Learning to Ask: Neural Question Generation for Reading Comprehension

Xinya Du[1]   **Junru Shao**[2]   **Claire Cardie**[1]

[1]Department of Computer Science, Cornell University

[2]Zhiyuan College, Shanghai Jiao Tong University

{xdu, cardie}@cs.cornell.edu   yz_sjr@sjtu.edu.cn

Abstract

We study automatic question generation for sentences from text passages in reading comprehension. We introduce an attention-based sequence learning model for the task and investigate the effect of encoding sentence- vs. paragraph-level information. In contrast to all previous work, our model does not rely on hand-crafted rules or a sophisticated NLP pipeline; it is instead trainable end-to-end via sequence-to-sequence learning. Automatic evaluation results show that our system significantly outperforms the state-of-the-art rule-based system. In human evaluations, questions generated by our system are also rated as being more natural (*i.e.*, grammaticality, fluency) and as more difficult to answer (in terms of syntactic and lexical divergence from the original text and reasoning needed to answer).

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 — to generate questions for reading comprehension materials (Heilman and Smith, 2010). Figure 1, for example, shows three manually generated questions that test a user's understanding of the associated text passage. Question generation systems can also be deployed as chatbot components (*e.g.*, asking questions to start a conversation or to request feedback (Mostafazadeh et al., 2016)) or, arguably, as a clinical tool for evaluating or improving mental health (Weizenbaum, 1966; Colby et al., 1971).

In addition to the above applications, question generation systems can aid in the development of

---

Sentence:

Oxygen is used in cellular respiration and released by photosynthesis, which uses the energy of sunlight to produce oxygen from water.

Questions:

– What life process produces oxygen in the presence of light?
  photosynthesis

– Photosynthesis uses which energy to form oxygen from water?
  sunlight

– From what does photosynthesis get oxygen?
  water

Figure 1: Sample sentence from the second paragraph of the article *Oxygen*, along with the natural questions and their answers.

annotated data sets for natural language processing (NLP) research in reading comprehension and question answering. Indeed the creation of such datasets, *e.g.*, SQuAD (Rajpurkar et al., 2016) and MS MARCO (Nguyen et al., 2016), has spurred research in these areas.

For the most part, question generation has been tackled in the past via rule-based approaches (*e.g.*, Mitkov and Ha (2003); Rus et al. (2010). The success of these approaches hinges critically on the existence of well-designed rules for declarative-to-interrogative sentence transformation, typically based on deep linguistic knowledge.

To improve over a purely rule-based system, Heilman and Smith (2010) introduced an overgenerate-and-rank approach that generates multiple questions from an input sentence using a rule-based approach and then ranks them using a supervised learning-based ranker. Although the ranking algorithm helps to produce more ac-

ceptable questions, it relies heavily on a manually crafted feature set, and the questions generated often overlap word for word with the tokens in the input sentence, making them very easy to answer.

Vanderwende (2008) point out that learning to ask good questions is an important task in NLP research in its own right, and should consist of more than the syntactic transformation of a declarative sentence. In particular, a natural sounding question often compresses the sentence on which it is based (*e.g.*, question 3 in Figure 1), uses synonyms for terms in the passage (*e.g.*, "form" for "produce" in question 2 and "get" for "produce" in question 3), or refers to entities from preceding sentences or clauses (*e.g.*, the use of "photosynthesis" in question 2). Othertimes, world knowledge is employed to produce a good question (*e.g.*, identifying "photosynthesis" as a "life process" in question 1). In short, constructing natural questions of reasonable difficulty would seem to require an *abstractive* approach that can produce fluent phrasings that do not exactly match the text from which they were drawn.

As a result, and in contrast to all previous work, we propose here to frame the task of question generation as a sequence-to-sequence learning problem that directly maps a sentence from a text passage to a question. Importantly, our approach is fully data-driven in that it requires no manually generated rules.

More specifically, inspired by the recent success in neural machine translation (Sutskever et al., 2014; Bahdanau et al., 2015), summarization (Rush et al., 2015; Iyer et al., 2016), and image caption generation (Xu et al., 2015), we tackle question generation using a conditional neural language model with a global attention mechanism (Luong et al., 2015a). We investigate several variations of this model, including one that takes into account paragraph- rather than sentence-level information from the reading passage as well as other variations that determine the importance of pre-trained vs. learned word embeddings.

