Answer-Aware Neural Question Generation with Normalized Beam Search

Spencer Rose
University of Victoria
spencerrose@uvic.ca

Abstract

This paper presents an enhanced deep sequence-to-sequence model for question generation. The model builds on the attention-based sequence to sequence architecture of Neural Question Generation (Du, et al. [3]), with two modifications to improve the quality of generated questions: (1) Answer-Aware source embedding to contextualize target questions with their answers; and (2) Normalized Beam Search decoder to improve the selection of candidate output sequences. Trained on large datasets of paired sentences and target questions with answers, experimental results show the proposed model learns to generate natural, contextually-relevant questions from sentence-level source text.

1. Introduction

Automatic Question Generation (QG) from natural language text aims to generate natural or human-like questions based on a given text input or context. QG has many application areas, including reading comprehension [8], healthcare [5], and intelligent conversational systems such as Alexa, Google Assistant and Siri. QG also forms a component of Visual Question-Answering neural networks conditioned on image features to generate relevant question [12]. Moreover, within the field of Natural Language Processing, QG presents particular challenges that makes it an important discourse processing task in its own right [16].

Past work in QG systems has relied on hand-crafted rules or heuristics for performing declarative-to-interrogative transformations of source text. Many such systems involve syntactic- or semantic-based parsing based on deep linguistic knowledge [8]. Encoding the rich and complex nature of human language has proven to be difficult: rule-based models are laborious to build, and their success depend heavily on complex, well-designed rules.

In this paper, we explore an alternative approach, Neural Question Generation, based on an evaluation and proposed enhancement of the work of Du, et al. [3].

Challenges. The primary challenge for QG is to generate natural questions strongly relevant to the source text. When we consider how a human might handle the task, we find that formulating a high quality question often requires context that is longer than one or two sentences – rather, it may require a paragraph or entire corpus. As a task for deep learning, tracking such lengthy input text typically poses severe implementation challenges.

A multi-layered Recurrent Neural Network (RNN) encoder-decoder with bidirectional Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU) is the standard architecture for encoding arbitrarily long-term dependencies. RNNs are particularly suited for long sentences where the target meaning requires the entire input. Normally, the encoder takes as input a word embedding of the source text and produces a final "sentence embedding" or context vector of fixed dimensionality. This vector encapsulates information from all input elements and gets passed to an encoder for generation. Bi-directionality for both encoder and decoder further enables the model to encode two passes of the whole source text. Still, long-range token dependencies prove problematic in practice.

Attention mechanisms, such as Bahdanau attention, [1], improve tracking by giving more weight to key tokens of the source text. Instead of a single fixed-length context vector passed from encoder to decoder, attention allows the model to selectively concentrate on a subset of the input time steps according the decoder states (by using a softmax activation function), thus improving a question's contextual relevance. This attention-based encoding is analogous to the human approach to QG in paying attention to certain parts of a sentence – i.e. aspects of the context based on the surrounding text.

A similar attention mechanism called "answer-aware" embedding [21] improves attention tracking by extending source embeddings with answer tags that mark answer tokens. This mechanism allows the model to focus on important contextual information for question formation.

On the decoder end, Beam Search is a widely used algorithm to select probable output sequences that lends a further improvement to the quality of the generated output. The algorithm uses breadth-first search to infer the best, or

most likely, token sequences from a list of generated candidates. Selection is made using the cumulative probabilities of candidates, and only the top N (beam size) nodes of questions are kept.

Contributions. The following paper proposes two main enhancements to the Du et al. [3] Neural QG model: (1) Answer Tagging (AT), a form of answer-aware attention applied to the source text [21]; and (2) Normalized Beam Search (NBS) algorithm, based on the work of Wu et al. [19], used to improve traditional beam search by normalizing beam scores by the encoder attentional weights and candidate sequence lengths.

Results show an improvement in convergence of training and validation loss for sentence-level, improved BLEU and METEOR scores when compared with baseline models. However, paragraph-level source embedding does not yield good results.

2. Contributions of the Original Paper

2.1. Neural Question Generation

In their 2017 paper "Learning to Ask" [3], Du, et al. present Neural Question Generation, the first end-to-end deep sequence-to-sequence learning approach to QG, with a particular focus on reading comprehension. Inspired by recent successes in neural machine translation, the authors employed an attention-based bidirectional LSTM encoder-decoder RNN, based on the work of Bahdanau, et al [1]. A variation of the model includes a global attention mechanism [10] to support both sentence-level and paragraph-level attention.

