CopyBERT: A Unified Approach to Question Generation with Self-Attention

Stalin Varanasi   Saadullah Amin   Günter Neumann

German Research Center for Artificial Intelligence (DFKI)
Multilinguality and Language Technology Lab
{stalin.varanasi,saadullah.amin,guenter.neumann}@dfki.de

Abstract

Contextualized word embeddings provide better initialization for neural networks that deal with various natural language understanding (NLU) tasks including question answering (QA) and more recently, question generation (QG). Apart from providing meaningful word representations, pre-trained transformer models, such as BERT also provide self-attentions which encode syntactic information that can be probed for dependency parsing and POS-tagging. In this paper, we show that the information from self-attentions of BERT are useful for language modeling of questions conditioned on paragraph and answer phrases. To control the attention span, we use semi-diagonal mask and utilize a shared model for encoding and decoding, unlike sequence-to-sequence. We further employ copy mechanism over self-attentions to achieve state-of-the-art results for question generation on SQuAD dataset.

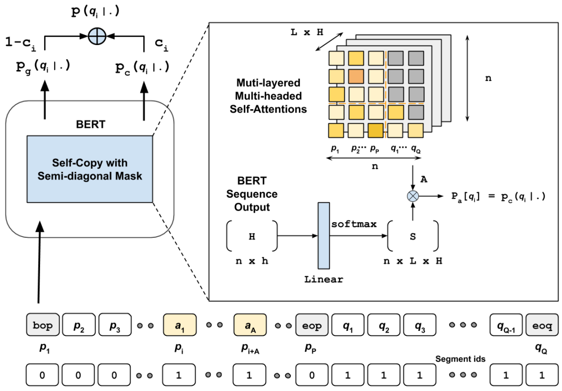
p (q_i | .)
1-c_i
c_i
p_g (q_i | .)
p_c (q_i | .)
BERT
Self-Copy with Semi-diagonal Mask
L x H
n
Muti-layered Multi-headed Self-Attentions
p_1 p_2 ··· p_P q_1 ··· q_Q
n
A
P_a [q_i] = p_c (q_i | .)
BERT Sequence Output
H
n x h
Linear
softmax
S
n x L x H
bop p_2 p_3 a_1 a_A eop q_1 q_2 q_3 q_Q-1 eoq
p_1 p_i p_i+A p_P q_Q
Segment ids
0 0 0 1 1 0 1 1 1 1 1

Figure 1: CopyBERT architecture for conditional question generation: Given a sequence of length n, with question tokens $\{q_i\}_{i=1}^{Q}$, paragraph tokens $\{p_i\}_{i=1}^{P}$ with answer phrase $\{a_i\}_{i=1}^{A}$ and semi-diagonal mask $\mathbf{M}$ (§3.2), the model explicitly uses H multi-headed self-attention matrices from L layers of transformers to create $\mathbf{A} \in \mathbb{R}^{n \times n \times L \times H}$. This matrix along with $\mathbf{S} \in \mathbb{R}^{n \times L \times H}$, obtained from the BERT sequence output $\mathbf{H} \in \mathbb{R}^{n \times h}$, is used to learn copy probability $p_c(q_i|.)$ (§3.3.2). Finally, a weighted combination $p(q_i|.)$ is obtained with simple generation probability $p_g(q_i|.)$ (§3.4).

1 Introduction

Automatic question generation (QG) is the task of generating meaningful questions from text. With more question answering (QA) datasets like SQuAD (Rajpurkar et al., 2016) that have been released recently (Trischler et al., 2016; Choi et al., 2018; Reddy et al., 2019; Yang et al., 2018), there has been an increased interest in QG, as these datasets can not only be used for creating QA models but also for QG models.

QG, similar to QA, gives an indication of machine's ability to comprehend natural language text. Both QA and QG are used by conversational agents. A QG system can be used in the creation of artificial question answering datasets which in-turn helps QA (Duan et al., 2017). It specifically can be used in conversational agents for starting a conversation or draw attention to specific information (Mostafazadeh et al., 2016). Yao et al. (2012) and Nouri et al. (2011) use QG to create and augment conversational characters. In a similar approach, Kuyten et al. (2012) creates a virtual instructor to explain clinical documents. In this paper, we propose a QG model with following contributions:

- We introduce copy mechanism for BERT-based models with a unified encoder-decoder framework for question generation. We further extend this copy mechanism using self-attentions.
- Without losing performance, we improve the speed of training BERT-based language models by choosing predictions on output embeddings that are offset by one position.

Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI, pages 25–31
July 9, 2020. ©2020 Association for Computational Linguistics

2 Related Work

Most of the QG models that use neural networks rely on a sequence-to-sequence architecture where a paragraph and an answer is encoded appropriately before decoding the question. Sun et al. (2018) uses an *answer-position* aware attention to enrich the encoded input representation. Recently, Liu et al. (2019) showed that learning to predict clue words based on answer words helps in creating a better QG system. With similar motivation, gated self-networks were used by Zhao et al. (2018) to fuse appropriate information from paragraph before generating question. More recently, self-attentions of a transformer has been shown to perform answer agnostic question generation (Scialom et al., 2019).

The pre-training task of masked language modeling for BERT (Devlin et al., 2019) and other such models (Joshi et al., 2019) make them suitable for natural language generation tasks. Wang and Cho (2019) argues that BERT can be used as a generative model. However, only few attempts have been made so far to make use of these pre-trained models for conditional language modeling. Dong et al. (2019) and Chan and Fan (2019) use a single BERT model for both encoding and decoding and achieve state-of-the-art results in QG. However, both of them use the `[MASK]` token as the input for predicting the word in place, which makes the training slower as it warranties recurrent generation (Chan and Fan, 2019) or generation with random masking (Dong et al., 2019). Both models only consider the output representations of BERT to do language modeling.

However, Jawahar et al. (2019) and Tenney et al. (2019) show that BERT learns different linguistic features at different layers. Also, Hewitt and Manning (2019) successfully probed for dependency trees from self-attention matrices of BERT. With this, we hypothesize that BERT can implicitly encode the different aspects of input for QG (Sun et al., 2018; Zhao et al., 2018) within the self-attentions across layers. As self-attention can learn soft-alignments, it can be used explicitly for copy mechanism (§3.3.2), and can yield better results (§4.3) than a model that only implicitly use self-attentions for QG (§3.3.1). Similar to Dong et al. (2019), we also employ a shared architecture for unified encoding-decoding but make an *explicit* use of self-attentions across layers, leading to similar or better results at a fraction of their training cost.

3 Model

In sequence-to-sequence learning framework, a separate encoder and a decoder model is used. Such an application to BERT will lead to high computational complexity. To alleviate this, we use a shared model for encoding and decoding (Dong et al., 2019). This not only leads to a reduced number of parameters but also allows for cross attentions between source and target words in each layer of the transformer model. While such an architecture can be used in any conditional natural language generation task, here we apply it for QG.

3.1 Question Generation

For a sequence of paragraph tokens $P = [p_1, p_2, ..., p_P]$, start and end positions of an answer phrase $s_a = (a_s, a_e)$ in the paragraph and question tokens $Q = [q_1, q_2, ..., q_Q]$ with $p_1 = \texttt{bop}$, $p_P = \texttt{eop}$ and $q_Q = \texttt{eoq}$ representing *begin of paragraph*, *end of paragraph* and *end of question* respectively, the task of question generation is to maximize the likelihood of Q given P and s_a. To this end, with m such training examples, we maximize the following objective:

$$\max_{\Theta} \sum_{j=1}^{m} \sum_{i=1}^{n} \log p(q_i^{(j)} | q_{<i}^{(j)}, P^{(j)}, s_a^{(j)})$$

where $q_{<i}$ represents previous question tokens $[q_1, q_2, ..., q_{i-1}]$. A fixed length n sequence is created by concatenating P and Q with pad tokens into $S = [P; Q]$. Similar to Devlin et al. (2019), each input token is accompanied by a segment id to differentiate between the parts of the text. The answer tokens in the paragraph and the question tokens are given segment ids 1 and the rest 0, as illustrated in Figure 1. We pass these as inputs to a pre-trained BERT-based model.

3.2 Semi-diagonal Masking

To control the information flow, we employ a semi-diagonal mask. A simple diagonal mask on the self-attentions of the transformer decoder ensures that each word only attends to the words that are seen thus far (Vaswani et al., 2017). Self-attentions of the encoder do not require such masking because the input words should inform each other while encoding. Since we use a unified encoder-decoder architecture, we ensure our masking is such that each word in the paragraph attends to all other words in the paragraph but not to any of the words in the question and each word in the question only attends to previous words in the question in addition to all the words in the paragraph. This results in a semi-diagonal mask which is also proposed by Dong et al. (2019) and shown in Figure 1.

