AutoEQA: Auto-Encoding Questions for Extractive Question Answering

Stalin Varanasi Saadullah Amin Günter Neumann

Saarland Informatics Campus, D3.2, Saarland University, Germany
Geman Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
{stalin.varanasi,saadullah.amin,guenter.neumann}@dfki.de

Abstract

There has been a significant progress in the field of extractive question answering (EQA) in the recent years. However, most of them rely on annotations of answer-spans in the corresponding passages. In this work, we address the problem of EQA when no annotations are present for the answer span, i.e., when the dataset contains only questions and corresponding passages. Our method is based on auto-encoding of the question that performs a question answering (QA) task during encoding and a question generation (QG) task during decoding. Our method performs well in a zero-shot setting and can provide an additional loss that boosts performance for EQA.

1 Introduction

Extractive question answering (EQA) is the task of finding an answer span to a question from a context paragraph. Most of the deep learning models for this task perform well when annotated data is present. Scaling such models to new domains often requires creation of new datasets (d'Hoffschmidt et al., 2020; Lim et al., 2019; Trischler et al., 2017; Kwiatkowski et al., 2019). However, collecting labels for these corpora is expensive and time consuming which may involve multiple steps such as article curation, question and answer sourcing. Alleviating the annotation efforts for any of these steps is not only of research but also of practical interest. In this work, we address the problem of extracting answer spans to a question from unannotated context paragraph.

Some works have already been proposed to solve EQA in both *semi-supervised* and *unsupervised* setting. Unsupervised methods focus on creating a synthetic corpus and further train a supervised model on the synthetic corpus (Lewis et al., 2019). In semi-supervised methods the focus is on different pre-training tasks that improve the initialization of the EQA models (Dhingra et al., 2018; Glass et al., 2020; Ram et al., 2021). Our work can be categorized as the latter with one key difference: to further perform question answering without annotations on answer spans. To validate our approach, we use the pre-trained BERT (Devlin et al., 2019) model using SQuAD (Rajpurkar et al., 2016).

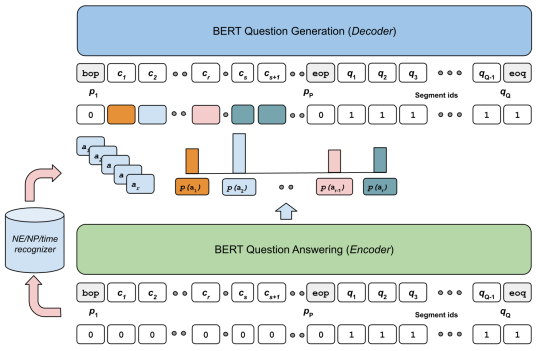

Figure 1: Schematic diagram of the proposed auto-encoding scheme. To the right, is the semi-diagonal mask on the self-attention layers for the decoding step. It enables the uni-directional language model of the question. We assume a latent distribution over *possible answer spans*, approximated by candidate phrases. See §2 for details.

Specifically, our method employs a conditional auto-encoding scheme that reconstructs question given a passage while assuming a latent distribution over the answer phrases. The encoder of our model is a Question Answering (QA) model that jointly encodes the context and the question to estimate the probability distribution over *possible* answer spans. This is further given as input along with passage to the decoder which is a Question Generation (QG) model. We use a shared architecture for both the encoder and the decoder. Therefore, our model can be viewed as a *self-supervised* machine comprehension model that learns from itself. We list our contributions as follows:

- We propose a novel method to perform unsupervised answer span extraction given a corpus of questions and associated paragraphs.

- We obtain an accuracy of 90% on unsupervised answer sentence selection.
- We obtain strong results (34.3 EM, 53.4 F1 on SQuAD dev set) for EQA when there is no annotation on the answer spans (Rajpurkar et al., 2016).

2 Method

Our model can be characterized as a discrete conditional variational auto-encoder (CVAE), where we seek to maximize the ground truth distribution of question given context $p_\theta(Q|c)$ with the assumption that there exist a latent variable *answer span*. We can then maximize the log-likelihood of $p_\theta(Q|c)$ with this assumption by the Evidence Lower Bound (ELBO) (Kingma and Welling, 2013):

$$\log p_\theta(Q|c) \geq \mathbb{E}_{a\sim q_\phi(a|c,Q)}[\log p_\theta(Q|a,c)] - D_{KL}[q_\phi(a|Q,c)||p(a|c)] \quad (1)$$

where Q is the question, c is the context, q_ϕ is the inference network, which estimates the probability of an answer a given the question and context, and p_θ is the decoder model to estimate the distribution $p_\theta(Q|a,c)$. In our case, since the architecture is shared, θ and ϕ represent the same set of parameters. Our auto-encoding scheme consists of three modules phrase extractor, encoder and decoder as shown in Figure 1.

