Asking the Crowd: Question Analysis, Evaluation and Generation for Open Discussion on Online Forums

Zi Chai[1,2], **Xinyu Xing**[1,2], **Xiaojun Wan**[1,2], **Bo Huang**[3]
[1]Institute of Computer Science and Technology, Peking University
[2]The MOE Key Laboratory of Computational Linguistics, Peking University
[3]Zhihu Institute
{chaizi, xingxinyu, wanxiaojun}@pku.edu.cn, huangbo@zhihu.com

Abstract

Teaching machines to ask questions is an important yet challenging task. Most prior work focused on generating questions with fixed answers. As contents are highly limited by given answers, these questions are often not worth discussing. In this paper, we take the first step on teaching machines to ask open-answered questions from real-world news for open discussion (openQG). To generate high-qualified questions, effective ways for question evaluation are required. We take the perspective that the more answers a question receives, the better it is for open discussion, and analyze how language use affects the number of answers. Compared with other factors, e.g. topic and post time, linguistic factors keep our evaluation from being domain-specific. We carefully perform variable control on 11.5M questions from online forums to get a dataset, OQRanD, and further perform question analysis. Based on these conclusions, several models are built for question evaluation. For openQG task, we construct OQGenD, the first dataset as far as we know, and propose a model based on conditional generative adversarial networks and our question evaluation model. Experiments show that our model can generate questions with higher quality compared with commonly-used text generation methods.

1 Introduction

Teaching machines to ask questions from given corpus, i.e. question generation (QG), is an important yet challenging task in natural language processing. In recent years, QG has received increasing attention from both the industrial and academic communities due to its wide applications. Dialog systems can be proactive by asking users questions (Wang et al., 2018), question answering (QA) systems can benefit from the corpus produced by a QG model (Duan et al., 2017), education (Heilman and Smith, 2010) and clinical (Weizenbaum et al., 1966; Colby et al., 1971) systems require QG as well.

We can divide all questions into two categories. *Fixed-answered* questions have standard answers, e.g. "who invented the car? (Karl Benz)". In contrast, different people may have distinct answers over *open-answered* questions like "what do you think of the self-driving car?". Most prior work about QG (QA) aimed to generate (answer) fixed-answered questions. As questions are targeting on answers which are certain spans of given corpus, they are always not worth discussing. Nowadays, with the help of online QA forums (e.g. Quora and Zhihu[1]), open-answered questions can greatly arouse **open discussion** that helps people under different backgrounds to share knowledge and ideas (high-qualified questions can help to attract more visitors for QA forums as well). This kind of questions are also useful for many tasks, e.g. making dialog systems more proactive.

In this paper, we focus on generating open-answered questions for open discussion, i.e. the openQG task. To make our model useful in practice, we generate questions from real-world news which are suitable for arousing open discussion. As far as we know, **no research** has focused on this task before due to the two difficulties:

- To generate high-qualified questions (for open discussion), we need to perform question evaluation, which is rather challenging.
- Questions in most existed QG (QA) datasets, e.g. SQuAD (Rajpurkar et al., 2016), are fixed-answered thus not suitable for openQG.

It is worth mentioning that a good question evaluation metric is not only a necessity to compare

[1] Quora and Zhihu are large-scale online English, Chinese QA forums, respectively (https://www.quora.com/, https://www.zhihu.com/).

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5032–5046
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

different models, but can also throw light on the text generation process, e.g. acting as the reward function through reinforcement learning. Based on the perspective that the more answers a question receives, the higher quality it has for open discussion, we analyze how **language use** affects the number of answers. Compared with other factors, e.g. the topic and post time, focusing on language use can keep our evaluation from being domain-specific. To this end, we carefully perform variable control on 11.5M online questions from Zhihu and build the "open-answered question ranking dataset (OQRanD)", containing 22K question pairs (questions in each pair **only** differ in language use). Based on OQRanD, we reach to some interesting conclusions on how linguistic factors affects the number that a question receives, and further build question evaluation models.

After building our linguistic-based question evaluation model, we propose a QG model based on conditional generative adversarial network (CGAN). During the adversarial training process, we perform reinforcement learning to introduce information from the evaluation model. This architecture was **not used** in QG before as far as we know, and experiments show that our model gets better performance compared with commonly-used text generation methods in the quality of generated questions. All the experiments are performed on the "open-answered question generation dataset (OQGenD)" we build, which contains 20K news-question pairs. It is **the first** dataset for openQG to the best of our knowledge.

