Automatic Opinion Question Generation

Yllias Chali
University of Lethbridge
4401 University Drive
Lethbridge, Alberta, T1K 3M4
chali@cs.uleth.ca

Tina Baghaee
University of Lethbridge
4401 University Drive
Lethbridge, Alberta, T1K 3M4
tina.baghaee@gmail.com

Abstract

We study the problem of opinion question generation from sentences with the help of community-based question answering systems. For this purpose, we use a sequence to sequence attentional model, and we adopt coverage mechanism to prevent sentences from repeating themselves. Experimental results on the Amazon question/answer dataset show an improvement in automatic evaluation metrics as well as human evaluations from the state-of-the-art question generation systems.

1 Introduction

Question generation (QG) can be considered as a task which affects many aspects of people's lives. One of the main significance of the question generation is its capability to improve one's learning ability. Studies have shown that asking questions can help students realize their knowledge deficits and encourages them to look for information to compensate for those deficits (Graesser and Person, 1994). Additionally, QG can be used as an aid to search engines by providing suggestions regarding the users' queries (Chali and Hasan, 2015). This way, the users can either choose one of those suggestions or obtain a better idea on how to modify their query to get better results. Moreover, QG can assist the reading comprehension task and the question answering community by providing a robust input for their systems (Serban et al., 2016; Rajpurkar et al., 2016; Yang et al., 2017).

In this work, we propose a sequence to sequence model that uses attention and coverage mechanisms for addressing the question generation problem at the sentence level. The attention and coverage mechanisms prevent language generation systems from generating the same word over and over again, and have been shown to improve a system's output (See et al., 2017).

We benefit from the community-based question answering systems. Specifically, we use the Amazon question/answer dataset (McAuley and Yang, 2016). The sentences are mostly informal and sometimes do not follow the correct grammatical structure. We utilize the answers that people post on the community question answering system as inputs to our model; hence, proposing an **opinion question generation** system which could be used as an interface to online forums helping users in browsing and querying them by making questions as suggestions.

In the subsequent section, we describe the related works to QG. The next section is on the task definition, followed by the demonstration of the model structure. After that, we discuss the experimental settings and at the end provide a thorough discussion of our results.

2 Related Work

After the first question generation shared task evaluation challenge (Rus et al., 2010), the question generation task has received a huge attention from the natural language generation community. Many of the traditional approaches involve human resources to create robust templates and then employing them to generate questions. For instance, Heilman and Smith (2010) approach is to overgenerate questions by some hand-written rules and then rank them using a logistic regression model. Labutov et al. (2015) benefit from a low-dimensional ontology for document segments. They crowdsource a set of promising question templates that are matched with that representation and rank the results based on their relevance to the source. Lindberg et al. (2013) employed a template-based approach while taking

Proceedings of The 11th International Natural Language Generation Conference, pages 152–158,
Tilburg, The Netherlands, November 5-8, 2018. ©2018 Association for Computational Linguistics

advantage of semantic information to generate natural language questions for on-line learning support. Chali and Hasan (2015) consider the automatic generation of all possible questions from a topic of interest by exploiting the named entity information and the predicate argument structures of the sentences.

Lately, more approaches have been presented that utilize the neural encoder-decoder architecture. Serban et al. (2016) address the problem by transducing knowledge graph facts into questions. They created a factoid question and answer corpus by using the Recurrent Neural Network architecture.

QG can also be combined with its complementary task, Question Answering (QA) for further improvement. Tang et al. (2017) consider QG and QA as dual tasks and train their relative models simultaneously. Their training framework takes advantage of the probabilistic correlation between the two tasks. QG has also entered other communities such as computer vision. Mostafazadeh et al. (2016) introduced the visual question generation task where the goal of the system is to create a question given an image.

One of the latest studies on the QG task has been conducted by Du et al. (2017). Their task is a QG on both sentences and paragraphs for the reading comprehension task, and they adopt an attention-based sequence learning model. Another recent work is by Yuan et al. (2017), they generate questions from documents using supervised and reinforcement learning.

In our work, we generate questions using community questions and answers and apply the encoder-decoder structure. To boost the performance of our system, we use attention and coverage mechanisms as suggested in See et al. (2017).

3 Task Formulation

Given an answer $A = (a_1, a_2, ..., a_N)$, we are going to generate a natural language question $Q = (q_1, q_2, ..., q_M)$, where its answer is embedded in A. Our goal is to find Q such that the conditional probability $p(Q|A)$ is maximized. We model $p(Q|A)$ as a product of word predictions:

$$p(Q|A) = \prod_{1}^{M} p(q_t|q_{1:t-1}, A)$$

This indicates that the probability of each q_t relies on the previously generated words and the input sentence A.