Formally, from S in §3.1, we have $I_p = [1, 2, ..., P]$ as the sequence of paragraph indices and $I_q = [P+1, P+2, .., P+Q]$ as the sequence of question indices with $n = P + Q$ (ignoring the pad tokens). The semi-diagonal mask $\mathbf{M} \in \mathbb{R}^{n \times n}$ is defined as:

$$\mathbf{M}_{i,j} = \begin{cases} -\infty & (i \in I_p \wedge j \in I_q) \vee \\ & (i \in I_q \wedge j > i) \\ 1, & \text{else} \end{cases}$$

3.3 Copy Mechanism

Pre-trained transformer models not only yield better contextual word embeddings but also give informative self-attentions (Hewitt and Manning, 2019; Reif et al., 2019). We explicitly make use of these pre-trained self-attentions into our QG models. This also matches with our motivation to use the copy mechanism (Gu et al., 2016) for BERT, as the self-attentions can be used to obtain attention probabilities over input paragraph text which are necessary for copy-mechanism.

For the input sequence S with the semi-diagonal mask $\mathbf{M} \in \mathbb{R}^{n \times n}$ and segment ids D, we first encode with BERT$(S, \mathbf{M}, D)$ to obtain hidden representations of the sequence $\mathbf{H} = \{\mathbf{h}_i\}_{i=1}^{n} \in \mathbb{R}^{n \times h}$. We then define copy probability $p_c(y_i|.) := p_c(y_i|q_{<i}, P, s_a)$ as:

$$p_c(y_i|.) = \begin{cases} \sum_{k=1:y_i=t_k}^{P+i-1} p_a(k|y_i), & t_k \in Y \\ 0, & \text{else} \end{cases}$$

where $p_a(k|y_i) \in \mathbb{R}$ is the attention probability of copying token $t_k \in Y = \{P\} \cup \{y_j\}_{j=1}^{i-1}$ (set of all the paragraph tokens and question predictions thus far) from input position k to question position i. The distribution $p_a \in \mathbb{R}^n$ is set to zero for tokens not appearing in Y, whereas we add the corresponding attention probabilities for tokens occurring multiple times. We summarize these per position probabilities compactly in a matrix $\mathbf{P}_a \in \mathbb{R}^{n \times n}$. Now, we define several methods to obtain $\mathbf{P}_a$ with different copy mechanisms.

3.3.1 Normal Copy

First, we employ a simpler way to obtain attention probabilities, called *normal copy*:

$$\mathbf{P}_a = \text{softmax}(\mathbf{H}\mathbf{W}_n\mathbf{H}^T) \in \mathbb{R}^{n \times n}$$

where $\mathbf{W}_n \in \mathbb{R}^{h \times h}$ is a parameter matrix.

3.3.2 Self-Copy

In a transformer architecture (Vaswani et al., 2017), if there are L layers and H attention heads at each layer, there will be $M = L \times H$ self-attention matrices of size $n \times n$. For example, in BERT-Large model (Devlin et al., 2019), there would be $24 \times 16 = 384$ such matrices. Each of these self-attention matrices carry unique information. In this method for copy mechanism, called *self-copy*, we obtain $\mathbf{P}_a$ as a weighted average of all these self-attentions[1].

We obtain at each time step, a probability score for each of the M self-attention matrices in $\mathcal{A} \in n \times n \times M$ signifying their corresponding importance. Given a parameter matrix $\mathbf{W}_a \in \mathbb{R}^{h \times M}$, we obtain:

$$\mathbf{S} = \text{softmax}(\mathbf{H}\mathbf{W}_a) \in \mathbb{R}^{n \times M}$$
$$\widetilde{\mathbf{P}}_a = [\mathcal{S}_1\mathcal{A}_1^T; ...; \mathcal{S}_n\mathcal{A}_n^T] \in \mathbb{R}^{n \times 1 \times n}$$

where $\mathcal{S} \in \mathbb{R}^{n \times 1 \times M}$ is a 3D tensor with added dimension 2 to $\mathbf{S}$, $\mathcal{A}^T \in \mathbb{R}^{n \times M \times n}$ is the transposed tensor of 3D self-attention matrices $\mathcal{A}$. $\mathcal{S}_i \in \mathbb{R}^{1 \times M}$ and $\mathcal{A}_i^T \in \mathbb{R}^{M \times n}$ are the i-th slices of the tensors $\mathcal{S}$ and $\mathcal{A}^T$. The final attention probabilities $\mathbf{P}_a$ are obtained by removing the dimension 2 from $\widetilde{\mathbf{P}}_a$. Thus, the final attention probabilities are obtained as a weighted average over all self-attention matrices.

