledge to select the dropout masking rate [13] or to follow a multivariate Gaussian distribution [8]. In contrast, our work provides a hyperparameter-free transformation framework to perturb token embeddings and later align their semantic to improve the robustness of the encoder.

3 Methodology

The proposed perturbation-for-alignment (PFA) framework is detailed in this section via introducing two general modules: embedding perturbation (EP) and semantic alignment (SA). Specifically, the EP module perturbs input embeddings, while the SA module ensures the semantic-representation similarity from original and perturbed embeddings. Proposed modules are integrated into a unified framework and are fully end-to-end trainable (shown in Fig 2).

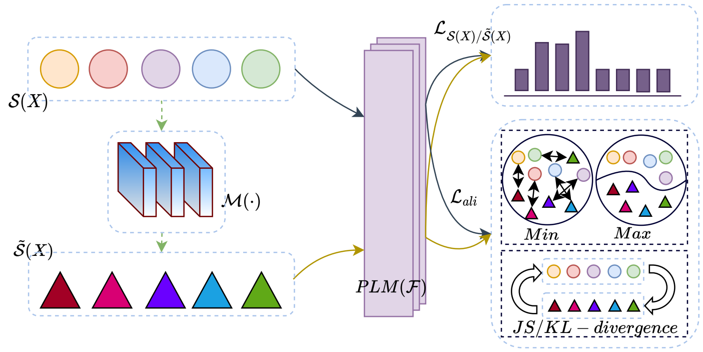


Fig. 2. Illustration of the proposed PFA framework for EQA. Original inputs are distorted via a embedding perturbation module, while their representation similarity is later maximized via a semantic alignment module.

3.1 PFA

Embedding perturbation. Let $\mathcal{S}(\boldsymbol{X})$ be the embedding for the tokenized passage sequence $\boldsymbol{X}(=\boldsymbol{p}_1\boldsymbol{p}_2\cdots\boldsymbol{p}_{|\boldsymbol{p}|})$. The traditional perturbation mainly involves adding the noise element-wisely (*i.e.* $\tilde{\mathcal{S}}(\boldsymbol{X}) = \mathcal{S}(\boldsymbol{X}) + \text{noise}$) or sampling with a dropout mask (*i.e.* $\tilde{\mathcal{S}}(\boldsymbol{X}) = \text{dropout}(\mathcal{S}(\boldsymbol{X}))$). In contrast, this paper introduces a more-general transformation to produce $\tilde{\mathcal{S}}(\boldsymbol{X})$. Specifically, a multilayer perceptron (MLP), written as $\mathcal{M}(\cdot)$, with one-hidden layer is employed. Accordingly, the proposed embedding perturbation is formulated as follows:

$$\tilde{\mathcal{S}}(\boldsymbol{X}) = \mathcal{M}(\mathcal{S}(\boldsymbol{X})). \tag{3}$$

Note that $\mathcal{M}(\cdot)$ can be any function, for example the existing dropout or noise superimposition, which effectively makes the previous methods special cases. The application of MLP here however, provides a family of transformations due to its functional form and well known universal approximation capability.

Additionally, to avoid triviality, *i.e.*, $\mathcal{M}(\cdot)$ being an identity mapping, we employ the bottleneck-structure technique to implement the MLP [4, 5], *i.e.*, from which the hidden layer has a smaller size than its adjacent layers. However, we point out that it might not be necessary due to the complex nature of the loss function landscape, especially the non-convexity. Interestingly, the size of the bottleneck becomes a controllable factor, towards generating a better performing $\mathcal{F}$ (which is evaluated in our ablation study). Overall, different from the additive noise or the random dropout, $\mathcal{M}(\cdot)$ offers much greater flexibility to program the perturbed (embedding) distribution within the same space.