Above all, the main contributions of this paper are threefold:

- We propose the openQG task, and build OQGenD, OQRanD from 11.5M questions for generating and evaluating questions.

- We study how language use affects the number of answers a question receives, and draw some interesting conclusions for linguistic-based question evaluation.

- We propose a model based on CGAN and our question evaluation model, which outperforms commonly-used text generation models in the quality of generated questions.

In this paper, the two datasets OQRanD and OQGend are available at `https://github.com/ChaiZ-pku/OQRanD-and-OQGenD`.

2 Related Work

2.1 Question Evaluation

Question evaluation is a rather challenging task. Automatic evaluation metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004) and METEOR (Lavie and Agarwal, 2007) were widely used to measure n-gram overlaps between generated questions and ground truth questions, however, they are far from enough since we cannot list all possible ground truth questions in openQG. To this end, we need to develop specific evaluation metrics for questions. Some researches (Heilman and Smith, 2010; Figueroa and Neumann, 2013) directly trained question ranking (QR) models via supervised learning, and used it to perform evaluation. However, these models are always domain-specific and not interpretable since we cannot tell what makes a question get a high (low) score. Rao and Daumé III (2018) took a step further, and pointed out that a good question is one whose expected answer will be useful. By using the "expected value of perfect information", they proposed a useful evaluation model. However, our task **significantly differs** from it in two aspects: first, there is no correct answer for open-answered questions thus it is hard to tell which answer is "useful". Second, the goal of openQG is to arouse open discussions instead of "solving a problem".

Intuitively, a good question evaluation metric should be interpretable and keeps away from being domain-specific. To this end, we first analyze how language use affects the number of answers, and then build evaluation models based on these conclusions. There are some researches (Guerini et al., 2011; Danescu-Niculescu-Mizil et al., 2012; Guerini et al., 2012; Tan et al., 2014) about how language use affects the reaction that a piece of text generates, but we are **the first** to focus on questions as far as we know.

2.2 Question Generation

QG was traditionally tackled by rule-based approaches (Heilman and Smith, 2010; Lindberg et al., 2013; Mazidi and Nielsen, 2014; Hussein et al., 2014; Labutov et al., 2015). In recent years, neural network (NN) approaches have taken the mainstream. Du et al. (2017) pioneered NN-based QG by using Seq2seq models (Sutskever et al., 2014). Many researches have tried to make it more suitable for QG tasks since then, including using answer position features (Zhou et al., 2017), pointer mechanism (Kumar et al., 2018a; Zhao et al., 2018), etc. Adding more constraints, e.g. controlling the topic (Hu et al., 2018) and difficulty (Gao et al., 2018) of QG, or combining it with QA (Duan et al., 2017; Wang et al., 2017; Tang et al., 2017) have also been studied. Recently, using adversarial training and reinforcement learning (Yuan et al., 2017; Kumar et al., 2018b; Yao et al., 2018) have become a new trend. As far as we know, the CGAN model we proposed has **not used before**. Besides, most prior researches aimed to generate fixed-answered questions, and we are **the first** to propose openQG task to the best of our knowledge.

It is worth mentioning that though we only focus on text-based QG, we can also generate questions from images, i.e. visual question generation (Ren et al., 2015; Fan et al., 2018) and knowledge graphs (Serban et al., 2016; Elsahar et al., 2018) as well.

3 Question Analysis and Evaluation

In this section, we deal with question analysis and evaluation. We first perform variable control and build OQGenD. After that, we analyze how language use affects the number of answers a question receives. Based on these conclusions, we further build question evaluation models.

3.1 Construction of OQRanD

The number of answers a question receives is affected by many factors. As pointed out by a number of prior researches, there are four dominated variables: **topic**, **author**, **time** and **language use**. In other words, we should control the first three variables to study the effect of language use. We perform our analysis based on an in-house dataset from Zhihu. There are 11.5M open-domain questions, and the following information is also provided for each question: the post time, the author (user ID), the author's followers and followees, the manually-tagged topics, the number of answers, viewers and followers.