In contrast to previous QG systems that employed rule-based sentence decomposition, Neural QG is a fully data-driven framework that formulates question generation as a sequence-to-sequence problem. The approach furthermore requires no manually-generated rules, or complex natural language process pipeline. [1]

Using human inspection and well-established machine translation metrics (e.g. BLEU [13] and METEOR [2]), the authors' best model out-performs state-of-the-art rule-based models. However, in the authors' comparison of sentence- vs. paragraph-level attention, the models trained with paragraph-level global attention performed slightly worse than those with a strictly sentence-level attention. This performance drop is explained by the authors as the noise of extraneous or irrelevant paragraph tokens. Furthermore, the use of Glove pre-trained word embeddings boosted performance versus randomized embeddings.

Though based on a standard architecture, the results of the experiment established a proof-of-concept for neural network question generation. [2]

2.2. Dataset

Neural QG training requires very large datasets (with approximately 800 billion tokens) consisting of sentence-question pairs. Both the original and current project models were trained on the Stanford Question and Answer Dataset (SQuAD) [15] - a training corpus composed of more than 100K questions posed by crowdsourced workers on 536 Wikipedia articles.

The models also use GloVe (Global Vectors for Word Representation) [14] **glove.840B.300d** pre-trained weights for both source and target embedding layers. The embedding layer stores a lookup table to map the words represented to their dense vector representations. GloVe contains word co-occurrence probabilities which semantically encode the source and target vocabulary, which helps generate target questions semantically similar (i.e. probabilistically close in the vector space) to the source tokens.

2.3. Model Weaknesses

The primary weakness of the Neural QG model is its limited ability to control the quality and accuracy of generated questions. Unlike sequence-to-sequence learning tasks such as machine translation, which achieve a roughly one-to-one mapping of source to target sequences, Neural QG may arbitrarily focus on quite different aspects of the source text, and thus generate questions of considerable variation in quality and relevance [6]. Though Question-Answer training datasets typically have document-question pairs that specify a unique answer, valid questions can be formed from any information or relations which uniquely specifies the given answer, hence a typical document-answer pair may be associated with multiple questions [22].

During optimization of the model, non-overlapping questions may be generated by slight modifications to the inference beam size, learning rate, or other hyperparameters. Therefore, trained attentional weights may not adequately encode a determinable sentence context for QG that can be properly assessed.

Furthermore, the original model's performance drop with paragraph-level attentional weights (i.e. the global attention mechanism) presents a major limitation on the ability of the model to produce high-quality questions. The

[1] A similar neural network model was developed by Serban et al. 2016 [17] to transduce facts – represented by a triple consisting of a subject (sentence), a relationship and an object (answer) – into simple natural language questions [14]. However, this model relied on structured representations of text that mapped to natural language text, and their generated questions, whereas the Du, et al. model uses freeform natural language text to generate a wider diversity of questions.

[2] Note that a follow-up paper published in 2017 by Du et al. [4] outlined improvements to this model using a layer that identifies "question-worthy" sentences within a context paragraph, which then become input for their former model. For this, they employ an hierarchical neural sentence labeling model and applied both sum and convolution operations for the encoding.