2.1 Phrase Extractor

For EQA, given that there is no supervised signal for answer spans, an exhaustive search over all the possible phrases would be sub-optimal as there can be many phrases not suitable for natural language questions (Trischler et al., 2017; Joshi et al., 2017). We limit our potential answer phrases to the named entities and tags from constituency trees [1]. We also allow overlapping answer phrases in the set of candidate answer phrases. This is necessary as the sub-phrases of a phrase can be answers to different questions. We further remove the phrases that overlapped with the question, because such phrases can be more significant for generating the question over the possible answer phrases. With our chosen phrases, it is possible to achieve a best 70% EM and 88% F1 on SQuAD. These results serve as upper bound on our model's performance.

[1]We used https://github.com/allenai/allennlp for constituency parsing and spaCy (Honnibal et al., 2020) for NER to choose our answer candidates

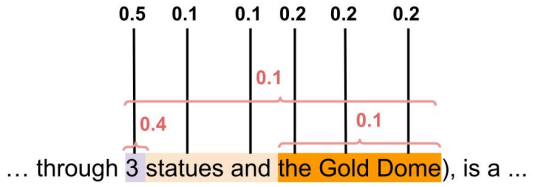

Figure 2: Example on how token scores are obtained from probabilities of overlapping phrases *3*, *the Gold Dome* and *3 statues and the Gold Dome*

2.2 Encoder

Our encoder is a pre-trained BERT (Devlin et al., 2019) model, which is referred to as the inference network, that estimates $q(a|Q,c)$ taking a paragraph concatenated with the corresponding question as input. This is similar to Devlin et al. (2019) while encoding two different text segments. Each token of the input is accompanied by a segment feature that takes values 0 or 1 representing different segments of the input (i.e., the question or the paragraph). Without a supervised signal, estimating probabilities on individual phrases might be difficult, so we decompose the probability of a phrase by using the probability of its sentence as follows:

$$q(a_{s_i}|Q,c) = q(a_{s_i}|s_i,Q,c)q(s_i|Q,c) \quad (2)$$

where s_i is the i-th sentence, a_{s_i} is one of the candidate phrases in it, Q and c are the question and the context paragraph respectively. To obtain the terms of the above expression, we define a scoring function that takes two text segments as input and outputs an affinity score. A text segment can either be a sentence, a question or a phrase. Each text segment is embedded as a vector from BERT output embeddings as follows:

$$\mathbf{v}_t = \frac{1}{|t|}\sum_{w_i \in t} \text{BERT}(w_i)$$

$$\text{score}(s,t) = \mathbf{v}_s^T \mathbf{W} \mathbf{v}_t \quad (3)$$

where t represents a text segment, $\text{BERT}(w_i)$ is the output embedding of BERT model for token w_i, $\mathbf{v}_t$ is the vector representation of the phrase t obtained as an average of BERT embeddings of the phrase tokens. The affinity score is obtained as a bilinear product of the vector representations of the text segments with learnable matrix $\mathbf{W} \in \mathbb{R}^{d\times d}$.

The conditional probability of a sentence given the question and the paragraph, $q(s_i|Q,c)$, is obtained as a *softmax* of the scoring function in Eq. 3 over all the sentences.

$$q(s_i|Q,c) = \frac{\exp(\text{score}(s_i,Q))}{\sum_{\forall s_k \in c} \exp(\text{score}(s_k,Q))}$$

Similarly, $q(a_{s_i}^{(j)}|s_i,Q,c)$ is obtained as a *softmax* of the scores between the question Q and the answer phrase $a_{s_i}^{(j)}$ over all answer phrases within the i-th sentence s_i of c:

$$q(a_{s_i}^{(j)}|s_i,Q,c) = \frac{\exp(\text{score}(a_{s_i}^{(j)},Q))}{\sum_{\forall a_{s_i}^{(k)} \in s_i} \exp(\text{score}(a_{s_i}^{(k)},Q))}$$