In evaluations on the SQuAD dataset (Rajpurkar et al., 2016) using three automatic evaluation metrics, we find that our system significantly outperforms a collection of strong baselines, including an information retrieval-based system (Robertson and Walker, 1994), a statistical machine translation approach (Koehn et al., 2007), and the overgenerate-and-rank approach of Heil-

man and Smith (2010). Human evaluations also rated our generated questions as more grammatical, fluent, and challenging (in terms of syntactic divergence from the original reading passage and reasoning needed to answer) than the state-of-the-art Heilman and Smith (2010) system.

In the sections below we discuss related work (Section 2), specify the task definition (Section 3) and describe our neural sequence learning based models (Section 4). We explain the experimental setup in Section 5. Lastly, we present the evaluation results as well as a detailed analysis.

2 Related Work

Reading Comprehension is a challenging task for machines, requiring both understanding of natural language and knowledge of the world (Rajpurkar et al., 2016). Recently many new datasets have been released and in most of these datasets, the questions are generated in a synthetic way. For example, bAbI (Weston et al., 2016) is a fully synthetic dataset featuring 20 different tasks. Hermann et al. (2015) released a corpus of cloze style questions by replacing entities with placeholders in abstractive summaries of CNN/Daily Mail news articles. Chen et al. (2016) claim that the CNN/Daily Mail dataset is easier than previously thought, and their system almost reaches the ceiling performance. Richardson et al. (2013) curated MCTest, in which crowdworker questions are paired with four answer choices. Although MCTest contains challenging natural questions, it is too small for training data-demanding question answering models.

Recently, Rajpurkar et al. (2016) released the Stanford Question Answering Dataset[1] (SQuAD), which overcomes the aforementioned small size and (semi-)synthetic issues. The questions are posed by crowd workers and are of relatively high quality. We use SQuAD in our work, and similarly, we focus on the generation of natural questions for reading comprehension materials, albeit via automatic means.

Question Generation has attracted the attention of the natural language generation (NLG) community in recent years, since the work of Rus et al. (2010).

Most work tackles the task with a rule-based approach. Generally, they first transform the input sentence into its syntactic representation, which

---

[1] https://stanford-qa.com

they then use to generate an interrogative sentence. A lot of research has focused on first manually constructing question templates, and then applying them to generate questions (Mostow and Chen, 2009; Lindberg et al., 2013; Mazidi and Nielsen, 2014). Labutov et al. (2015) use crowdsourcing to collect a set of templates and then rank the relevant templates for the text of another domain. Generally, the rule-based approaches make use of the syntactic roles of words, but not their semantic roles.

Heilman and Smith (2010) 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. Mostafazadeh et al. (2016) introduce visual question generation task, to explore the deep connection between language and vision. Serban et al. (2016) propose generating simple factoid questions from logic triple (subject, relation, object). Their task tackles mapping from structured representation to natural language text, and their generated questions are consistent in terms of format and diverge much less than ours.

To our knowledge, none of the previous works has framed QG for reading comprehension in an end-to-end fashion, and nor have them used deep sequence-to-sequence learning approach to generate questions.

3   Task Definition

In this section, we define the question generation task. Given an input sentence $\mathbf{x}$, our goal is to generate a natural question $\mathbf{y}$ related to information in the sentence, $\mathbf{y}$ can be a sequence of an arbitrary length: $[y_1, ..., y_{|\mathbf{y}|}]$. Suppose the length of the input sentence is M, $\mathbf{x}$ could then be represented as a sequence of tokens $[x_1, ..., x_M]$. The QG task is defined as finding $\overline{\mathbf{y}}$, such that:

$$\overline{\mathbf{y}} = \arg\max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}) \qquad (1)$$

where $P(\mathbf{y}|\mathbf{x})$ is the conditional log-likelihood of the predicted question sequence $\mathbf{y}$, given the input $\mathbf{x}$. In section 4.1, we will elaborate on the global attention mechanism for modeling $P(\mathbf{y}|\mathbf{x})$.

4   Model

Our model is partially inspired by the way in which a human would solve the task. 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 (Bahdanau et al., 2015; Cho et al., 2014), and adopt the global attention mechanism (Luong et al., 2015a) to make the model focus on certain elements of the input when generating each word during decoding.

Here, we investigate two variations of our models: one that only encodes the sentence and another that encodes both sentence and paragraph-level information.