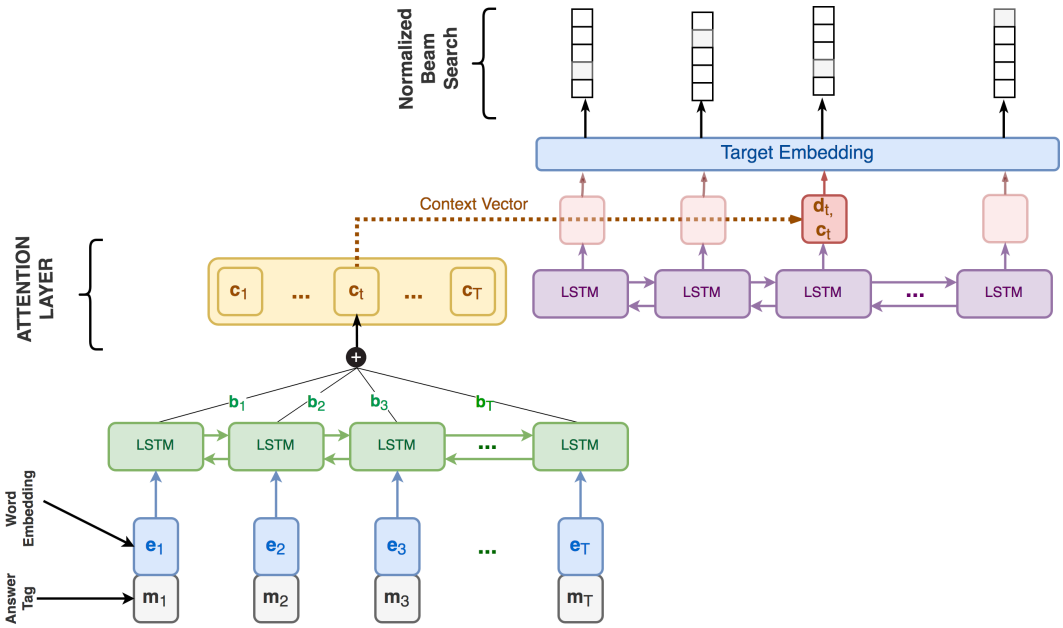

Figure 1: Question Generation RNN Encoder-Decoder Network Architecture showing Answer Tagging and Normalized Beam Search.

authors themselves note that 20% of the SQuAD dataset questions that require a context categorized at "paragraph-level", which leaves out a significant portion of the dataset. Other than human inspection of the generated question, it is also not clear how such a categorization might be implemented as an input filter during validation. Future QG systems therefore must be able to adequately encode much longer context lengths to improve the quality of generated questions.

3. Contributions of this Project

For this project, a similar model to the original Du et al. [3] architecture was implemented as a baseline to test the following model enhancements. Note that GRU was used as the core RNN unit instead of LSTM (used in the original model), as it was found the model loss (perplexity) curve of GRU-based model converged faster than with LSTM.

3.1. Answer-Aware Encoding

One approach to improving the encoding of paragraph-length source input is through "answer-aware" embedding [23], which can focus attentional weight on questions that target their given answers. Fortunately, the SQuAD dataset provides such answers and their location in the text for easy mapping.

Building on the work of Zhao et al. [23] answer tagging extends each token in the source word embedding with a tag that indicates whether or not the word in the source text is in the answer. By encoding the answer position in the encoder hidden states context vectors, the intention is to promote "answer-aware" questions. Answer tagging also helps incorporate paragraph-level source text by by providing greater weight to answer words, allowing the model to neglect noise when processing a long context.

Answer tagging can be modeled for the encoder GRU hidden state $\mathbf{h}$ at time step t with the word embedding representation of word x_t as follows,

$$\mathbf{h}_t = GRU(\mathbf{h}_{t-1}, [\mathbf{e}_t, \mathbf{m}_t]) \tag{1}$$

$[\mathbf{e}_t, \mathbf{m}_t]$ represents the concatenation of the embedded word $\mathbf{e}_t$ with the meta-word $\mathbf{m}_t$ that indicates whether the word is part of a string that is also in the answer [22] [23]).

3.2. Normalized Beam Search

In sequence to sequence models, the Beam Search algorithm is frequently used to select the best, or most probable output sequence based on the decoder's final log-softmax linear transformation. For this implementation, the model learns through "teacher forcing": the decoder reuses its output of previous time steps to predict the next token. At each time step, Beam Search then uses weighted breadth-first search to choose the top generated log-softmax probabilities. Only the top K scoring values are maintained at each step, and the final sequence of probabilities that maximize the scores is selected.

Beam Search decoders tend to generate very short sequences (e.g. questions like "What is?"), since shorter sequence lengths maximize the Beach Search scores. To correct this, two normalization penalities based on the work of

Decoder Type	Output (Paragraph-level Source Text)
Source	*despite the high position given to muslims some policies of the yuan emperors severely discriminated against them restricting halal slaughter and other islamic practices like circumcision as well as kosher butchering for jews forcing them to eat food the mongol way toward the end corruption and the persecution became so severe that muslim generals joined han chinese in rebelling against the mongols the ming founder zhu yuanzhang had muslim generals like lan yu who rebelled against the mongols and defeated them in combat some muslim communities had a chinese surname which meant barracks and could also mean thanks many hui muslims claim this is because that they played an important role in overthrowing the mongols and it was given in thanks by the han chinese for assisting them during the war fighting the mongols among the ming emperor zhu – rebellion but the rebellion was crushed and the muslims were massacred by the yuan loyalist commander chen –*
Greedy Decoder	*who was the leader of the ming dynasty?*
Normalized Beam Search	*who is leading the muslim area for to eu and and on europe?*
Target Question (SQuAD)	*who founded the ming dynasty?*

Table 1: Example Generate Questions (Answer-Aware enabled)

Wu et al. (2016) [19] were implemented: (1) Length Normalization, to normalize the sequence score by sequence length; and (2) Coverage Normalization, a penalty to encourage the model to translate all of the provided input.