Semantic alignment. With perturbed embeddings, the semantic alignment module is further employed to robustify the encoder $\mathcal{F}$ via tolerating perturbed signals conveyed in $\tilde{\mathcal{S}}(\boldsymbol{X})$. This is done by explicitly encouraging the representation similarity after encoding original and perturbed embeddings. Let $\mathbf{C}$ and $\tilde{\mathbf{C}}$ represent latent representation associated with original and perturbed embeddings, *i.e.*, $\mathbf{C} = \mathcal{F}(\mathcal{S}(\boldsymbol{X}))$ and $\tilde{\mathbf{C}} = \mathcal{F}(\tilde{\mathcal{S}}(\boldsymbol{X}))$, where $\mathbf{C}/\tilde{\mathbf{C}} \in \mathbb{R}^{|\boldsymbol{X}| \times l}$ and l is the hidden dimension. The alignment objective can be formulated as minimizing the following `instance`-wise distance:

$$\mathcal{L}^{i}_{ali} = \mathrm{dist}(\mathbf{C}, \tilde{\mathbf{C}}), \tag{4}$$

where $\mathrm{dist}(\cdot, \cdot)$ is a distance function. As such, this proposed loss minimization ensures the original semantic is preserved even with perturbations, to further improve the model tolerance to input variations. Moreover, in addition to matching collections of instances as in Eq. (4), one can also align their representations at the `distribution` level. Let $P_{\mathbf{C}}$ and $P_{\tilde{\mathbf{C}}}$ be the corresponding distributions of $\mathbf{C}$ and $\tilde{\mathbf{C}}$. Therefore, the alignment objective can also be reformulated as:

$$\mathcal{L}^{d}_{ali} = \mathrm{dist}(P_{\mathbf{C}}, P_{\tilde{\mathbf{C}}}). \tag{5}$$

The analysis of alternative implementations (including the distance function) will be detailed in Section 3.2.

Overall objective function. In summary, the following joint loss is utilized for the proposed Perturbation-for-Alignment (PFA) framework:

$$\mathcal{L} = \mathcal{L}_{\mathcal{S}(\boldsymbol{X})} + \mathcal{L}_{\tilde{\mathcal{S}}(\boldsymbol{X})} + \mathcal{L}^{i/d}_{ali}, \tag{6}$$

where $\mathcal{L}_{\mathcal{S}(\boldsymbol{X})}$ represents the standard (cross-entropy) loss using the original embeddings (followed by Eq. (2)), while $\mathcal{L}_{\tilde{\mathcal{S}}(\boldsymbol{X})}$ is the loss obtained by swapping $\mathcal{S}(\cdot)$ with $\tilde{\mathcal{S}}(\cdot)$ in Eq. (1) and further inferring answers as Eq. (2). The last term then ensures that perturbed embeddings remain semantically close to the original ones. During inference, the EP and SA modules are discarded, and testing samples follow the traditional three steps to: (1) lookup token embeddings, (2) extract latent representation via the trained encoder, (3) apply the trained decision layer to identify the start and end position of answers.

3.2 Analysis

There are many choices for aligning original and perturbed representations, either from the instance (Eq. (4)) or distribution level (Eq. (5)). For instance, the matrix Frobenius Norm and cosine similarity can be leveraged for the instance alignment, *e.g.* forcing representation vectors being similar. In this paper, we are particularly interested in the distribution-level alignment, for which the Wasserstein distance [15] is employed as a measurement:

Definition 1 (Wasserstein distance [15]) *Let $(\mathcal{X}, d)$ be a Polish metric space, and $p \in [1, \infty]$. For any two probability measures μ, ν on $\mathcal{X}$, the Wasserstein distance of order p between μ and ν is defined by*

$$W_p(\mu, \nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathcal{X}} d(x,y)^p d\pi(x,y) \right)^{\frac{1}{p}}, \tag{7}$$
$$s.t. \int_{\mathcal{X}} \pi(x,y)dy = \mu, \int_{\mathcal{X}} \pi(x,y)dx = \nu.$$

where $\Pi(\mu, \nu)$ is the set of all couplings of μ and ν.

It is metrized version of the optimal transportation (OT) cost, where $d(x, y)$ is replaced by a cost function $c(x, y) \geq 0$ that is not necessarily a metric. The one we are particularly interested in is W_1, *i.e.*, when $p = 1$, usually called Kantorovich-Rubinstein distance. In a nutshell, the Wasserstein distance or OT cost is measuring the dissimilarity of two sets exhausting all possible joint probabilities given that the marginal probabilities are fixed. This is an ideal choice towards our purpose of aligning representations from original and perturbed embeddings. Notably, there are many variants for the OT cost, for example the Sinkhorn distance [1], defined for discrete observations. Therefore, the proposed SA module is not limited to a particular measure but open to vast possibilities. One interesting extension comes from the dual form of Eq. (7) [15]:

$$W_1(\mu, \nu) = \sup_{f, \|f\|_{\text{Lip} \leq 1}} \left\{ \int_{\mathcal{X}} f d\mu - \int_{\mathcal{X}} f d\nu \right\}, \tag{8}$$

where f is a measurable function defined from $\mathcal{X}$ to $\mathbb{R}$ and $\|f\|_{\text{Lip}}$ is the Lipschitz constant of f, *a.k.a* $\|f\|_{\text{Lip}} = \inf\{c : cd(x, y) \geq |f(x) - f(y)|\}, \forall x, y \in \mathcal{X}$. The dual form of W_1 distance indicates that one can maximize Eq. (8) over all nicely smooth functions defined on $\mathcal{X}$ to obtain its value. In our context, if we choose μ and ν to be probability measures of embeddings, *i.e.*, $\mathcal{S}(\boldsymbol{X})$ and $\tilde{\mathcal{S}}(\boldsymbol{X})$, then Eq. (8) shows that W_1 distances between these two distributions is the largest mode differences under all smooth transformations as the integrals are exactly the expectations of f. This not only gives us the interpretation of the distribution alignment, but also provides a way to compute the W_1 distance approximately, for example, via a regularization

$$W_1(\mu,\nu) \approx \max_f \int_{\mathcal{X}} f d\mu - \int_{\mathcal{X}} f d\nu - \gamma \int_{\mathcal{X}} \|\nabla f\|^2 d\xi, \tag{9}$$

for a hyperparameter $\gamma \geq 0$ (*e.g.* $\gamma = 1$). Note that the regularization term, the last term in Eq. (9), which is an integral with respect to some Radon measure ξ, is to clamp f roughly to be 1-Lipschitz. To further simplify the computation, we choose

$$\xi = (1-\alpha)\mu + \alpha\nu, \tag{10}$$

for any $\alpha \in (0,1)$. Accordingly, Eq. (9) can be effectively rewritten as the following

$$W_1(\mu,\nu) \approx \max_f \int_{\mathcal{X}} (f - \gamma(1-\alpha)\|\nabla f\|^2) d\mu - \int_{\mathcal{X}} (f + \gamma\alpha\|\nabla f\|^2) d\nu, \tag{11}$$

where the integrals can be computed easily by evaluating expectation. We point out that the value of α is not critical as we minimize this distance so that μ and ν become more or less the same.

Additionally, although f is nothing more than a dummy function in Eq. (9), one can have a preference on its structure and hence assign some meaning to it at the cost of further restricting the approximation capacity to the original W_1 distance. Specifically, in this paper we consider

$$f = \mathcal{H} \circ \mathcal{F}, \tag{12}$$

where $\mathcal{H}$ is a function that maps the output of $\mathcal{F}$ to $\mathbb{R}$. If we set $\mathcal{H}$ to be a binary MLP classifying instances from either $\mathcal{S}(\boldsymbol{X})$ or $\tilde{\mathcal{S}}(\boldsymbol{X})$, which is the choice in our numerical experiments, then the proposed PFA can be cast as a **min-max** optimization, *i.e.*, combining Eq. (9) with the minimization of Eq. (6), such that f cannot differentiate two distributions. Based on this understanding, we reach to Eq. (5) that admits more choices, such as the Kullback-Leibler (KL) divergence and Jensen-Shannon (JS) divergence. These divergences also serve for the purpose of Eq. (5) exactly to align $P_{\mathbf{C}}$ and $P_{\tilde{\mathbf{C}}}$.

4 Experiment

4.1 Setup

Experiments and analysis are carried out on four benchmark datasets from MRQA 2019 [3], including SQuAD (1.1), HotpotQA, NewsQA, and NaturalQ. Their statistics are shown in Table 1.

Training Details. The RoBERTa-base model [9] is adopted as the contextual encoder, with the dropout rate of 0.1. The Adam optimizer with a dynamic

Table 1. 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

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 to a minimum value of $2e^{-5}$ (by the cosine annealing). The training is performed with batches of 8 sequences of length 512. The maximal number of training epoch is 10. The F1-evaluation metric, measured by the number of overlapping tokens between the predicted and ground-truth answers, is adopted. At last, all models are performed using a machine of the NVIDIA A100 GPU server.