4.1   Decoder

Similar to Sutskever et al. (2014) and Chopra et al. (2016), we factorize the the conditional in equation 1 into a product of word-level predictions:

$$P(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{|y|} P(y_t|\mathbf{x}, y_{<t})$$

where probability of each y_t is predicted based on all the words that are generated previously (*i.e.*, $y_{<t}$), and input sentence $\mathbf{x}$.

More specifically,

$$P(y_t|\mathbf{x}, y_{<t}) = \text{softmax}(\mathbf{W}_s\text{tanh}(\mathbf{W}_t[\mathbf{h}_t; \mathbf{c}_t]))$$
$$(2)$$

with $\mathbf{h}_t$ being the recurrent neural networks state variable at time step t, and $\mathbf{c}_t$ being the attention-based encoding of $\mathbf{x}$ at decoding time step t (Section 4.2). $\mathbf{W}_s$ and $\mathbf{W}_t$ are parameters to be learned.

$$\mathbf{h}_t = \text{LSTM}_1(y_{t-1}, \mathbf{h}_{t-1}) \qquad (3)$$

here, LSTM is the Long Short-Term Memory (LSTM) network (Hochreiter and Schmidhuber, 1997). It generates the new state $\mathbf{h}_t$, given the representation of previously generated word y_{t-1} (obtained from a word look-up table), and the previous state $\mathbf{h}_{t-1}$.

The initialization of the decoder's hidden state differentiates our basic model and the model that incorporates paragraph-level information.

For the basic model, it is initialized by the sentence's representation $\mathbf{s}$ obtained from the sentence encoder (Section 4.2). For our paragraph-level model, the *concatenation* of the sentence

encoder's output $\mathbf{s}$ and the paragraph encoder's output $\mathbf{s}'$ is used as the initialization of decoder hidden state. To be more specific, the architecture of our paragraph-level model is like a "Y"-shaped network which encodes both sentence- and paragraph-level information via two RNN branches and uses the concatenated representation for decoding the questions.

4.2 Encoder

The attention-based sentence encoder is used in both of our models, while the paragraph encoder is only used in the model that incorporates paragraph-level information.

Attention-based sentence encoder:

We use a bidirectional LSTM to encode the sentence,

$$\overrightarrow{\mathbf{b}_t} = \overrightarrow{\mathrm{LSTM}_2} \left(x_t, \overrightarrow{\mathbf{b}_{t-1}} \right)$$
$$\overleftarrow{\mathbf{b}_t} = \overleftarrow{\mathrm{LSTM}_2} \left(x_t, \overleftarrow{\mathbf{b}_{t+1}} \right)$$

where $\overrightarrow{\mathbf{b}_t}$ is the hidden state at time step t for the forward pass LSTM, $\overleftarrow{\mathbf{b}_t}$ for the backward pass.

To get *attention-based* encoding of $\mathbf{x}$ at decoding time step t, namely, $\mathbf{c}_t$, we first get the context dependent token representation by $\mathbf{b}_t = [\overrightarrow{\mathbf{b}_t}; \overleftarrow{\mathbf{b}_t}]$, then we take the weighted average over $\mathbf{b}_t$ ($t = 1, ..., |\mathbf{x}|$),

$$\mathbf{c}_t = \sum_{i=1,...,|\mathbf{x}|} a_{i,t} \mathbf{b}_i \qquad (4)$$

The attention weight are calculated by the bi-linear scoring function and softmax normalization,

$$a_{i,t} = \frac{\exp\left(\mathbf{h}_t^T \mathbf{W}_b \mathbf{b}_i\right)}{\sum_j \exp\left(\mathbf{h}_t^T \mathbf{W}_b \mathbf{b}_j\right)} \qquad (5)$$

To get the sentence encoder's output for initialization of decoder hidden state, we concatenate last hidden state of the forward and backward pass, namely, $\mathbf{s} = [\overrightarrow{\mathbf{b}_{|\mathbf{x}|}}; \overleftarrow{\mathbf{b}_1}]$.

Paragraph encoder:

Given sentence $\mathbf{x}$, we want to encode the paragraph containing $\mathbf{x}$. Since in practice the paragraph is very long, we set a length threshold L, and truncate the paragraph at the Lth token. We call the truncated paragraph "paragraph" henceforth.