3.3.3 Two-Hop Self-Copy

A self-attention matrix as mentioned above can be considered as an adjacency matrix of a graph whose nodes are words. The probability scores represent soft edge between two words. A self-attention matrix, thus, can be considered as 1-hop attention. We would like to explore 2-hop attentions, i.e, we look for neighbouring nodes of neighbouring nodes. Note that if $\mathbf{P}_a$ is an adjacency matrix, the nodes that are connected in two hops are given by $\mathbf{P}_a^2$. Both 1-hop attentions and 2-hop attentions can be useful for copying mechanism. Let $\mathbf{P}_{\text{1-hop}} = \mathbf{P}_a$ and $\mathbf{P}_{\text{2-hop}} = \mathbf{P'}_a^2$ where $\mathbf{P'}_a$ and $\mathbf{P}_a$ are defined as mentioned in §3.3.2 with different parameters, then we define *two-hop self-copy* as follows:

$$\mathbf{P}_a(q_i) = h_i\mathbf{P}_{\text{1-hop}}(q_i) + (1 - h_i)\mathbf{P}_{\text{2-hop}}(q_i)$$

where $h_i = \sigma(\mathbf{h}_{q_i}^T\mathbf{W}_{\text{h}})$ and $\mathbf{W}_{\text{h}} \in \mathbb{R}^h$ is a parameter matrix.

[1] The semi-diagonal mask is applied to all such self-attention matrices.

3.4 Copy-Generate Probability

Once the copy probability p_c is obtained, the combined probability is obtained as weighted combination with the generation probability p_g:

$$p(q_i|.) = (1 - c_i)p_g(q_i|.) + c_i p_c(q_i|.)$$

where c_i is the likelihood to generate a token from the vocabulary or copy a token from the source and predicted tokens at position i:

$$c_i = \sigma(\mathbf{h}^T_{q_{i-1}} \mathbf{w})$$

with $\mathbf{h}_{q_{i-1}} \in \mathbb{R}^{h \times 1}$ as the hidden representation for the question token at position $i - 1$, $\mathbf{w} \in \mathbb{R}^{h \times 1}$ is a parameter vector and σ is sigmoid non-linearity. The generation probability is given by:

$$p_g(q_i|.) = \text{softmax}(\mathbf{h}^T_{q_{i-1}} \mathbf{V})$$

where $\mathbf{V} \in \mathbb{R}^{h \times |V|}$ is a parameter matrix over input vocabulary of size $|V|$.

4 Experiments

We apply the different variations of CopyBERT model as mentioned in the previous section on SQuAD v1.1 (Rajpurkar et al., 2016). For our experiments[2], we follow the training, validation and test split as used in Du et al. (2017).

4.1 Training Setup

For training, we used a batch size of 6, learning rate of $3e^{-5}$ with early stopping. The loss reaches its minimum between 2 to 3 epochs. We also trained with a batch size of 24 using gradient accumulation and found it gave similar results after the same number of optimization steps. We fixed the maximum sequence length as 384 and chose the part (document stride) of the paragraph that contained the answer phrase in case of exceeded sequence length. We decoded using beam search with a beam width of 5 and stopping at the generated token eoq. In our experiments we used `[CLS]` as bop token, `[MASK]` as eop token and `[SEP]` as eoq token.

4.2 Evaluation Metrics and Models

For evaluating our models, we report standard metrics of BLEU4, METEOR and ROUGE-L. As baselines, we take two of the non-BERT state-of-the-art models (Du and Cardie, 2018; Zhang and Bansal, 2019) and the two BERT-based QG models (Dong et al., 2019; Chan and Fan, 2019). We experimented with 4 settings: one without using any copy mechanism (No Copy), one using normal copy (Normal Copy; §3.3.1), one using self-copy (Self-Copy; §3.3.2) and finally with two-hop self-copy (Two-Hop Self-Copy; §3.3.3).