4.2 Main results

Our proposed PFA method is compared with several baseline models:

- Base [9] is implemented via fine-tuning the RoBERTa-base model.
- BLANC [12] applies a block-attention strategy to predict answers and supporting contexts (spans surrounding around answers) simultaneously[6];
- SSMBA [11] randomly substitutes tokens with `[Mask]` to modify input embeddings, and then recovers them to produce new samples[7];
- SWEP [8] augments the data by perturbing the input embedding with an adjustable Gaussian noise[8];
- KALA [7] augments the original contextual representation using related entity and relational representation from the external Knowledge Graph[9].

Notably, SSMBA, SWEP and KALA can be cast as the data-augmentation methods, from the perspectives of either modifying input tokens or introducing external knowledge. Therefore, PFA is compared with them specifically to evaluate its capability on the data augmentation, as perturbed embeddings also play a role in bringing additional training samples. All contender methods are re-implemented using their released code packages and kept with the original configurations. Additionally, the embedding perturbation (EP) module is implemented using a one-hidden-layer MLP with the bottleneck structure, where the size for the input, hidden (or `bottleneck`) and output layer is 768, 500, and 768, respectively, and a LeakyReLU activation function. For the semantic alignment (SA) module, the W_1 distance is adopted together with a binary classifier for $\mathcal{H}$ (from Eq. (12)), and α=0.5.

The comparison from Table 2 provides the strong evidence for the proposed PFA in addressing the EQA task. The proposed method consistently improves

[6] available from `https://github.com/yeonsw/BLANC`

[7] available from `https://github.com/nng555/ssmba`

[8] available from `https://github.com/seanie12/SWEP`

[9] available from `https://github.com/Nardien/KALA`

Table 2. Comparison among PFA and existing methods, in which the best result is **bolded**. Statistically significant gains achieved by the proposed method at p-values <0.01 are marked with †.

	SQuAD	**HotpotQA**	**NewsQA**	**NaturalQ**
Base	90.3±0.2	78.7±0.3	69.8±0.4	79.6±0.2
BLANC	91.1±0.2	77.8±0.1	70.7±0.5	80.3±0.1
SSMBA	90.1±0.3	77.3±0.4	69.2±0.2	79.8±0.2
SWEP	91.0±0.1	78.6±0.2	71.7±0.1	80.2±0.3
KALA	90.9±0.4	77.3±0.5	72.7±0.3	80.1±0.4
PFA	**92.4**±0.2†	**79.3**±0.1†	**73.3**±0.4†	**82.2**±0.3†

the state-of-the-art models. For instance, PFA outperforms the strongest baseline (SWEP) by 1.4, 0.7, 1.6, and 2.0 absolute points with respect to the SQuAD, HotpotQA, NewsQA, and NaturalQ datasets, respectively. We also notice that the KALA method relies on the quality of the constructed Knowledge Graph, which leads to a notable performance variation. For instance, KALA achieves a even worse result than the Base model on the HotpotQA dataset. In addition, the significance test (*i.e.*, the one-sample T-test) is also implemented, and the p-values of our results being greater than relevant strongest baselines are $6.2e^{-6}$, $1.3e^{-7}$, $5.6e^{-7}$, and $4.9e^{-8}$, respectively. The results clearly verifies the effectiveness and stability of our proposed method.

On the other hand, in terms of the computational complexity, PFA has a similar scale of parameters as the vanilla model (RoBERTa-base). Specifically, during training PFA only needs additional parameters of two MLP classifiers for Eq. (3) and Eq. (12); that is, approximately 2.4M parameters are added (compared to the original 128.0M of RoBERTa-base). During inference, both the EP and SA modules are discarded as PFA only requires the trained encoder.

4.3 Ablation study

Experiments are conducted using the SQuAD dataset, and all results are reported as an averaged F1 over 10 runs.

On the encoder flexibility. We first evaluate the impact from the fundamental encoder. Specifically, the BERT-base encoder [2] is employed as the `Base`, and the strongest baseline `SWEP` and `KALA` from Table 2 is also re-implemented. Most of the experimental settings, such as the batch size and the sequence length, are the same as RoBERTa, except the learning rate is set as $3e^{-5}$. Comparison results are presented in Table 3, where PFA demonstrates highest F1 scores across all datasets. These findings highlight the stability/robustness of PFA on the underlying encodes (both RoBERTa and BERT), outperforming current best models. To maintain consistency, subsequent ablation studies are still conducted using RoBERTa.

Table 3. Impact analysis of the underlying encoder.