The cumulative score is therefore calculated based on the log-probabilities of the output word at each time step normalized to the current target length, plus the coverage penalty:

$$s(Y,X) = \frac{\log P(Y|X)}{lp(Y)} + cp(X,Y) \tag{2}$$

The coverage normalization $cp(X,Y)$ is defined as:

$$cp(X,Y) = \beta \sum_{i=1}^{|X|} \log(\min(\sum_{j=1}^{|Y|} a_{i,j}, 1.0)) \tag{3}$$

where $a_{i,j}$ is the attention probability of the j-th target word y_j on the i-th source word x_i, $|X|$ is the source length, $|Y|$ is the current target length and β is a hyperparameter.

A greedy decoder, which selects the maximum probability of each time step (i.e. beam size of one), was used as a baseline.

3.3. Results

The proposed model was not able to duplicate the results from the original experiments by Du et al. However, a similar model architecture was adapted to establish a baseline model for testing of the proposed model modifications. [3] Both a sentence-level model (see loss curves in Figure 2) and paragraph-level model (see loss curves in Figure 3) were implemented. The baseline model was tested on sentence-level source input.

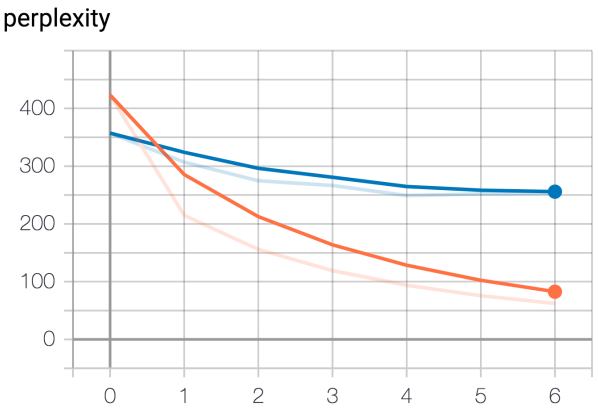

Figure 2: Sentence-Level: Training (Orange)/Validation (Blue) Loss - perplexity/epochs

Trials of the baseline sentence-level model exhibited high overfitting in (see figure 2). In the paragraph-level source, the learning rate was dropped to 0.0001, L2 regularization added, and dropout rate set to 0.5. The GRU hidden unit size was set to 512 with two layers for both encoder and decoder. Optimization is performed using stochastic gradient descent (SGD), with Adam optimizer using default settings (β1=0.9, β2=0.999 and ϵ=10-8). The mini-batch size for the update was set at 64.

$$\mathcal{L} = -\sum_{i=1}^{S} log\big[P\big(\mathbf{y}^{(i)}|\mathbf{x}^{(i)};\theta\big)\big] \tag{4}$$

[3] Pytorch implementation adapted from Alexander Rush, The Annotated Transformer, `http://nlp.seas.harvard.edu/2018/04/03/attention.html` and Joost Bastings, The Annotated Encoder-Decoder with Attention. `https://bastings.github.io/annotated_encoder_decoder/` and Vaswani et al. [18]

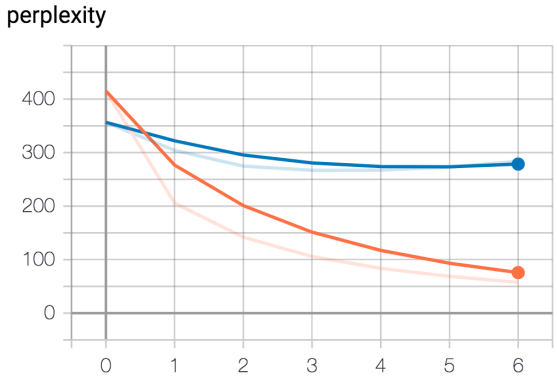

Figure 3: Paragraph-level: Training (Orange)/Validation (Blue) Loss - perplexity/epochs

Method	BLEU Score	METEOR Score
Baseline	3.44	2.11
para w/ AT	6.05	3.44
para w/o AT	1.03	4.12
sentence w/ AT	**12.43**	**17.23**
sentence w/o AT	11.19	13.30