With these two expressions, one can obtain the probability distribution of the phrases from Eq. 2. Now, we have probabilities over different (*overlapping*) phrases and we transfer these phrase-level probabilities into token-level scores to obtain a real valued segment feature vector as follows (shown in Figure 2):

$$t_i = \sum_{\forall a_i \in s_j; t_i \in a_i} q(a_i|s_j,Q,c)$$

The purpose of the binary segment features is to differentiate some part of the text from the rest and to signify connection between them. The pre-trained weights of BERT model include segment embeddings for input segment features 0 or 1. However, the output of the encoder model is a vector of real numbers $\in [0,1]$. To accommodate this input whilst not loosing the well-informed weights of BERT, we obtain the segment embeddings for each token as an interpolation between the binary segment embeddings of BERT:

$$\text{vec}_{\text{seg}}(t_i) = \text{vec}_{\text{seg}}(0)t_i + \text{vec}_{\text{seg}}(1)(1-t_i)$$

where $\text{vec}_{\text{seg}}(t_i)$ is the segment embedding at position i, given a segment feature $t_i \in [0,1]$, $\text{vec}_{\text{seg}}(0)$ and $\text{vec}_{\text{seg}}(1)$ are segment embeddings for the input segment features 0 and 1 respectively.

2.3 Decoder

The decoder is a BERT model, which shares weights with the encoder. It performs the task of generating question given paragraph and the answer span. Here we employ a unified transformer architecture model similar to (Dong et al., 2019; Varanasi et al., 2020; Chan and Fan, 2019).

Model	Top-1
SUPERVISED	
Selector (Min et al., 2018)	91.2
BR-MPGE-AS$_{\text{Base}}$ (Tian et al., 2020)	**92.1**
UNSUPERVISED	
SBERT (Reimers and Gurevych, 2019)	63.5
TF-IDF (Min et al., 2018)	**81.2**
AutoEQA-GS$_{\text{Base}}$	75.0
UNSUPERVISED ANSWER SPAN	
AutoEQA-QG$_{\text{Base}}$	87.6
AutoEQA-QG$_{\text{Large}}$	**90.3**

Table 1: Answer sentence accuracy at top-1 sentence selection on SQuAD dev set (v1.1) (Rajpurkar et al., 2016) at different levels of supervision. Base and Large refers to *bert-base* and *bert-large* (Devlin et al., 2019) models respectively.

To encode *answer span*, we use segment features of BERT. The first term in Eq. 1 is an expectation over an estimated distribution of the inference network. This requires sampling which can be simulated by adding Gumbel-noise (Maddison et al., 2016; Jang et al., 2016) to the distribution and further taking the *softmax* with a scaling factor τ, which decides the peakiness of the distribution. However during training, we allow soft answer selection instead of choosing a single answer. The probabilities on the answer phrases are transferred as scores per token and these scores are provided as *soft* segment ids for corresponding tokens. Similar to Sun et al. (2018) and Dong et al. (2019), we use a question generation model to decode the question given a paragraph and an answer phrase as input. We hypothesize that the tasks of encoder and decoder complement each other as one single transformer model perform both QA and QG simultaneously. We use BERT based copy-mechanism (Gu et al., 2016) while generating the question as proposed by Varanasi et al. (2020). The copy-mechanism interpolates the probability distribution over the vocabulary with the probability distribution over the paragraph which is obtained from self attention scores across different layers of BERT.

3 Experiments

For EQA experiments, we used the SQuAD v1.1 (Rajpurkar et al., 2016) dataset and conducted both *sentence level* and *phrase level* answer span selection. We trained on paragraph-question pairs without using the labels for answers (i.e., $87,594$

Model	EM	F1
BASELINE		
Random (Rajpurkar et al., 2016)	1.3	4.3
Sliding Window (Rajpurkar et al., 2016)	13.0	20.0
Context Only (Kaushik and Lipton, 2018)	10.9	14.8
ANSWER SPAN SELECTION VIA PRE-TRAINING		
Cloze Corpus + BIDAF+SA (Dhingra et al., 2018)$^{\gamma}$	10.0	15.0
Cloze Corpus$_{\text{Large}}$ (Dhingra et al., 2018)$^{\gamma}$	28.0	35.8
Span Pre-train$_{\text{Base}}$ (Glass et al., 2020)*	3.8	10.4
Span Pre-train$_{\text{Large}}$ (Glass et al., 2020)*	10.9	23.2
ANSWER SPAN SELECTION VIA AUTO-ENCODING QUESTION		
AutoEQA-QG$_{\text{Base}}$	32.59	49.4
AutoEQA-QG$_{\text{Large}}$	**34.3**	**53.4**
SUPERVISED		
BERT$_{\text{Base}}$ (Devlin et al., 2018)	80.8	88.5
BERT$_{\text{Large}}$ (Devlin et al., 2018)	**84.1**	**90.9**