	Base	SWEP	KALA	PFA
SQuAD	88.6	89.4	89.2	89.7
HotpotQA	74.9	75.1	75.3	75.8
NewsQA	64.7	65.4	65.8	66.6
NaturalQ	77.1	77.7	77.5	78.8

On the breakdown. We investigate individual aspects of EP and SA to manifest their efficacy. Specifically, for comparison purposes we take "Base" to represent the vanilla training; "Base+EP" applies both original and perturbed embeddings for fine-tuning the model; "Base+SA" considers to utilize original embeddings, without perturbed ones, for the model training, in addition to the semantic similarity of original-perturbed pairs. The EP is implemented using a MLP with a Bottleneck layer of the size 500 and the W_1 distance for SA.

The results are summarized in Table 4, while all variants stably improve F1 scores. Specifically, with both original and perturbed embeddings (*i.e.*, Base+EP), the model obtains 91.2, indicating the benefit of employing perturbation as a special data augmentation. Interestingly, we also observe that the Base+SA model achieves a even higher performance (91.7). Notably, the SA module enforces the similarity representation of original-perturbed pairs. That is, the encoder is fine-tuned so that the presentation of perturbed embeddings is similar to that of original ones. The perturbation tolerance of the encoder is enhanced, leading to a better F1 score. Empirically, SA brings a larger performance boost, in comparison with EP, which indicates the importance of imposing the semantic alignment to the embedding perturbation.

Table 4. Effect of individual modules on F1.

	Base	Base+EP
F1	90.3	91.2
	Base+SA	Base+EP+SA
F1	91.7	92.4

On the EP module. In addition to the MLP generator in Eq. (3), we also consider to perturb embeddings via an element-wise noise addition and a dropout mask, respectively, while the SA module is fixed using the W_1 distance. Specifically, let $\varDelta (> 0)$ be a pre-determined hyperparameter ($\varDelta \in \{0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.9\}$). Accordingly, the noise is then introduced to follow a uniform distribution on $[-\varDelta, \varDelta]$ or applying a dropout rate as $\varDelta$. The comparison is shown in Table 5, while the performance from the noise and dropout perturbation is clearly worse than that of MLP (92.4). The reason could be the structural change as a perturbation introduced by the MLP is stronger than that of noise and dropout, and therefore better in robustifying the encoder $\mathcal{F}$ (to tolerate more variation).

Table 5. Effect of different EP implementations including the noise addition and dropout mask.

$\varDelta$	0.1	0.3	0.5	0.7	0.9
Noise	91.1	91.4	91.4	91.2	91.3
Dropout	91.0	91.1	91.3	91.2	90.8

On the other hand, as discussed in the *Analysis* section, the choice of α (from Eq. (10)) is not critical. Herein we also evaluate the impact from α empirically. The result from Table 6 illustrates that PFA is insensitive to α, as a stable performance (with *std.* of ±0.15) is achieved.

Table 6. Effect of α

α	0.1	0.3	0.5	0.7	0.9
F1	92.3	92.4	92.4	92.3	92.7

On the bottleneck layer. The EP module implements a bottleneck structure of MLP

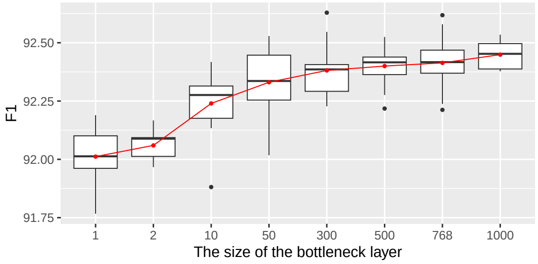


Fig. 3. Boxplot of F1 performance as the function of the size of the bottleneck layer in the EP module. The red curve shows the trend of the mean F1.

(as shown in Eq. (3)) to avoid the trivial solution (*i.e.*, an identity mapping), although it is very unlikely to happen in the optimization process (due to complex loss function landscapes in terms of model parameters). To investigate the impact from the structure, specifically, we vary the size of the hidden (or bottleneck) layer, in {1, 2, 10, 50, 300, 500, 768, 1000}, and the corresponding result, using the SQuAD dataset, is reported in Fig. 3. Clearly, we observed the larger size leads to the better performance in the statistical sense. Even with 1 hidden neuron, our method still achieves an averaged 91.95 F1 score (better than 90.30 of Base). We conjecture that more neurons increase the MLP modeling power, which may in turn give stronger perturbation and hence further improves the encoder's tolerance to the embedding variation.

