# Question Generation using Seq2Seq with Attention

**Aditya Parkhi**[1] , **Ayaan Mohammed**[1]
[1]Department of Computer Science, North Carolina State University
aparkhi@ncsu.edu, amohamm4@ncsu.edu

## Abstract

We analyze automatic question generation for sentences in reading comprehension. By implementing a sequence learning model for the task, we investigate the effect of attention and beam search decoding on a trainable sequence-to-sequence learning model. Evaluation metric results show that our model performs satisfactorily on short sequences but does not generate meaningful sentences for large sequences. In human evaluations, questions generated by our system are rated as being accurate (grammaticality, correctness).

## 1 Introduction

Question generation (QG) aims to create natural questions from a given a sentence or paragraph. One key application of question generation is in the area of education which is to generate questions for reading comprehension materials. Recurrent neural networks [1] and gated recurrent [2] neural networks have been firmly established as state-of-the-art approaches in sequence modeling and machine translation. RNNs are specialized neural-based approaches that are effective at processing sequential information. An RNN recursively applies a computation to every instance of an input sequence conditioned on the previous computed results. These sequences are typically represented by a fixed-size vector of tokens which are fed sequentially (one by one) to a recurrent unit. The main strength of an RNN is the capacity to memorize the results of previous computations and use that information in the current computation. This makes RNN models suitable to model context dependencies in inputs of arbitrary length so as to create a proper composition of the input. RNNs have been used to study various NLP tasks such as machine translation, image captioning, and language modeling, among others. We conduct extensive analysis to evaluate our models in terms of learning, the ability to handle long sentences, choices of attentional architectures, alignment quality, and translation outputs.

## 2 Related Work

A lot of research has focused on first manually constructing question templates, and then applying them to generate questions (Mostow [7]). Generally, the rule-based approaches make use of the syntactic roles of words, but not their semantic roles. QGSTEC [8] deals with the methodology to generate questions focusing on person Named Entities (NE), temporal or location information, agent based semantic roles associated with the words in the input sentences. Heilman and Smith [13] introduce an overgenerate-and-rank approach: their system first overgenerates questions and then ranks them. Although they incorporate learning to rank, their system's performance still depends critically on the manually constructed generating rules. Rajpurkar [4] released the Stanford Question Answering Dataset1 (SQuAD), which is a reading comprehension dataset, consisting of questions posed by crowd workers on a set of Wikipedia articles, where the answer to every question is a segment of text from the corresponding reading passage, or the question might be unanswerable. We use SQuAD in our implementation with focus on the generation of natural questions for reading comprehension materials.

## 3 Attention-Based Model

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. We implement a local attention mechanism[5] selectively focusing on a small window of context which is differentiable. This approach has an advantage of avoiding the expensive computation incurred in the soft attention and at the same time, is easier to train than the hard attention approach. In concrete details, the model first generates an aligned position pt for each target word at time t. The context vector ct is then derived as a weighted average over the set of source hidden states within the window $[p_{tD}, p_{t+D}]$ where D is empirically selected.

## 4 Model

Our model is based on Xinya Du's [3] representation, which is based on the intuition that, to ask a natural question, people usually pay attention to certain parts of the input sentence, as well as associating context information from the paragraph We model the conditional probability using RNN encoder-

decoder architecture and adopt the global attention mechanism [6] to make the model focus on certain elements of the input when generating each word during decoding

### 4.1 Encoder

Attention-based sentence encoder: We use a bidirectional GRU to encode the sentence, where $\overrightarrow{b_t}$ is the hidden state at time step t for the forward pass GRU, $\overleftarrow{b_t}$ for the reverse pass.

$$\overrightarrow{b_t} = \overrightarrow{GRU2}(x_t, \overrightarrow{b_{t+1}})$$

$$\overleftarrow{b_t} = \overleftarrow{GRU2}(x_t, \overleftarrow{b_{t-1}})$$

### 4.2 Decoder

A stack of several recurrent units where each predicts an output $y_t$ at a time step t. Each recurrent unit accepts a hidden state from the previous unit and produces and output as well as its own hidden state. In our question-answering problem, the output sequence is a collection of all words from the answer. Each word is represented as $y_i$ where i is the order of that word. Any hidden state $h_i$ is computed using the formula

$$h_t = f(W^{(hh)} h_{t-1})$$

We are implementing teacher forcing which is a strategy for training recurrent neural networks that uses model output from a prior time step as an input.