Model	BLEU4	METEOR	ROUGE-L
CorefNQG (Du and Cardie, 2018)	15.16	19.12	-
SemdriftQG (Zhang and Bansal, 2019)	18.37	22.65	6.68
Recurrent-BERT (Chan and Fan, 2019)	20.33	23.88	48.23
UniLM (Dong et al., 2019)	22.12	**25.06**	51.07
BERT + No Copy	19.37	22.49	49.12
BERT + Normal Copy	20.30	23.03	49.35
BERT + Self-Copy	21.17	23.48	49.91
BERT + Two-Hop Self-Copy	20.90	23.37	49.89
SpanBERT + Self-Copy	**22.71**	24.48	**51.60**

Table 1: Question generation results on SQuAD test split from Du et al. (2017). BERT refers to BERT-Large(cased) model (Devlin et al., 2019)

4.3 Results

Table 1 shows our results [3]. First, we note that the baseline performance of BERT-Large (cased) model with No Copy (19.37 BLEU4) is comparable with the results reported by Chan and Fan (2019) (20.33 BLEU4). We see a clear increase in performance when Normal Copy is used (20.30 BLEU4). Further, we see considerable gain in BLEU4 by using Self-Copy (+1.8 over No Copy and +0.87 over Normal Copy), supporting the hypothesis of using multi-layered, multi-headed self-attentions for copy mechanism. UniLM, which is a pre-trained model from BERT-Large checkpoint with three sequence generation pre-training tasks (Dong et al., 2019) and further fine-tuned on SQuAD dataset for 10 epochs achieves 22.12 BLUE4 score. We achieve comparable performance by only using self-copy mechanism. Figure 2 shows attention patterns of self-copy in question generation.

To further validate the self-copy mechanism, we also experimented by initializing with a variant of BERT[4] called SpanBERT (Joshi et al., 2019), which is pre-trained to predict longer masked spans to encourage better entity masking and has already shown to improve QA results when compared to BERT (Joshi et al., 2019). Although, Two-Hop Self-Copy did not improve upon the Self-Copy,

[2] The code is available at https://github.com/StalVars/CopyBERT

[3] We used the evaluation script from https://github.com/microsoft/unilm/tree/master/unilm-v1

[4] Note that Self-Copy mechanism can be applied with any BERT-like pre-trained model