On the SA module. Additionally, the SA's effect is also studied while fixing EP as the MLP. In other words, we simply replace the W_1 distance with the instance-level alignment (*i.e.*, *cos*ine similarity) and the distribution level of the Kullback-Leibler (KL) and Jensen-Shannon (JS) divergence, respectively; all other hyperparameters and structures are unchanged, although they might be suboptimal for those variants.

Table 7. Effect of the SA module with different implementations, including the instance-level (*cos*) and the distribution-level (KL, JS, and W_1) alignment.

	cos	KL	JS	W_1
SQuAD	92.2	92.4	92.3	92.4
HotpotQA	78.6	79.1	78.8	79.3
NewsQA	73.0	73.4	73.5	73.3
NaturalQ	82.1	82.2	82.2	82.2

Comparisons are summarized in Table 7, and *cos* is associated with a lowest result, suggesting the rigidness of aligning from the instance level that slightly compromises the performance. Yet, the averaged performance from *cos* is still notably higher than that of Base (using the vanilla RoBERTa-base model). Other than that, both the KL and JS divergence lead to competitive performances as W_1, which empirically indicates the flexibility of the proposed SA module.

On the low-resource fine-tuning. The following experiment validates PFA with the low-resource setting, as perturbed embeddings also play a role in providing additional (training) data. Accordingly, only a small amount (say k) of (randomly-selected) training samples are utilized for fine-tuning PFA, where $k = \{20\%, 40\%, 60\%, 80\%, 100\%\}$ (100% represents the full dataset).

Fig 4 shows the averaged F1-performance obtained with different percentages of training samples for the SQuAD dataset. Compared to the Base and SWEP (the strongest baseline from Table 2), PFA significantly improves the model performance with all percentages of the training samples. For instance, with only 40% of labeled data, PFA has achieved even higher accuracy than SWEP with 80% samples. Empirically, the result demonstrates the superiority of the proposed PFA on training robust encoders with the presence of perturbation samples; accordingly, the latent representation is strengthened thanks to the robustness of the encoder, and in turn enhances the downstream performance.

On out-of-domain generalizability. At last, PFA is evaluated via the model generalizability. Specifically, following SSMBA and SWEP, the model is first trained on a single source dataset (SQuAD in this case), and further evaluated on unseen datasets (*i.e.*, HotpotQA, NewsQA, and NaturalQ) without further fine-tuning.

The averaged F1 scores obtained using the proposed and existing methods is presented in Fig. 5. Clearly, a direct application of the Base model to downstream dataset (the vanilla RoBERTa-base model trained from SQuAD) achieves the worst F1 (60.1 on average), which indicates the distribution difference from diverse datasets and the low model generalizability (from SQuAD to others). Additionally, SWEP is observed with a better F1 performance (61.3), while PFA scores the best performance (63.5) on average across three target datasets. Due to perturbed embeddings, PFA enforces the encoder to produce more perturbation-invariant (in comparison with original token embeddings) representation, which significantly contributes to the model flexibility and generalizability. The result further indicates that PFA offers a robust starting point for fine-tuning downstream tasks, that can also be regarded as a supplementary pre-training strategy.

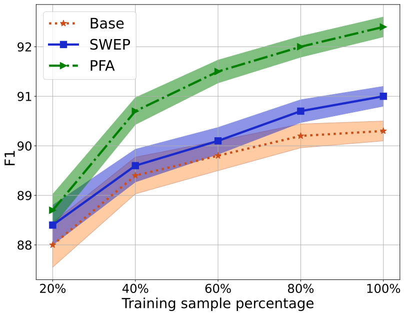


Fig. 4. Averaged F1-accuracy as a function of the training sample size.

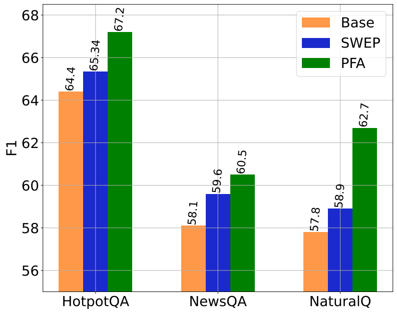


Fig. 5. Comparative F1-accuracy for the out-of-domain generalizability.