Denoting the paragraph as $\mathbf{z}$, we use another bidirectional LSTM to encode $\mathbf{z}$,

$$\overrightarrow{\mathbf{d}_t} = \overrightarrow{\mathrm{LSTM}_3} \left(z_t, \overrightarrow{\mathbf{d}_{t-1}} \right)$$
$$\overleftarrow{\mathbf{d}_t} = \overleftarrow{\mathrm{LSTM}_3} \left(z_t, \overleftarrow{\mathbf{d}_{t+1}} \right)$$

With the last hidden state of the forward and backward pass, we use the concatenation $[\overrightarrow{\mathbf{d}_{|\mathbf{z}|}}; \overleftarrow{\mathbf{d}_1}]$ as the paragraph encoder's output $\mathbf{s}'$.

4.3 Training and Inference

Giving a training corpus of sentence-question pairs: $\mathcal{S} = \left\{ \left(\mathbf{x}^{(i)}, \mathbf{y}^{(i)} \right) \right\}_{i=1}^{S}$, our models' training objective is to minimize the negative log-likelihood of the training data with respect to all the parameters, as denoted by θ,

$$\mathcal{L} = -\sum_{i=1}^{S} \log P\left(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}; \theta\right)$$
$$= -\sum_{i=1}^{S} \sum_{j=1}^{|\mathbf{y}^{(i)}|} \log P\left(y_j^{(i)} | \mathbf{x}^{(i)}, y_{<j}^{(i)}; \theta\right)$$

Once the model is trained, we do inference using beam search. The beam search is parametrized by the possible paths number k.

As there could be many rare words in the input sentence that are not in the target side dictionary, during decoding many UNK tokens will be output. Thus, post-processing with the replacement of UNK is necessary. Unlike Luong et al. (2015b), we use a simpler replacing strategy for our task. For the decoded UNK token at time step t, we replace it with the token in the input sentence with the highest attention score, the index of which is $\arg\max_i a_{i,t}$.

5 Experimental Setup

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, the baselines to compare, and their experimental settings. Lastly, we introduce the evaluation methods by automatic metrics and human raters.

5.1 Dataset

With the SQuAD dataset (Rajpurkar et al., 2016), we extract sentences and pair them with the ques-
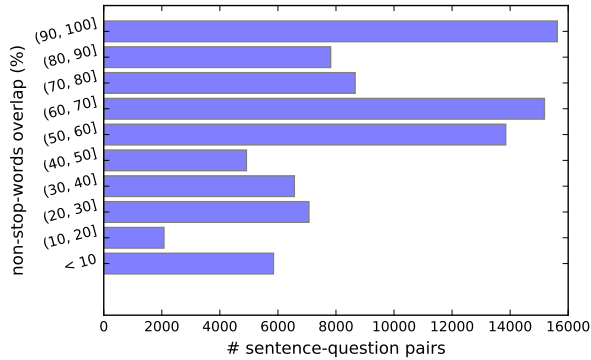
Figure 2: Overlap percentage of sentence-question pairs in training set. y-axis is # non-stop-words overlap with respect to the total # tokens in the question (a percentage); x-axis is # sentence-question pairs for a given overlap percentage range.

tions. We train our models with the sentence-question pairs. The dataset contains 536 articles with over 100k questions posed about the articles. The authors employ Amazon Mechanical Turks crowd-workers to create questions based on the Wikipedia articles. Workers are encouraged to use their own words without any copying phrases from the paragraph. Later, other crowd-workers are employed to provide answers to the questions. The answers are spans of tokens in the passage.

Since there is a hidden part of the original SQuAD that we do not have access to, we treat the accessible parts (∼90%) as the entire dataset henceforth.

We first run Stanford CoreNLP (Manning et al., 2014) for pre-processing: tokenization and sentence splitting. We then lower-case the entire dataset. With the offset of the answer to each question, we locate the sentence containing the answer and use it as the input sentence. In some cases (< 0.17% in training set), the answer spans two or more sentences, and we then use the concatenation of the sentences as the input "sentence".