Table 2: Comparison of different unsupervised and semi-supervised models on SQuAD dev set. γ is implemented and reported by Lewis et al. (2019), * are the models provided by the authors.

paragraph-question pairs). We maximize the objective for log-likehood (first term in Eq. 1) where we trained for 3 epochs on the training set and kept the model that has the best log-likelihood of the question. We observed that using KL-Divergence term in the Eq. 1 causes posterior collapse rather quickly. So we limit the weight to 0.1 while using simulated annealing for some iterations after 3rd epoch. As mentioned above, removing phrases that are common with the *question* helped to avoid local minima. We used *bert-base-cased* and *bert-large-cased* (Devlin et al., 2018) models in our experiments, with initial learning rate $3e^{-5}$ using Adam (Kingma and Ba, 2015) optimizer with 0.1 proportion of linear warm-up for learning rate.

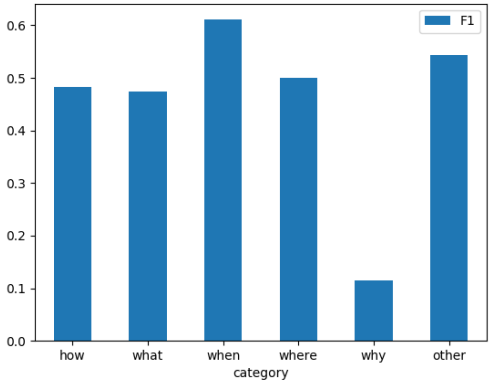

Figure 3: Average F1 scores for different question types

3.1 Unsupervised Sentence Level QA

Answer sentence selection is an important task that benefits EQA further in terms of the accuracy and speed. Min et al. (2018) showed that by reducing the context to a sentence, one can not only reduce the training and inference time but also at times obtain better accuracy. As we factored the probability of a sentence into the probability of a candidate answer phrase that it contains, our model naturally scores a sentence high if it impacts the likelihood of the question. We used a modified version of SQuAD for answer-sentence span selection, similar to Tian et al. (2020).[2] Table 1 provides a comparison of our results on SQuAD dev set to some of the unsupervised and supervised methods on answer sentence selection task. We provide our own baseline, AutoEQA-GS$_{\text{Base}}$, by auto-encoding a *missing* (*gap*) sentence from a SQuAD paragraph instead of the question. We achieve 75% accuracy on top-1 sentence. This suggests that the architecture of AutoEQA by design captures semantic similarity necessary for question-answering.

TF-IDF (Min et al., 2018) uses word frequency in the question and the sentence to provide a similarity score. Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) is a state-of-the-art sentence embedding model which is trained for Textual Sim-

[2]We used spaCy for marking sentences

ilarity tasks (STS). It is noteworthy that our model AutoEQA-GS$_{\text{Base}}$ surpasses SBERT when there is no supervision for both paragraph or answer span. For supervised sentence selection models, Min et al. (2018) uses sentence-aware question embeddings to find similarity between sentences and questions and Tian et al. (2020) uses multi-perspective graph encoding to capture sentence relations to further benefit answer-sentence selection task. While both of these models use supervision with elaborate architecture for answer sentence selection, they only marginally outperform AutoEQA-QG model in *span* unsupervised setting. This suggests the potential for AutoEQA-QG loss to enhance for sentence level EQA models.

3.2 Unsupervised Extractive Question Answering

For evaluation on answer phrases, we compare our model with other possible *answer span* selection techniques. The baseline models use heuristics to train on simple features that do not require annotation for EQA. The first baseline model is the sliding window approach reported by Rajpurkar et al. (2016) that finds answers using word overlap with the question. Secondly, they also propose a supervised logistic regression model which is trained on hand crafted features. Kaushik and Lipton (2018) use supervision to extract the most likely answer span from the context but they completely ignore the question. These models mark the baseline.