Table 2: Average evaluation results of different models by BLEU and METEOR metrics. For a detailed explanation of the baseline system, please refer to Section 2.1. The best performing system for each column is highlighted in boldface. AT = Answer Tagging

The model was trained and tested on the Stanford Question and Answer Dataset (SQuAD) [15] and evaluated using BLEU and METEOR [2] automatic methods. BLEU, a metric developed for machine translation, has been adapted for QG to compare n-grams of the hypothesis question with the n-grams of the reference source sentence (i.e. not the entire paragraph) and count the number of matches. METEOR has similarly been adapted to score question hypotheses by aligning them to one or more reference sentences.

Example generated questions are also presented to show the quality of Normalized Beam Search versus the Greedy Decoder inferences.

4. Discussion and Future Directions

For paragraph-level training, the results show some improvement in the quality of generated questions with answer-aware embedding versus without it. However, for most examples, paragraph-level embedding produced poorer results versus sentence-level. An example generated question is provided in Table 1.

Though the proposed model shows some improvement over the model baseline, in spite of different parameter configurations, overfitting in the training was a persistent problem after around epoch 4-5. Training time averaged 2 hours for eight epochs with the default configuration, making it time-consuming to do a parameter grid search. Also, training and validation losses for the optimized model were much higher than without AT enabled.

For the Normalized Beam Search, due to its high number of parameters, hyperparameter tuning was difficult, and led to large fluctuations in output length. The resulting questions did not show a substantial improvement over the greedy decoder. In view of further study on this algorithm, recent research by Huang et al. [9] has produced a provably optimal beam search algorithm that returns the optimal-score complete hypothesis.

One significant difficultly in evaluating Neural QG accuracy and quality is that no metric exists to adequately measure the quality of the generated questions. This is because it is difficult to determine whether a particular question is good without knowing the context in which it is posed [16]. Paragraph-level questions will show significantly lower BLEU and METEOR, which rely on n-gram overlap, are furthermore not adequate, since GloVe word embeddings may replace semantically similar words in legitimate questions.

4.1. Future Directions

A potential area for further exploration in question generation is in the use of Convolutional Neural Networks (CNN) to improve encoding of the source text based on a larger – i.e. paragraph or corpus – source context. CNN and RNN have been shown to provide complementary information: while the RNN computes a weighted combination of all sentence tokens, the CNN extracts the most informative tokens for the particular source-target relation (e.g. answer-based questions) and only considers their resulting activations [21]. Yet the use of attention-based CNN in applications where text generation forms a strong component remains under-explored [20].

Previous work in attention-based CNN for machine translation suggests possible applications for question generation. As we have seen in the foregoing discussion, the quality of generated questions frequently depends on the scope of the context, while tagging relevant tokens can improve training accuracy slightly. Answer tagging at the paragraph-level attends to position-invariant features in the source context in the context vectors of the decoder. More precisely, answer tokens offer token-contiguous markers in the vector space that can occur at any position in the source text. This kind of position-invariant information is traditionally well-suited for CNN.

CNN with tagging adapted for machine translation extends the word embedding with a tagging bit (0 or 1) in the input layer to indicate whether the token is one of the tar-

Decoder Type	Output (Sentence-level Source Text)
Source	*in 1979, the soviet union deployed its 40th army into afghanistan, attempting to suppress an islamic rebellion against an allied marxist regime in the afghan civil war*
Greedy Decoder	*What was the name of the soviet union that was an alliance to the war?*
Normalized Beam Search	*What did the war in the war i to the invasion of the treaty to war?*
Target Question (SQuAD)	*Who deployed its army into afghanistan in 1979?*

Table 3: Example Generated Questions (Answer-Aware enabled)

get words [11]. This tagging bit is then activated during the training for predicting the target words, helping to pinpoint the relevant parts of the source sentences. A similar "answer tagging" mechanism may be employed for question generation adapted from the results of this project.

Attention-based CNN has shown better performance than attention-based LSTM for answer selection, where CNN is particularly effective at key-phrase matching [21] and sentence modelling [20]. The "vanishing gradient" problem of RNN can also be addressed through residual networks that gate the input embedding. Improved speed is another potential benefit of using CNN for QG. For machine translation, the work of Gehring et al [7]. 2016 has shown CNN can speed up decoding by a factor of two at the same or higher accuracy as a strong bi-directional LSTM.