4 Model Structure

For modeling $p(Q|A)$, we use the simple RNN encoder-decoder architecture (Cho et al., 2014) with the global attentional model (Luong et al., 2015), which lets the decoder learn to focus on a particular range of the input sequence during the generation task. To improve upon this model, we apply coverage mechanism (See et al., 2017), which prevents the word repetition problem.

4.1 Encoder

An encoder network maps an input sequence into word vectors and then converts them into hidden states $b_1, ..., b_N$. In our case, the encoder is a two layer bidirectional LSTM network (Hochreiter and Schmidhuber, 1997). We concatenate the output of the forward hidden states $\overrightarrow{b_j}$ and the backward hidden states $\overleftarrow{b_j}$, namely, $b_j = [\overrightarrow{b_j}; \overleftarrow{b_j}]$ for input token j. This b_j is used later by the decoder to calculate the context vector c_t, which stores the relevant source-side information and simplifies the prediction of the next target word. c_t is computed as a weighted sum of b_i:

$$c_t = \sum_{i=1}^{N} a_t(i) b_i \qquad (1)$$

where a_t is an alignment vector and is calculated according to the general attention model:

$$a_t(i) = \frac{exp(h_t^T W_a b_i)}{\sum_j exp(h_t^T W_a b_j))} \qquad (2)$$

To initialize the decoder's hidden state, we concatenate the hidden states of the forward and the backward pass of the encoder.

4.2 Decoder

The decoder is a two layer unidirectional LSTM. It keeps a coverage vector s, which is the sum of the previous alignment vectors:

$$s_t = \sum_{t\prime=0}^{t-1} a_{t\prime}$$

It shows how much coverage each input word has received from the attention mechanism so far and it helps the mechanism to avoid attending to the same words again once they have been attended to initially (See et al., 2017). It should be mentioned that s_0 is a zero vector since nothing has been covered on the first time step. This coverage vector will be added to the source hidden states b_i:

$$b_i = \tanh(b_i + w_s s_t(i))$$

This b_i will be substituted in equations (1) and (2) where w_s is a parameter to be learned. This way, with the help of s_t, the attention mechanism always has a memory of its past decisions.

The decoder predicts the next word q_t given the context vector c_t and all the previously predicted words $\{q_1, ..., q_{t-1}\}$. We use a softmax layer to produce the predictive distribution:

$$p(q_t|q_{1:t-1}, A) = softmax(W_s\widetilde{h}_t)$$

$\widetilde{h}_t$ is the attentional hidden state which is calculated given the target hidden state h_t and the source context vector c_t:

$$\widetilde{h}_t = \tanh(W_c[c_t; h_t])$$

where W_s and W_c are learnable parameters. The hidden state at time step t of the decoder is generated by:

$$h_t = LSTM(q_{t-1}, h_{t-1})$$

where q_{t-1} is the previously generated word and h_{t-1} is the former hidden state.

Moreover, we use the input feeding approach (Luong et al., 2015), which informs the decoder which words were considered for the past alignments. We do this by concatenating the attentional hidden state $\widetilde{h}_t$ with the inputs at the next time steps.

4.3 Training and Generation

The training objective is to minimize the negative log-likelihood of the training corpus. Considering $S = \{(a_i, q_i)\}_1^{|S|}$ as our whole training data, we define the objective as:

$$J_t = \sum_{i=1}^{|S|} - \log p(q_i|a_i) \tag{3}$$

In addition to this primary loss function, it is required to introduce a coverage loss to penalize an overlap between the coverage vector and the attention distribution, which means attending to the same location multiple times.

$$covloss_t = \sum_i \min(a_t(i), s_t(i))$$

After being reweigted by some hyperparameter λ, this amount is added to equation (3):

$$J_t = \sum_{i=1}^{|S|} - \log p(q_i|a_i) + \lambda covloss_t$$

In the generation step, we utilize the beam search for the inference to maximize the conditional probability.

Since the size of our vocabulary is limited to a small number, many unknown words (*UNK*) will be generated during the inference. We substitute the (*UNK*) tokens with the words with the highest attention weight from the source sentence.