Secondly, we report models that pre-train on *answer span* selection methods to improve EQA. Dhingra et al. (2018) creates a noisy corpus from Wikipedia articles where questions are sentences with missing phrases called *cloze* questions. Recently, Glass et al. (2020) created a similar *cloze* question corpus with documents retrieved per each cloze question using information retrieval methods. Both models train on *answer span* selection that is required for the task of EQA. From table 2, one can see that AutoEQA out performs them with large margin. The difference between EM and F1 scores for our models suggests that there are more overlaps between the model's predictions and ground truth though it does not predict the exact phrase. This provides a scope of improvement on phrase selection.

While the selection of candidate answer phrases themselves can limit AutoEQA, some answer phrases might be inherently difficult to learn. For better understanding, we look at the performance statistics on different question categories. Figure 3 shows the average F1 scores on different question types. AutoEQA naturally performs better in the question categories *when*, *where*, and *what* attributing to the fact that the answers for these questions tend to be *named entities*. The model performed poorly in the *why* questions. This could be because of their lengthy answer phrases. It is interesting to note that (Lewis et al., 2019) too performed badly in this category. The category *other* refers to *which* and *who* questions combined with *no-question* word questions. Overall, we seem to see a correlation with answer types being Named Entities and the model's performance. Nearly 75% of the predicted answers are less than 10 words distant from the ground truth.

4 Related Works

Recently, data augmentation has become a popular way to do unsupervised EQA (Lewis et al., 2019; Li et al., 2020; Fabbri et al., 2020), where synthetic questions are generated either by heuristics or by unsupervised question generation methods. Brown et al. (2020) show that very large-scale language models can generate answers without supervision. While these works have their own benefits, they are different from the problem we intend to address and hence can't be compared directly. For example, Lewis et al. (2019) achieves similar performance to ours using millions of artificially created data points for EQA corpora while we achieve our results by using only 87k training samples suggesting the efficiency of our method when supervision for *question, paragraph* pairs is provided.

5 Conclusion

In this work, we propose a novel method for Unsupervised answer span selection. We showed that using auto-encoding of question, one can get considerable gains (34.3% EM and 53.4% F1 score). Methods for unsupervised key phrase extraction can benefit AutoEQA in choosing well-informed and dynamic phrases.

Acknowledgments

The work was partially funded by the German Federal Ministry of Education and Research (BMBF) through the projects CoRA4NLP (01IW20010) and XAINES (01IW20005). The authors thank Anna Vechkaeva for helpful discussions.

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*.

Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent BERT-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

Bhuwan Dhingra, Danish Pruthi, and Dheeraj Rajagopal. 2018. Simple and effective semi-supervised question answering. *arXiv preprint arXiv:1804.00720*.

Martin d'Hoffschmidt, Maxime Vidal, Wacim Belblidia, and Tom Brendlé. 2020. Fquad: French question answering dataset. *arXiv preprint arXiv:2002.06071*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. *arXiv preprint arXiv:1905.03197*.

Alexander R Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Template-based question generation from retrieved sentences for improved unsupervised question answering. *arXiv preprint arXiv:2004.11892*.

Michael Glass, Alfio Gliozzo, Rishav Chakravarti, Anthony Ferritto, Lin Pan, GP Shrivatsa Bhargav, Dinesh Garg, and Avirup Sil. 2020. Span selection pre-training for question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2782.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Divyansh Kaushik and Zachary C Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. *arXiv preprint arXiv:1808.04926*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. *arXiv preprint arXiv:1906.04980*.

Zhongli Li, Wenhui Wang, Li Dong, Furu Wei, and Ke Xu. 2020. Harvesting and refining question-answer pairs for unsupervised qa. *arXiv preprint arXiv:2005.02925*.

Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2019. Korquad1. 0: Korean qa dataset for machine reading comprehension. *arXiv preprint arXiv:1909.07005*.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Sewon Min, Victor Zhong, Richard Socher, and Caiming Xiong. 2018. Efficient and robust question answering from minimal context over documents. *arXiv preprint arXiv:1805.08092*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3930–3939.

Zhixing Tian, Yuanzhe Zhang, Xinwei Feng, Wenbin Jiang, Yajuan Lyu, Kang Liu, and Jun Zhao. 2020. Capturing sentence relations for answer sentence selection with multi-perspective graph encoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9032–9039.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

Stalin Varanasi, Saadullah Amin, and Günter Neumann. 2020. Copybert: A unified approach to question generation with self-attention. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 25–31.