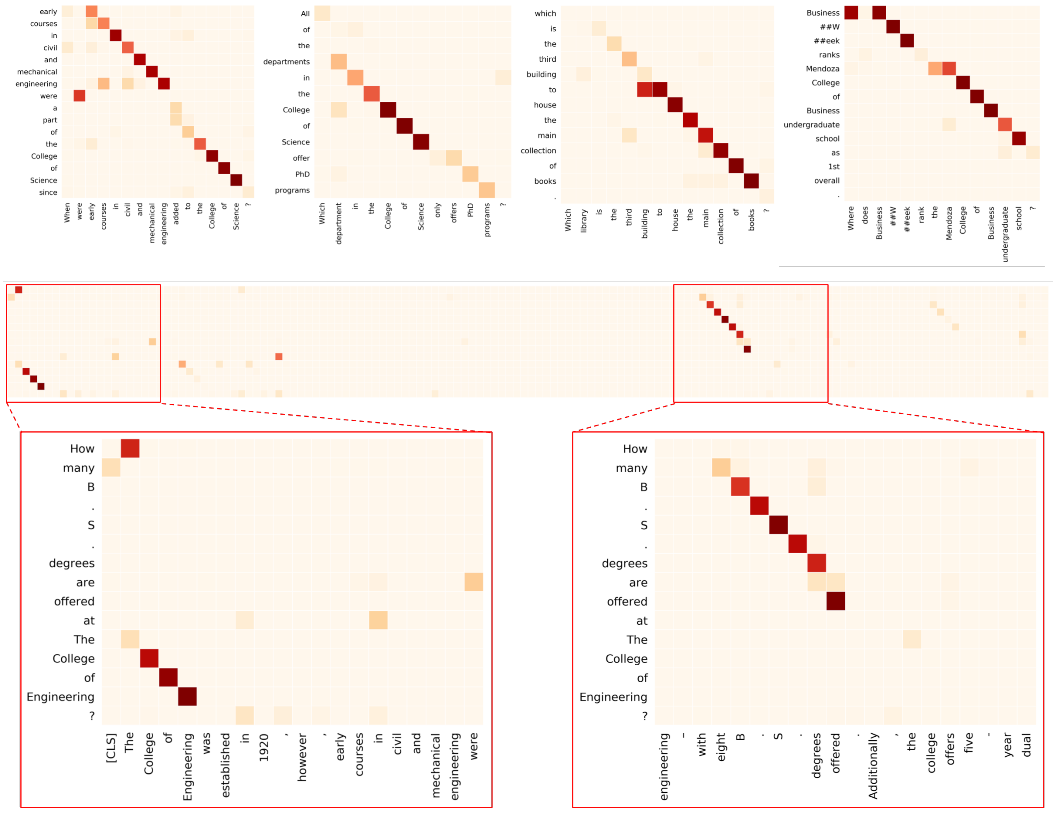

Figure 2: CopyBERT attention visualizations of copy probability on SQuAD examples. *Top*: Attention focused paragraph tokens on y-axis and generated question tokens on x-axis, where we see that the learnt copy probabilities consistently extract words from the paragraph context. *Bottom*: Long-span attention pattern over the paragraph words (x-axis), where the copy probability looks for question words (y-axis) even when most of the question words are present in the local context around the answer phrase.

these attentions can serve as explainability of QG, a good intuition behind copying different words, which we plan to explore in our future work.

4.4 Training Speed

CopyBERT trains significantly faster than UniLM. For UniLM, to fine-tune further on QG task it takes around 10 epochs to obtain its best performance. This is because the model uses input token `[MASK]` to predict a target question word and as a result can only train with some percentage of randomly chosen words to ensure that the probability is conditioned on previous question words. CopyBERT, in contrast, takes only 2 to 3 epochs to achieve its best performance. It took CopyBERT around 14 hours on a single GPU with 12GB main memory to train for 3 epochs, whereas UniLM took around 45 hours on the same hardware to train for 10 epochs to achieve similar results as reported in Dong et al. (2019). We expect Recurrent-BERT (Chan and Fan, 2019) to take even longer time to train due to its sequential nature.

5 Conclusion

We showed that having a unified encoder-decoder transformer model initialized with contextualized word embeddings and further extended with copy mechanism can already give state-of-the-art, without additional pre-training on generation tasks (Dong et al., 2019). We also sped up the training of QG models that use BERT by choosing predictions on output embeddings that are offset by one position (§3.3). This work shows the significance of explicitly using self-attentions of BERT like models. These models can further be used in other tasks such as abstractive summarization and machine translation to see qualitative improvements.

Acknowledgements

The authors would like to thank the anonymous reviewers for helpful feedback. The work was partially funded by the German Federal Ministry of Education and Research (BMBF) through the project DEEPLEE (01IW17001).

References

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.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184.

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.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13042–13054.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1907–1917.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Pascal Kuyten, Timothy Bickmore, Svetlana Stoyanchev, Paul Piwek, Helmut Prendinger, and Mitsuru Ishizuka. 2012. Fully automated generation of question-answer pairs for scripted virtual instruction. In *International Conference on Intelligent Virtual Agents*, pages 1–14. Springer.

Bang Liu, Mingjun Zhao, Di Niu, Kunfeng Lai, Yancheng He, Haojie Wei, and Yu Xu. 2019. Learning to generate questions by learning what not to generate. *arXiv preprint arXiv:1902.10418*.

Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Margaret Mitchell, Xiaodong He, and Lucy Vanderwende. 2016. Generating natural questions about an image. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1802–1813.

Elnaz Nouri, Ron Artstein, Anton Leuski, and David Traum. 2011. Augmenting conversational characters with generated question-answer pairs. In *2011 AAAI Fall Symposium Series*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of bert. In *Advances in Neural Information Processing Systems*, pages 8592–8600.

Thomas Scialom, Benjamin Piwowarski, and Jacopo Staiano. 2019. Self-attention architectures for answer-agnostic neural question generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6027–6032, Florence, Italy. Association for Computational Linguistics.

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.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. corr abs/1611.09830.

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

Alex Wang and Kyunghyun Cho. 2019. Bert has a mouth, and it must speak: Bert as a markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Xuchen Yao, Emma Tosch, Grace Chen, Elnaz Nouri, Ron Artstein, Anton Leuski, Kenji Sagae, and David Traum. 2012. Creating conversational characters using question generation tools. *Dialogue & Discourse*, 3(2):125–146.

Shiyue Zhang and Mohit Bansal. 2019. Addressing semantic drift in question generation for semi-supervised question answering. *arXiv preprint arXiv:1909.06356*.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910.