Figure 2 shows the distribution of the token overlap percentage of the sentence-question pairs. Although most of the pairs have over 50% overlap rate, about 6.67% of the pairs have no non-stop-words in common, and this is mostly because of the answer offset error introduced during annotation. Therefore, we prune the training set based on the constraint: the sentence-question pair must have at least one non-stop-word in common. Lastly we add <SOS> to the beginning of the sen-

# pairs (Train)	70484
# pairs (Dev)	10570
# pairs (Test)	11877
Sentence: avg. tokens	32.9
Question: avg. tokens	11.3
Avg. # questions per sentence	1.4

Table 1: Dataset (processed) statistics. Sentence average # tokens, question average # tokens, and average # questions per sentence statistics are from training set. These averages are close to the statistics on development set and test set.

tences, and <EOS> to the end of them.

We randomly divide the dataset at the article-level into a training set (80%), a development set (10%), and a test set (10%). We report results on the 10% test set.

Table 1 provides some statistics on the processed dataset: there are around 70k training samples, the sentences are around 30 tokens, and the questions are around 10 tokens on average. For each sentence, there might be multiple corresponding questions, and, on average, there are 1.4 questions for each sentence.

5.2 Implementation Details

We implement our models [2] in Torch7 [3] on top of the newly released OpenNMT system (Klein et al., 2017).

For the source side vocabulary $\mathcal{V}$, we only keep the 45k most frequent tokens (including <SOS>, <EOS> and placeholders). For the target side vocabulary $\mathcal{U}$, similarly, we keep the 28k most frequent tokens. All other tokens outside the vocabulary list are replaced by the UNK symbol. We choose word embedding of 300 dimensions and use the glove.840B.300d pre-trained embeddings (Pennington et al., 2014) for initialization. We fix the word representations during training.

We set the LSTM hidden unit size to 600 and set the number of layers of LSTMs 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 64. Dropout with probability

---

[2]The code is available at https://github.com/xinyadu/nqg.
[3]http://torch.ch/

Model	BLEU 1	BLEU 2	BLEU 3	BLEU 4	METEOR	ROUGE$_L$
IR$_{BM25}$	5.18	0.91	0.28	0.12	4.57	9.16
IR$_{Edit Distance}$	18.28	5.48	2.26	1.06	7.73	20.77
MOSES+	15.61	3.64	1.00	0.30	10.47	17.82
DirectIn	31.71	21.18	15.11	11.20	14.95	22.47
H&S	38.50	22.80	15.52	11.18	15.95	30.98
Vanilla seq2seq	31.34	13.79	7.36	4.26	9.88	29.75
Our model (no pre-trained)	41.00	23.78	15.71	10.80	15.17	37.95
Our model (w/ pre-trained)	**43.09**	**25.96**	**17.50**	**12.28**	**16.62**	**39.75**
+ paragraph	42.54	25.33	16.98	11.86	16.28	39.37

Table 2: Automatic evaluation results of different systems by BLEU 1–4, METEOR and ROUGE$_L$. For a detailed explanation of the baseline systems, please refer to Section 5.3. The best performing system for each column is highlighted in boldface. Our system which encodes only sentence with pre-trained word embeddings achieves the best performance across all the metrics.

0.3 is applied between vertical LSTM stacks. We clip the gradient when the its norm exceeds 5.

All our models are trained on a single GPU. We run the training for up to 15 epochs, which takes approximately 2 hours. We select the model that achieves the lowest perplexity on the dev set.

During decoding, we do beam search with a beam size of 3. Decoding stops when every beam in the stack generates the <EOS> token.

All hyperparameters of our model are tuned using the development set. The results are reported on the test set.

5.3 Baselines

To prove the effectiveness of our system, we compare it to several competitive systems. Next, we briefly introduce their approaches and the experimental setting to run them for our problem. Their results are shown in Table 2.

IR 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+ (Koehn et al., 2007) 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 ques-

tions. 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.

DirectIn is an intuitive yet meaningful baseline in which the longest sub-sentence of the sentence is *directly* taken as the predicted question. [4] To split the sentence into sub-sentences, we use a set of splitters, *i.e.*, {"?", "!", ",", ".", ";"}.

H&S is the rule-based overgenerate-and-rank system that was mentioned in Section 2. When running the system, we set the parameter just-wh true (to restrict the output of the system to being only wh-questions) and set max-length equal to the longest sentence in the training set. We also set downweight-pro true, to down weight questions with unresolved pronouns so that they appear towards the end of the ranked list. For comparison with our systems, we take the top question in the ranked list.

Seq2seq (Sutskever et al., 2014) is a basic encoder-decoder sequence learning system for machine translation. We implement their model in Tensorflow. The input sequence is reversed before training or translating. Hyperparameters are tuned with dev set. We select the model with the lowest perplexity on the dev set.