## 5 Implementation

We experiment with our neural question generation model on the processed SQuAD dataset. In this section, we firstly describe the corpus of the task. We then give implementation details of our neural generation model. Subsequently, we introduce the evaluation methods by automatic metrics and human rater. We extract the question, answer, the sentence containing the answer and the index at which the sentence containing the answer starts from the SQuAD dataset and store them in a list. We have normalized the data by converting data to Unicode, lower case and removing special characters. We are using python library pandas to perform these data analysis and manipulations. Pandas will read our data as data frame and split it into our source and target sentence. We implement our model in PyTorch1.1.3[15] on top of the newly released minimal NMT [14] system.We replace rare tokens outside the vocabulary list by UNK symbol. We choose word embedding of 300 dimensions and use the glove.6B.300d pre-trained embeddings [16] for initialization. We fix the word representations during training. We set the GRU hidden unit size to 600 and set the number of layers of GRUs to 2 in both the encoder and the decoder. Optimization is performed using stochastic gradient descent (SGD), with an initial learning rate of 1.0. We start halving the learning rate at epoch 8. The mini-batch size for the update is set at 32 with dropout rate set at 0.3. [1]

[1]https://github.com/aditya140/ques_gen

## 6 Training

Our model is using adaptive moment estimation (Adam) as optimizer for using past gradients to calculate current gradients. Cross entropy is used as loss function to calculate the losses which leads to maximizing the probability of selecting the correct word at each time step with torch text for batching and warm restarts with annealing for learning rate (learning rate set at 1.2e4 ). The training process begins with feeding the pair of a sentence to the model to predict the correct output. At each step, the output from the model will be calculated with the true words to find the losses and update the parameters. For speed we have implemented torch text's BucketIterator [17] here to get batches containing sentences of almost the same length. We have used a train function with teacher forcing [18] to run encoder training, get the output from encoder to decoder and train the decoder, and finally for backward propagation.

## 7 Results and Analysis

We are using Cross-entropy loss [19] to measure the performance of our model. Cross-entropy loss increases as the predicted probability diverges from the actual label. As the iterations increase, we observe that log loss decreases rapidly. This means our model rate of error decreases linearly with iterations.

**Answer**: oklahoma is the 20th largest state in the united states , covering an area of 69,898 square miles ( 181,035 km2 ) , with 68,667 square miles ( 177847 km2 ) of land and 1,281 square miles ( 3,188 km2 ) of water
**Target**: how many square miles is oklahoma
**Predicted**: how many square miles is oklahoma

**Answer**: the french marines and naval infantry intended for the invasion of northern germany were dispatched to reinforce the french army of chalons and fell into captivity at sedan along with napoleon iii
**Target**: who also was captured at sedan
**Predicted**: where were the french marines dispatched

**Answer**: The game was played on february 7, 2016, at levi 's stadium in the san francisco bay area at Santa Clara , california
**Target**: what stadium did super bowl 50 take place in
**Predicted**: what venue did super bowl 50 take place

Figure1: List of Generated Sequences

We have implemented cosine annealing, which decreases learning rate in the form of half a cosine curve, we start out with relatively high learning rates for several iterations in the beginning to quickly approach a local minimum, then gradually decrease the learning rate as we get closer to the minimum, ending with small learning rate iterations. We have split the dataset into 70:30 ratio for validation. The model sets apart this fraction of the training data, will not train on it, and will evaluate the loss and any model metrics on this

data at the end of each epoch. We have observed the loss to be increasing which means that our model is overtraining.

# 8 Evaluation

## 8.1 Automatic Evaluation

We use the evaluation package Coco-caption [9] to rate the generated sequences. The package includes BLEU 1, BLEU 2, BLEU 3, BLEU 4 [10], METEOR [11] and ROUGEL [12] evaluation scripts. BLEU measures the average n-gram precision on a set of reference sentences, with a penalty for overly short sentences. METEOR is a recall-oriented metric, which calculates the similarity between generations and references by considering synonyms, stemming and paraphrases. ROUGE is commonly employed to evaluate n-grams recall of the summaries with gold standard sentences as references. It is essentially a metric for evaluating automatic summarization of texts as well as machine translation. It works by comparing an automatically produced summary or translation against a set of reference summaries (typically human-produced).

Table 1: Evaluation Results on 12000 pairs

| Metric | Score |
|---|---|
| BLEU 1 | 24.74532590851 |
| BLEU 2 | 12.8089235897 |
| BLEU 3 | 7.12667451620 |
| BLEU 4 | 4.098701330683 |
| METEOR | 9.6699067579 |
| ROGUE | 25.30183794252 |

## 8.2 Human Evaluation

We also perform human evaluation studies to measure the quality of questions generated by our system. We consider two modalities: grammatically and correctness of question. We randomly sampled 100 sentence-question pairs. We ask two professional English speakers to rate the pairs in terms of the modalities above on a 1–5 scale (5 for the best).

Table 2: Human Evaluation Results on 100 pairs

| Metric | Score |
|---|---|
| Grammaticality | 2.7/5 |
| Question Correctness | 1.6/5 |

# 9 Baselines

**IR** [22] stands for our information retrieval baselines. Similar to Rush et al. (2015), we implement the IR baselines to control memorizing questions from the training set. We use two metrics to calculate the distance between a question and the input sentence, i.e., BM-25 (Robertson and Walker, 1994) and edit distance (Levenshtein, 1966). According to the metric, the system retrieves the training set to find the question with the highest score.