Although we mainly focus on the number of answers, the counts of viewers and followers of the question are also interesting. Especially, if a question receives more answers, can we expect it to be viewed and followed by more people as well? To figure it out, we perform correlation analysis using the Pearson correlation coefficient (PCC) (Lee Rodgers and Nicewander, 1988). PCC is a measure of the linear correlation between two random variables. It is a real number between [-1, 1], where 1 means there is a total positive linear correlation, 0 means no linear correlation exixts, and -1 means there is a total negative linear correlation. PCC between the number of answers and viewers is 0.93, and that number between the number of answers and followers is 0.86. So a question with more answers can always attract more visitors and followers.

As for variable control, we first focus on **topic**. Since each of the 11.5M questions has one of the 37 manually-tagged topics (all topics are listed in the appendix), we divide them into 37 subsets, and further extract question-pairs in each subset independently. In each pair, we want the topics of two questions as close as possible. Since questions are short texts (often about 10 words), topics are greatly reflected by nouns. We measure topic-similarity for questions q_1, q_2 by:

$$TS(q_1, q_2) = \frac{\text{\# nouns in both } q_1 \text{ and } q_2}{\text{\# nouns in } q_1 + \text{\# nouns in } q_2} \quad (1)$$

where "#" means "the number of". The larger $TS(q_1, q_2)$ for q_1, q_2, the closer they are in topics. We set a boundary μ, and filter out question pairs whose $TS(q_1, q_2) < \mu$. A number of values for μ is tried, and we finally choose $\mu = 0.3$ since the topics of (q_1, q_2) are already close enough without discarding too much data. Finally, we get 24.2M topic-controlled (TC) question pairs.

Based on TC pairs, we further control the effect of **authors**. Since users with more followers are expected to get more responses, we need to eliminate the effect of their social network. To do so, we collect all active users provided by Zhihu and build a "follower network". In this network, each user is a node, and there is an edge from A to B if user A follows user B. We run PageRank algorithms (Page et al., 1999) on the network, and get a PageRank value for each user (real values are rounded to integers). By excluding TC pairs whose authors do not have the same PageRank value, we get 10.8M topic- and author-controlled (TAC) question pairs.

Controlling the effect of **time** is rather complex, since few questions are posted at exactly the same time. An earlier question may benefit from "first-move advantage" (Borghol et al., 2012), but a later question might be preferred because the earlier can become "stale" (Tan et al., 2014). For a TAC pair (q_1, q_2), we use (n_1, n_2) to denote the

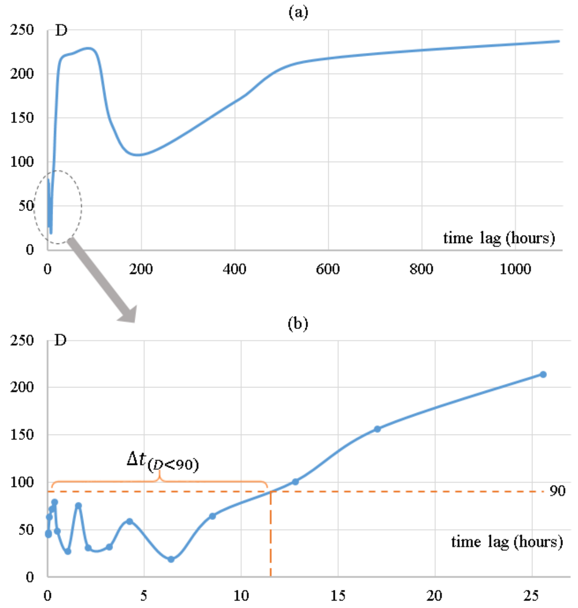

Figure 1: The effect of time lag (Δt) on D.

number of their answers, and (t_1, t_2) to show their posted times. The idea is: we first study how time factors affect the number of answers, i.e. how $\Delta t = |t_1 - t_2|$ affects $\Delta n = |n_1 - n_2|$. After that, we can find if certain Δt has small effects. By picking TAC pairs with such Δt, the effect of time can be greatly reduced.