---

[4]We also tried using the *entire* input sentence as the prediction output, but the performance is worse than taking sub-sentence as the prediction, across all the automatic metrics except for METEOR.

	Naturalness	Difficulty	Best %	Avg. rank
H&S	2.95	1.94	20.20	2.29
Ours	**3.36**	**3.03**[*]	**38.38**[*]	**1.94**[**]
Human	3.91	2.63	66.42	1.46

Table 3: Human evaluation results for question generation. Naturalness and difficulty are rated on a 1–5 scale (5 for the best). Two-tailed t-test results are shown for our method compared to H&S (statistical significance is indicated with [*]($p < 0.005$), [**]($p < 0.001$)).

5.4 Automatic Evaluation

We use the evaluation package released by Chen et al. (2015), which was originally used to score image captions. The package includes BLEU 1, BLEU 2, BLEU 3, BLEU 4 (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014) and ROUGE$_L$ (Lin, 2004) evaluation scripts. BLEU measures the average n-gram precision on a set of reference sentences, with a penalty for overly short sentences. BLEU-n is BLEU score that uses up to n-grams for counting co-occurrences. 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. ROUGE$_L$ (measured based on longest common subsequence) results are reported.

5.5 Human Evaluation

We also perform human evaluation studies to measure the quality of questions generated by our system and the H&S system. We consider two modalities: *naturalness*, which indicates the grammaticality and fluency; and *difficulty*, which measures the sentence-question syntactic divergence and the reasoning needed to answer the question. We randomly sampled 100 sentence-question pairs. We ask four professional English speakers to rate the pairs in terms of the modalities above on a 1–5 scale (5 for the best). We then ask the human raters to give a ranking of the questions according to the overall quality, with ties allowed.

6 Results and Analysis

Table 2 shows automatic metric evaluation results for our models and baselines. Our model which only encodes sentence-level information achieves

Sentence 1: the largest of these is the eldon square shopping centre , one of the largest city centre shopping complexes in the uk .
Human: what is one of the largest city center shopping complexes in the uk ?
H&S: what is the eldon square shopping centre one of ?
Ours: what is one of the largest city centers in the uk ?

Sentence 2: free oxygen first appeared in significant quantities during the paleoproterozoic eon -lrb- between 3.0 and 2.3 billion years ago -rrb- .
Human: during which eon did free oxygen begin appearing in quantity ?
H&S: what first appeared in significant quantities during the paleoproterozoic eon ?
Ours: how long ago did the paleoproterozoic exhibit ?

Sentence 3: inflammation is one of the first responses of the immune system to infection .
Human: what is one of the first responses the immune system has to infection ?
H&S: what is inflammation one of ?
Ours: what is one of the first objections of the immune system to infection ?

Sentence 4: tea , coffee , sisal , pyrethrum , corn , and wheat are grown in the fertile highlands , one of the most successful agricultural production regions in Africa.
Human: (1) where is the most successful agricultural prodcution regions ? (2) what is grown in the fertile highlands ?
H&S: what are grown in the fertile highlands in africa ?
Ours: what are the most successful agricultural production regions in africa ?

Sentence 5: as an example , income inequality did fall in the united states during its high school movement from 1910 to 1940 and thereafter .
Human: during what time period did income inequality decrease in the united states ?
H&S: where did income inequality do fall during its high school movement from 1910 to 1940 and thereafter as an example ?
Ours: when did income inequality fall in the us ?

Sentence 6: however , the rainforest still managed to thrive during these glacial periods , allowing for the survival and evolution of a broad diversity of species .
Human: did the rainforest managed to thrive during the glacial periods ?
H&S: what allowed for the survival and evolution of a broad diversity of species?
Ours: why do the birds still grow during glacial periods ?

Sentence 7: maududi founded the jamaat-e-islami party in 1941 and remained its leader until 1972.
Human: when did maududi found the jamaat-e-islami party ?
H&S: who did maududi remain until 1972 ?
Ours: when was the jamaat-e-islami party founded ?

Figure 3: Sample output questions generated by human (ground truth questions), our system and the H&S system.