5 Experiments

5.1 Dataset

We use the Amazon question/answer dataset (McAuley and Yang, 2016). We set the minimum length of the questions to 4 tokens, including the question mark to filter out poorly structured sentences. The answers must be at least 10 tokens long. Moreover, we set the maximum length of the questions and the answers to 20 and 35 tokens, respectively. As there are many URLs in the dataset, we replace them with a *URL* token to reduce the vocabulary size. We lower-case the entire dataset and use the NLTK toolkit [1] for sentence tokenization. There can be many examples where the questions are not grammatically correct. People may just ask: “Waterproof ?”. The same problem occurs with the answers: the answer might be a single “Yes”. We use 80% of the dataset as the training set, and the rest is divided between the validation set and the test set. Table 1 shows the total number of examples in each dataset after removing very long or very short sentences from the training and the validation datasets.

	Train	Validation	Test
# pairs	233729	28969	70648

Table 1: Statistics of the dataset

[1] http://www.nltk.org

5.2 Experimental Setting

Our base model is from OpenNMT system (Klein et al., 2017), and we use the PyTorch [2] library, a deep learning framework that provides maximum flexibility and speed. It accelerates the computation on both CPU and GPU by a great amount, and the memory usage is extremely efficient in PyTorch compared to other options. We fix the size of the answer and the question vocabularies to 50k. Only the most frequent words are kept, and the rest are replaced with the *UNK* token. We set the word embedding dimension to 300 and we use *glove.840B.300d* (Pennington et al., 2014) as the pre-trained word embedding on both the encoder and the decoder sides. These embeddings are updated during training. The LSTM hidden unit size is set to 600 and we set the number of layers to 2. We employ the stochastic gradient descent (SGD) as the optimization method with an initial learning rate of 1.0 and halve the learning rate after 10 epochs. The training continues for 20 epochs with the batch size of 64 and dropout probability of 0.3. The hyperparameter λ that is used for weighting the coverage loss is set to 1[3]. The decoding is done using the beam search with the beam size of 5, and the generation is stopped when we reach the *EOS* token. In the end, we choose the model with the lowest perplexity on the validation set.

5.3 Baseline

We compare our model[4] to that of Du et al. (2017). We only experiment with their sentence-level model and run the same Amazon question and answer dataset on the system provided by the first author. We keep the source and target vocabulary size the same as ours, (i.e., 50k) and set the maximum and the minimum length of the questions and answers the same as our model. Everything else is left to the default values.

5.4 Automatic Evaluation Metrics

For evaluating our system automatically, we use three different evaluation metrics. The first one is BLEU (Papineni et al., 2002) that uses the n-gram similarity between a prediction and a set of references. We calculate BLEU score for unigrams and bigrams. The next one is METEOR (Denkowski and Lavie, 2014), which scores predictions by aligning them to ground truth sentences with the help of stemming, synonyms and paraphrases. The last evaluation metric is Rouge (Lin, 2004). It compares the generated sentences with the references based on n-gram. For this task, we use ROUGE$_L$, which reports the results based on the longest common subsequence. We use the evaluation package by Chen et al. (2015).

6 Results and Discussion

Table 2 shows the results of our system and the baseline. Our model improves the BLEU 1 score by at least 1.5 points. It also achieves a better result regarding the BLEU 2 and the METEOR whereas the ROUGE is lower than the baseline. If we consider the results reported in Du et al. (2017), we notice that the BLEU scores are much higher compared to our work. The reason is that they use the SQuAD dataset (Rajpurkar et al., 2016), which is a human-generated corpus. The sentences are well-structured, grammatically correct with fewer unnecessary punctuation and colloquialism. However, when working with the community-based question answering systems, the structure of sentences do not always follow the correct path. These sentences often contain useless information and symbols.

	Baseline	Our Model
BLEU 1	12.89	**14.67**
BLEU 2	6.95	**7.74**
METEOR	8.76	**9.43**
ROUGE$_L$	**25.91**	25.21

Table 2: BLEU 1-2, METEOR and ROUGE$_L$ scores on the test set. Bold numbers demonstrate the best performing system for each evaluation metric.

Another problem is that multiple questions can be generated from a single sentence. The system may generate a question which is correct both semantically and grammatically and also asks about accurate information in the sentence. However, if it is not the same as the ground-truth, the results will be affected.

Figure 1 shows some examples generated by our system and Du et al. (2017), where the coverage mechanism becomes useful and prevents the model from generating the same word 'material' twice.

[2] http://pytorch.org

[3] We also experimented with $\lambda = 2$ but did not find it to be helpful.

[4] https://github.com/Tina-19/Question-Generation

Answer 1: I really don't know, I did full size cupcakes, mini ones it would hold a ton!
GT Question: How many mini-cupcakes will this hold?
DSC: what size is it?
Ours: how many cupcakes will it hold?