4.4 Qualitative study

We further investigate the model characteristic via visualizing the iterative loss evolution and the formed embedding/representation space.

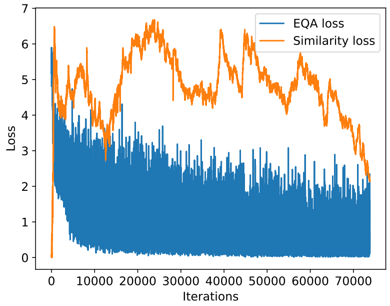


Fig. 6. Loss visualization as the function of the training iteration. The orange and blue curves are alignment and EQA loss respectively.

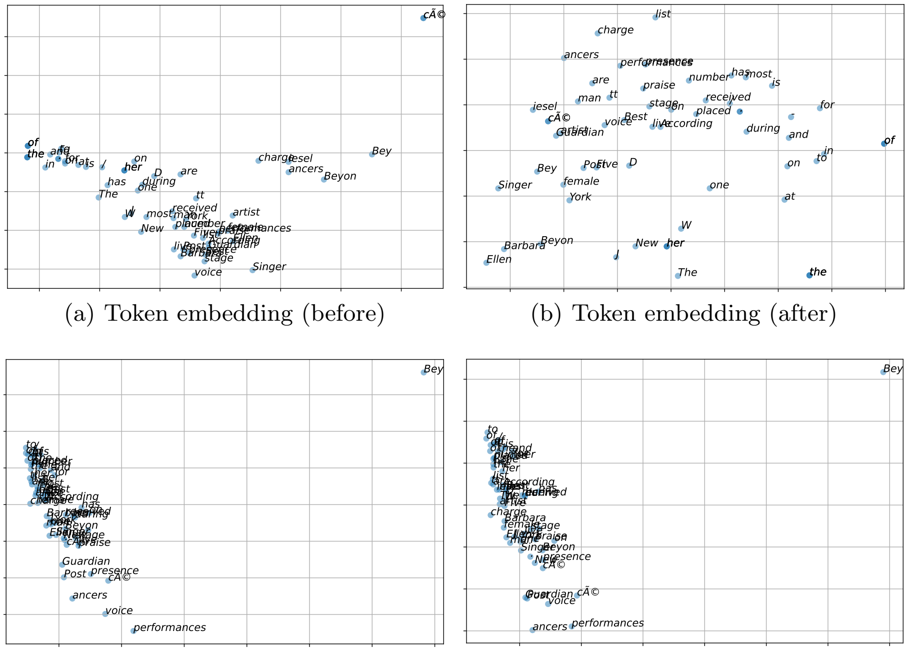
Loss visualization. Fig. 6 illustrates the iterative training loss obtained from the EQA loss ($i.e.\ \mathcal{L}_{\mathcal{S}} + \mathcal{L}_{\tilde{\mathcal{S}}}$ from Eq. (6)) against the semantic alignment (Wasserstein) loss ($i.e.\ \mathcal{L}^{d}_{ali}$ from Eq. (6)), for the SQuAD dataset. The learning curve shows that the EQA and alignment loss are well balanced, which justifies the simple choice of the weights for all loss components (currently all 1 for three losses). On the other hand, one can also add weights during the loss formulation, for example, $\mathcal{L} = \mathcal{L}_{\mathcal{S}(\boldsymbol{X})} + \lambda_1 \mathcal{L}_{\tilde{\mathcal{S}}(\boldsymbol{X})} + \lambda_2 \mathcal{L}^{i/d}_{ali}$, to allow more control. However, a detailed study on the choice of λ_1 and λ_2 is required and we leave it as future research.

Token embedding and representation visualization. We further visualize the latent space formed by the token embedding and subsequent representation, shown in Fig. 7. This 2D visualization is performed using the PCA to reduce the token embedding and representation dimension and visualize one passage sample from the SQuAD dataset.

Clearly, the perturbation projects the initial token embeddings (Fig. 7(a)) to different locations (Fig. 7(b)). This comparison evidences the proposed embedding perturbation module via producing embedding variations. On the other hand, relevant contextualized representations obtained with/out the perturbation are still similar, shown in Fig. 7(c) and (d) respectively. That demonstrates the encoder robustness of being tolerant against embedding variations and generating similar representations, from the semantic alignment module.