Category	(%)	H&S			Ours			Ours + paragraph		
		BLEU-3	BLEU-4	METEOR	BLEU-3	BLEU-4	METEOR	BLEU-3	BLEU-4	METEOR
w/ sentence	70.23 (243)	20.64	15.81	16.76	**24.45**	**17.63**	17.82	24.01	16.39	**19.19**
w/ paragraph	19.65 (68)	6.34	< 0.01	10.74	3.76	< 0.01	11.59	**7.23**	**4.13**	**12.13**
All*	100 (346)	19.97	14.95	16.68	23.63	**16.85**	17.62	**24.68**	16.33	**19.61**

Table 4: An estimate of categories of questions of the processed dataset and per-category performance comparison of the systems. The estimate is based on our analysis of the 346 pairs from the dev set. Categories are decided by the information needed to generate the question. Bold numbers represent the best performing method for a given metric. *Here, we leave out performance results for "w/ article" category (2 samples, 0.58%) and "not askable" category (33 samples, 9.54%).

the best performance across all metrics. We note that IR performs poorly, indicating that memorizing the training set is not enough for the task. The baseline DirectIn performs pretty well on BLEU and METEOR, which is reasonable given the overlap statistics between the sentences and the questions (Figure 2). H&S system's performance is on a par with DirectIn's, as it basically performs syntactic change without paraphrasing, and the overlap rate is also high.

Looking at the performance of our three models, it's clear that adding the pre-trained embeddings generally helps. While encoding the paragraph causes the performance to drop a little, this makes sense because, apart from useful information, the paragraph also contains much noise.

Table 3 shows the results of the human evaluation. We see that our system outperforms H&S in all modalities. Our system is ranked best in 38.4% of the evaluations, with an average ranking of 1.94. An inter-rater agreement of Krippendorff's Alpha of 0.236 is achieved for the overall ranking. The results imply that our model can generate questions of better quality than the H&S system. An interesting phenomenon here is that human raters gave higher score for our system's outputs than the human questions. One potential explanation for this is that our system is trained on *all* sentence-question pairs for one input sentence, while we randomly select one question among the several questions of one sentence as the human generated question, for the purpose of rating. Thus our system's predictions tend to be more diverse.

For our qualitative analysis, we examine the sample outputs and the visualization of the alignment between the input and the output. In Figure 3, we present sample questions generated by H&S and our best model. We see a large gap between our results and H&S's. For example, in
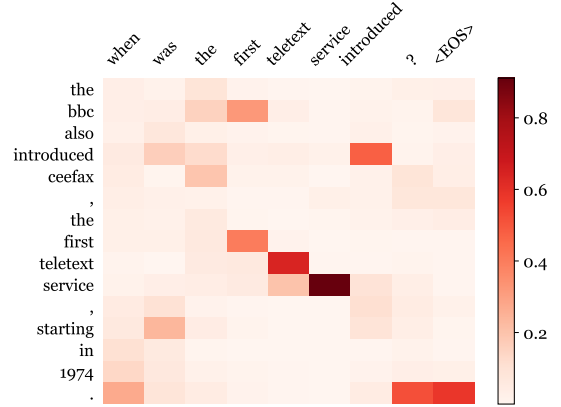

Figure 4: Heatmap of the attention weight matrix, which shows the soft alignment between the sentence (left) and the generated question (top).

the first sample, in which the focus should be put on "the largest." Our model successfully captures this information, while H&S only performs some syntactic transformation over the input without paraphrasing. However, outputs from our system are not always "perfect", for example, in pair 6, our system generates a question about the reason why birds still grow, but the *most related* question would be why many species still grow. But from a different perspective, our question is more challenging (readers need to understand that birds are one kind of species), which supports our system's performance listed in human evaluations (See Table 3). It would be interesting to further investigate how to interpret why certain irrelavant words are generated in the question. Figure 4 shows the attention weights ($\alpha_{i,t}$) for the input sentence when generating each token in the question. We see that the key words in the output ("introduced", "teletext", etc.) aligns well with those in the input sentence.

Finally, we do a dataset analysis and fine-

grained system performance analysis. We randomly sampled 346 sentence-question pairs from the dev set and label each pair with a category. [5] The four categories are determined by *how much* information is needed to ask the question. To be specific, "w/ sentence" means it only requires the sentence to ask the question; "w/ paragraph" means it takes other information in the paragraph to ask the question; "w/ article" is similar to "w/ paragraph"; and "not askable" means that world knowledge is needed to ask the question or there is mismatch of sentence and question caused by annotation error.