Answer 2: Nothing out of the ordinary. just a simple screw driver. if I recall correctly, I think it may have came with the tools needed to assemble. good luck and congratulations
GT Question: What tools are required to assemble unit?
DSC: What is the assembly required?
Ours: what tools do I need to assemble this?

Answer 3: You can definitely still do pushups with the wraps on. The wraps just give extra support, they really don't impact your range of motion at all.
GT Question: Can I do pushups while wearing these wraps, or is the material too stiff?
DSC: Can you still use the material while wearing the material?
Ours: Can I do pushups while wearing these wraps?

Answer 4: I would go with a medium it fits well and when you adjust it with the helmet it's tight to the chin.
GT Question: What size to buy for 14 yr old 125lb and 5'5?
DSC: I'm a woman with a small head, what size should I get?
Ours: What size should I get for a child who is 5'6"?

Answer 5: There's the ability to forward the bp measurement information via email to friends, family and doctors so I assume that once it's been sent an email you can print - it however I haven't tested this functionality yet. At the very least when you bring up the bp readings on your screen you can do a screen capture and then print that screen capture.
GT Question: Is it possible to print the BP readings?
DSC: What is the difference between the BP and the BP?
Ours: How do you print from the BP?

Figure 1: Examples of generated questions: ground truth (GT), Du et al. (2017) (DSC) and our model, with their answers.

7 Human Evaluations

To further assess the performance of our system, we performed human evaluations on the results. Three English-speaker students were asked to give a score from 1 (very poor) to 5 (very good) to the questions generated from both systems according to two criteria: **syntactic correctness** and **relevance**. Syntactic correctness indicates the grammaticality and the fluency and relevance demonstrates whether the question is meaningful and related to the sentence it is generated from. The three assessors performed the evaluations on 100 randomly selected question and answer pairs from the results. The comparison of human evaluations between our system and the Du et al. (2017) model is shown in Table 3. Bold numbers demonstrate the best performing system for each evaluation criteria, and we see that our system outperforms the Du et al. (2017) model on both criteria.

	Baseline	Our Model
Syntactic correctness	4.4	**4.52**
Relevance	2.93	**3.37**

Table 3: Human evaluation results for the syntactic correctness and relevance between our model and Du et al. (2017).

8 Conclusion

In this work, we presented a sequence to sequence learning model to address the opinion question generation task. We showed the training process using the global attention and applied the coverage mechanism to improve the model. We took advantage of community-based question answering systems which contain informal speech and its sentences do not always follow grammatical rules. Experimental results show an improvement in the automatic evaluation metrics as well as the human evaluations compared to the baseline system.

Acknowledgements

We would like to thank the anonymous reviewers for their useful comments. The research reported in this paper was conducted at the University of Lethbridge and supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada discovery grant and the University of Lethbridge.

References

Yllias Chali and Sadid A. Hasan. 2015. Towards topic-to-question generation. *Computational Linguistics*, 41(1):1–20.

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 nau, 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*, pages 1724–1734. Association for Computational Linguistics.

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*, pages 376–380, Baltimore, Maryland. Association for Computational Linguistics.

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

Arthur C. Graesser and Natalie K. Person. 1994. Question asking during tutoring. *American Educational Research Journal*, 31(1):104–137.

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*, pages 609–617.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*.

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*, pages 889–898, Beijing, China. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

David Lindberg, Fred Popowich, John C. Nesbit, and Philip H. Winne. 2013. Generating natural language questions to support learning on-line. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 105–114, Sofia, Bulgaria. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics.

Julian McAuley and Alex Yang. 2016. Addressing complex and subjective product-related queries with customer reviews. In *Proceedings of the 25th International Conference on World Wide Web*, pages 625–635.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

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)*, pages 1532–1543. Association for Computational Linguistics.

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

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*, pages 251–257. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083. Association for Computational Linguistics.

Iulian V. Serban, Alberto García-Durán, Çalar Gülçehre, 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*, pages 588–598, Berlin, Germany.

Duyu Tang, Nan Duan, Tao Qin, and Ming Zhou. 2017. Question answering and question generation as dual tasks. *arXiv preprint*, arXiv:1706.02027.

Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William Cohen. 2017. Semi-supervised qa with generative domain-adaptive nets. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1040–1050, Vancouver, Canada. Association for Computational Linguistics.

Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Saizheng Zhang, Sandeep Subramanian, and Adam Trischler. 2017. Machine comprehension by text-to-text neural question generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 15–25, Vancouver, Canada. Association for Computational Linguistics.