5. Conclusion

This paper introduced two modifications to the Neural Question Generation model introduced by Du et al.: (1) Answer Tagging for source embedding and (2) Normalized Beam Search target decoding. These additional mechanisms were to address the original model's limited ability to both encode paragraph-level information for contextualized questions, and select likely questions weighted by the attentional weights. Experimental showed only modest improvement using AT to both BLEU and METEOR automatic metrics at sentence-level passages, as well as based on human evaluation of sample generated questions. Normalized Beam Search did not result in better quality generation over greedy decoder.

References

[1] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[2] Denkowski, Michael, and Alon Lavie. "Meteor universal: Language specific translation evaluation for any target language." In Proceedings of the ninth workshop on statistical machine translation, pp. 376-380. 2014.

[3] Du, Xinya, Junru Shao, and Claire Cardie. "Learning to ask: Neural question generation for reading comprehension." arXiv preprint arXiv:1705.00106 (2017).

[4] Du, Xinya, and Claire Cardie. "Identifying where to focus in reading comprehension for neural question generation." In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2067-2073. 2017.

[5] Feldman, Keith, Spyros Kotoulas, and Nitesh V. Chawla. "TIQS: Targeted Iterative Question Selection for Health Interventions." Journal of Healthcare Informatics Research (2018): 1-23.

[6] Gao, Yifan, Jianan Wang, Lidong Bing, Irwin King, and Michael R. Lyu. "Difficulty controllable question generation for reading comprehension." arXiv preprint arXiv:1807.03586 (2018).

[7] Gehring, Jonas, Michael Auli, David Grangier, and Yann N. Dauphin. "A convolutional encoder model for neural machine translation." arXiv preprint arXiv:1611.02344 (2016).

[8] Heilman, Michael, and Noah A. Smith. "Good question! statistical ranking for question generation." In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 609-617. Association for Computational Linguistics, 2010.

[9] Huang, Liang, Kai Zhao, and Mingbo Ma. "When to finish? optimal beam search for neural text generation (modulo beam size)." arXiv preprint arXiv:1809.00069 (2018).

[10] Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025 (2015).

[11] Meng, Fandong, Zhengdong Lu, Mingxuan Wang, Hang Li, Wenbin Jiang, and Qun Liu. "Encoding source language with convolutional neural network for machine translation." arXiv preprint arXiv:1503.01838 (2015).

[12] Misra, Ishan, Ross Girshick, Rob Fergus, Martial Hebert, Abhinav Gupta, and Laurens van der Maaten. "Learning by asking questions." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 11-20. 2018.

[13] Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "BLEU: a method for automatic evaluation of machine translation." In Proceedings of the 40th annual meeting on association for computational linguistics, pp. 311-318. Association for Computational Linguistics, 2002.

[14] Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543. 2014.

[15] Rajpurkar, Pranav, Robin Jia, and Percy Liang. "Know What You Don't Know: Unanswerable Questions for SQuAD." arXiv preprint arXiv:1806.03822 (2018).

[16] Rus, Vasile, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. "The first question generation shared task evaluation challenge." (2010).

[17] Serban, Iulian Vlad, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. "Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus." arXiv preprint arXiv:1603.06807 (2016).

[18] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In Advances in Neural Information Processing Systems, pp. 5998-6008. 2017.

[19] Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun et al. "Google's neural machine translation system: Bridging the gap between human and machine translation." arXiv preprint arXiv:1609.08144 (2016).

[20] Yin, Wenpeng, Katharina Kann, Mo Yu, and Hinrich Schütze. "Comparative study of CNN and RNN for natural language processing." arXiv preprint arXiv:1702.01923 (2017).

[21] Yin, Wenpeng, Hinrich Schütze, Bing Xiang, and Bowen Zhou. "ABCNN: Attention-based convolutional neural network for modeling sentence pairs." Transactions of the Association for Computational Linguistics 4 (2016): 259-272.

[22] Yuan, Xingdi, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Sandeep Subramanian, Saizheng Zhang, and Adam Trischler. "Machine comprehension by text-to-text neural question generation." arXiv preprint arXiv:1705.02012 (2017).

[23] Zhao, Yao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. "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, pp. 3901-3910. 2018.

[24] Zhou, Qingyu, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. "Neural question generation from text: A preliminary study." In National CCF Conference on Natural Language Processing and Chinese Computing, pp. 662-671. Springer, Cham, 2017.