**MOSES+** [21] is a widely used phrase-based statistical machine translation system. Here, we treat sentences as source language text, we treat questions as target language text, and we perform the translation from sentences to questions. We train a tri-gram language model on target side texts with KenLM (Heafield et al., 2013), and tune the system with MERT on dev set. Performance results are reported on the test set.

Table 3: Scores compared to other baselines

| Score | Our Model | $IR_{bm25}$ | MOSES+ |
|---|---|---|---|
| BLEU 1 | 24.74 | 5.18 | 15.61 |
| BLEU 2 | 12.80 | 0.91 | 3.64 |
| BLEU 3 | 7.12 | 0.28 | 1.00 |
| BLEU 4 | 4.09 | 0.12 | 0.30 |
| METEOR | 9.66 | 4.57 | 10.47 |
| ROGUE | 25.30 | 9.16 | 17.82 |

# 10 Conclusion and Future Work

We have presented a fully data-driven neural networks approach to automatic question generation for reading comprehension. We use an attention based neural networks approach for the task and investigate the effect of encoding sentence information. Our best model performes better in both automatic evaluations and human evaluations as compared to other rule based approaches. The model does not produce any staggering results for the question generation task with the dataset and modules we have used. From Results, we can see that current evaluation metrics do not support the evaluation of samples where a meaningful question is generated but is not present in the dataset. Hence better metrics are needed to evaluate such results. Possibly training the model by experimenting with multiple- head attention techniques, or more experimenting with data preprocessing, lemmatization with efficient batching and optimization techniques could yield better results. We would like to explore how to better use these mechanisms. We would also like to consider implementing a transformer model architecture with stacks of encoders and decoders for better connectivity between the modules to further improve the quality of generated questions.

# References


[1] Tom Young, Devamanyu Hazarika, Soujanya Poria, Erik Cambria *Recent Trends in Deep Learning Based Natural Language Processing*.

[2] Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. *Empirical evaluation of gated recurrent neural networks on sequence modeling.* CoRR, abs/1412.3555, 2014

[3] Xinya Du Junru Shao Claire Cardie *Learning to Ask: Neural Question Generation for Reading Comprehension.*

[4] 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 (EMNLP). Association for Computational Linguistics, Austin, Texas, pages 2383–2392.* https://aclweb.org/anthology/D16- 1264

[5] Ashish Vaswani Noam Shazeer Niki Parmar Jakob Uszkoreit *Attention Is All You Need*

[6] Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015b. Addressing the rare word problem in neural machine translation. *In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, Beijing, China, pages 11–19.* http://www.aclweb.org/anthology/P15-1002.

[7] Jack Mostow and Wei Chen. 2009. Generating instruction automatically for the reading strategy of selfquestioning. *In Proceedings of the 2nd Workshop on Question Generation (AIED 2009). pages 465–472.*

[8] Santanu Pal, Tapabrata Mondal, Partha Pakray, Dipankar Das and Sivaji Bandyopadhyay GSTEC System Description – JUQGG: *A Rule based approach*

[9] https://github.com/tylin/coco-caption

[10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of 40th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, pages 311–318. https://doi.org/10.3115/1073083.1073135.

[11] [11] Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. *In Proceedings of the Ninth Workshop on Statistical Machine Translation. Association for Computational Linguistics, Baltimore, Maryland, USA, pages 376– 380. http://www.aclweb.org/anthology/W14-3348.*

[12] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *In Stan Szpakowicz Marie-Francine Moens, editor, Text Summarization Branches Out: Proceedings of the ACL-04 Workshop. Association for Computational Linguistics, Barcelona, Spain, pages 74–81.* http://aclweb.org/anthology/W/W04/W04-1013.pdf.

[13] Michael Heilman and Noah A. Smith. 2010. *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. Association for Computational Linguistics, Los Angeles, California, pages 609–617.* http://www.aclweb.org/anthology/N10-1086.

[14] Denny Britz†, Anna Goldie, Minh-Thang Luong, Quoc Le Massive Exploration of Neural Machine Translation Architectures

[15] https://github.com/pytorch/pytorch

[16] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. *Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Doha, Qatar, pages 1532–1543.* http://www.aclweb.org/anthology/D14-1162.

[17] https://github.com/pytorch/text/blob/master/docs/source/data.rst

[18] https://www.tensorflow.org/tutorials/text/ *nmt with attention*

[19] https://ml-cheatsheet.readthedocs.io/en/latest loss functions

[20] Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Lisbon, Portugal, pages 379–389. http://aclweb.org/anthology/D15-1044.

[21] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. *Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions. Association for Computational Linguistics, Stroudsburg, PA, USA, pages 177–180.* http://dl.acm.org/citation.cfm?id=1557769.1557821.