5 Conclusion

This paper investigates the task of Extractive Question Answering (EQA), for which a span of passage tokens are identified as answers to the given question. Existing methods mainly focus on the contextual-aware representation, while the impact from input token embeddings is underexplored. In this paper, a Perturbation for Aliment (PFA) framework is introduced to bridge this gap. Concretely, an embedding perturbation module utilizes a general transformation to produce embedding variations, while a semantic aliment module ensures the representation similarity between the original and perturbed embeddings. The goal is to strengthen the encoder so that the generated representation is stable against the embedding variation and hence beneficial to subsequent EQA tasks. This framework is also versatile due to many interpretations in terms of the semantic alignment unified under the Wasserstein-distance dual form. Intensive experiments based on four benchmarking datasets are conducted, and PFA obtains notable improvements compared to state-of-the-arts. To our knowledge, this is the first work that explores the token-embedding variation and its impact on



(a) Token embedding (before)

(b) Token embedding (after)

(c) Token representation (before)

(d) Token representation (after)

Fig. 7. The 2D PCA visualization of token embedding and representation before and after the EP and SA module, using one SQuAD example (the first 70 tokens) with the ID of e5348d10-cd31-11ed-b387-97a8ff18f3ed.

EQA, while existing work focus on the token representation aspect. We will continue exploring this idea for other downstream tasks as our future work.

References

1. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. In: Burges, C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K. (eds.) Advances in Neural Information Processing Systems. vol. 26 (2013)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
3. Fisch, A., Talmor, A., Jia, R., Seo, M., Choi, E., Chen, D.: MRQA 2019 Shared Task: Evaluating generalization in reading comprehension. In: Proceedings of the 2nd Workshop on Machine Reading for Question Answering. pp. 1–13. Association for Computational Linguistics, Hong Kong, China (Nov 2019)
4. Gehring, J., Miao, Y., Metze, F., Waibel, A.: Extracting deep bottleneck features using stacked auto-encoders. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 3377–3381 (2013)
5. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org
6. Joshi, M., Chen, D., Liu, Y., Weld, D.S., Zettlemoyer, L., Levy, O.: SpanBERT: Improving pre-training by representing and predicting spans. vol. 8, pp. 64–77. MIT Press, Cambridge, MA (2020)
7. Kang, M., Baek, J., Hwang, S.J.: 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. pp. 5144–5167. Association for Computational Linguistics, Seattle, United States (Jul 2022). https://doi.org/10.18653/v1/2022.naacl-main.379, https://aclanthology.org/2022.naacl-main.379
8. Lee, S., Kang, M., Lee, J., Hwang, S.J.: 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). pp. 5583–5595. Association for Computational Linguistics, Online (Aug 2021)
9. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized BERT pretraining approach. vol. abs/1907.11692. ArXiv preprint (2019)
10. Luo, D., Zhang, P., Ma, L., Zhu, X., Zhou, M., Liang, Q., Wang, B., Wang, L.: Evidence augment for multiple-choice machine reading comprehension by weak supervision. In: Artificial Neural Networks and Machine Learning – ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part V. p. 357–368. Springer-Verlag, Berlin, Heidelberg (2021)
11. Ng, N., Cho, K., Ghassemi, M.: 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). pp. 1268–1283. Association for Computational Linguistics, Online (Nov 2020)

12. Seonwoo, Y., Kim, J.H., Ha, J.W., Oh, A.: Context-aware answer extraction in question answering. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 2418–2428. Association for Computational Linguistics, Online (Nov 2020)
13. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**(56), 1929–1958 (2014)
14. Sun, K., Yu, D., Yu, D., Cardie, C.: 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). pp. 2633–2643. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
15. Villani, C.: Optimal Transport Old and New. Grundlehren der mathematischen Wissenschaften, 338, Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
16. Wang, R., Tang, D., Duan, N., Wei, Z., Huang, X., Ji, J., Cao, G., Jiang, D., Zhou, M.: K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 1405–1418. Association for Computational Linguistics, Online (Aug 2021)
17. Wei, J., Zou, K.: 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). pp. 6382–6388. Association for Computational Linguistics, Hong Kong, China (Nov 2019)
18. Zhang, C., Luo, C., Lu, J., Liu, A., Bai, B., Bai, K., Xu, Z.: 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. p. 1945–1948. SIGIR '20, Association for Computing Machinery, New York, NY, USA (2020)