Table 4 shows the per-category performance of the systems. Our model which encodes paragraph information achieves the best performance on the questions of "w/ paragraph" category. This verifies the effectiveness of our paragraph-level model on the questions concerning information outside the sentence.

7 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- vs. paragraph-level information. Our best model achieves state-of-the-art performance in both automatic evaluations and human evaluations.

Here we point out several interesting future research directions. Currently, our paragraph-level model does not achieve best performance across all categories of questions. We would like to explore how to better use the paragraph-level information to improve the performance of QG system regarding questions of all categories. Besides this, it would also be interesting to consider to incorporate mechanisms for other language generation tasks (*e.g.*, copy mechanism for dialogue generation) in our model to further improve the quality of generated questions.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations Workshop (ICLR)*.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 2358–2367. http://www.aclweb.org/anthology/P16-1223.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325* .

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1724–1734. http://www.aclweb.org/anthology/D14-1179.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 93–98. http://www.aclweb.org/anthology/N16-1012.

Kenneth Mark Colby, Sylvia Weber, and Franklin Dennis Hilf. 1971. Artificial paranoia. *Artificial Intelligence* 2(1):1–25. https://doi.org/10.1016/0004-3702(71)90002-6.

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.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified kneser-ney language model estimation. In *Proceedings of the 51st Annual Meeting of the*

Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, Sofia, Bulgaria, pages 690–696. http://www.aclweb.org/anthology/P13-2121.

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.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems (NIPS)*. pages 1693–1701.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. 2016. Summarizing source code using a neural attention model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 2073–2083. http://www.aclweb.org/anthology/P16-1195.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *ArXiv e-prints* .

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej 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.

Igor Labutov, Sumit Basu, and Lucy Vanderwende. 2015. Deep questions without deep understanding. 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 889–898. http://www.aclweb.org/anthology/P15-1086.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet physics doklady*. volume 10, page 707.

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.

David Lindberg, Fred Popowich, John Nesbit, and Phil Winne. 2013. Generating natural language questions to support learning on-line. In *Proceedings of the 14th European Workshop on Natural Language Generation*. Association for Computational Linguistics, Sofia, Bulgaria, pages 105–114. http://www.aclweb.org/anthology/W13-2114.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015a. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1412–1421. http://aclweb.org/anthology/D15-1166.

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.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny F., Steven B., and David M. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Baltimore, Maryland, pages 55–60. http://www.aclweb.org/anthology/P14-5010.

Karen Mazidi and Rodney D. Nielsen. 2014. Linguistic considerations in automatic question generation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Baltimore, Maryland, pages 321–326. http://www.aclweb.org/anthology/P14-2053.

Ruslan Mitkov and Le An Ha. 2003. Computer-aided generation of multiple-choice tests. In Jill Burstein and Claudia Leacock, editors, *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*. pages 17–22. http://www.aclweb.org/anthology/W03-0203.pdf.

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)*. Association for Computational Linguistics, Berlin, Germany, pages 1802–1813. http://www.aclweb.org/anthology/P16-1170.

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

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* .

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.

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.

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.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, USA, pages 193–203. http://www.aclweb.org/anthology/D13-1020.

Stephen E. Robertson and Steve Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Springer-Verlag New York, Inc., New York, NY, USA, SIGIR '94, pages 232–241. http://dl.acm.org/citation.cfm?id=188490.188561.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. The first question generation shared task evaluation challenge. In *Proceedings of the 6th International Natural Language Generation Conference*. Association for Computational Linguistics, Stroudsburg, PA, USA, pages 251–257. http://dl.acm.org/citation.cfm?id=1873738.1873777.

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.

Iulian Vlad Serban, Alberto García-Durán, Caglar Gulcehre, Sungjin Ahn, Sarath Chandar, Aaron Courville, and Yoshua Bengio. 2016. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 588–598. http://www.aclweb.org/anthology/P16-1056.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems (NIPS)*. pages 3104–3112.

Lucy Vanderwende. 2008. The importance of being important: Question generation. In *Proceedings of the 1st Workshop on the Question Generation Shared Task Evaluation Challenge, Arlington, VA*.

Joseph Weizenbaum. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Commun. ACM* 9(1):36–45. https://doi.org/10.1145/365153.365168.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. In *International Conference on Learning Representations Workshop (ICLR)*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*. volume 14, pages 77–81.