To study how Δt affects Δn, we should leave Δt as the only variable, i.e. control the effect of language use in TAC pairs. To do so, we measure the distance between q_1 and q_2 by normalized edit distance:

$$d(q_1, q_2) = \frac{edit(q_1, q_2)}{max(len(q_1), len(q_2))} \quad (2)$$

where $edit(q_1, q_2)$ is the edit distance, and $len(\cdot)$ is the length of a question. The smaller $d(q_1, q_2)$ between q_1 and q_2, the more similar they are in language use. We further rank all TAC pairs by d values from small to large, and pick up the first 2% pairs to get 217K topic-, author- and language-controlled (TALC) question pairs. Now that Δt is the only difference, the smaller effect it has, the smaller Δn is expected. The number of TALC pairs decreases exponentially with the growth of Δt. As pointed out by Tan et al. (2014), directly computing $E(\Delta n|\Delta t)$ is not reliable since the estimate will be dominated by TALC pairs with small Δn. Instead, we should use the deviation estimate:

$$D = \sum_{0 \leq n_1 \leq 9} |\hat{E}(n_2|n_1) - n_1| \quad (3)$$

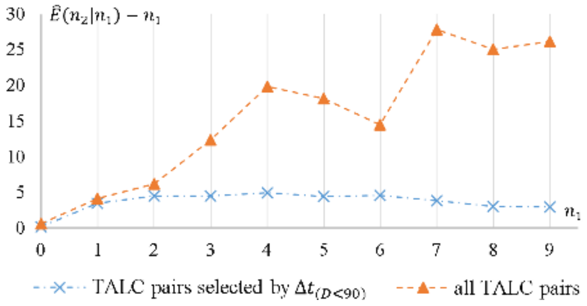

Figure 2: D under different n_1 (the smaller, the better).

where $\hat{E}(n_2|n_1)$ is the average n_2 over question pairs whose q_1 has n_1 answers, and TALC pairs whose $n_1 > 9$ are not considered since the number is too few, making the results less reliable.

In Figure 1(a), we show how D varies with Δt (a smaller effect of Δt makes D closer to 0). As we can see, D is rather small when Δt is close to 0, which is in accordance with common sense. As Δt grows, D increases sharply, which is largely caused by the "first move advantage" described in (Borghol et al., 2012). Although D decreases when Δt is about 100 hours (we think the main reason is: earlier questions starts to become "stale"), it is not so small as before. When Δt is about 200 hours (the later questions also starts to become "stale"), D increases again and maintains at a high level. Figure 1(b) shows the case when Δt is close to 0.

As mentioned above, if we control Δt to make D rather small, the effect of time will be greatly reduced. However, we may filter out too many data if making Δt too close to 0. Intuitively, 90 seems like a good upper-bound, and we use $\Delta t_{D<90}$ to denote the time interval composed by all Δt that make $D < 90$. To further test this upper-bound, we pick out TALC pairs whose $\Delta t \in \Delta t_{D<90}$, and compute the deviation $|E(n_2|n_1) - n_1|$ under different n_1 to get Figure 2 (in contrast, we also show the case when Δt is not controlled). As we can see, by choosing pairs whose $\Delta t \in \Delta t_{D<90}$, we can greatly reduce deviations. Since $|E(n_2|n_1) - n_1| < 5$ under each n_1, we can further eliminate the remaining time-effect by enlarging Δn. Based on thse conclusions, we perform time-control on all TAC pairs by choosing pairs whose $\Delta t \in \Delta t_{D<90}$ and $\Delta n > 20$ (20 is much larger than 5). To study the effect of language use, we want q_1, q_2 not so close. So we further discard the remaining pairs whose $d(q_1, q_2) < 0.6$, and get 22K question pairs to build OQRanD.

notation	t-test efficacy
↑↑↑↑, ↓↓↓↓	$p \leq 0.0001$
↑↑↑, ↓↓↓	$p \leq 0.001$
↑↑, ↓↓	$p \leq 0.01$
↑, ↓	$p \leq 0.05$

Table 1: The number of arrows and t-test efficacy.

length	↓↓↓↓	puctuation	↓↓↓↓
noun	↓↓↓↓	1^{st} ppron	↓↓↓↓
verb	↑↑↑↑	2^{nd} ppron	↑↑↑↑
adjective	↓↓↓	3^{rd} ppron	↓↓↓↓
adverb	↑↑↑↑	please-word	↓↓↓↓
preposition	↓↓↓	positive-word	↑↑↑↑
pronoun	-	negative-word	-
quantifier	↑↑↑	sentiment-word	↑
numeral	-	common-word	↑

Table 2: Significance tests on text features. The "ppon" denotes for "personal pronoun".

3.2 The Effects of Language Use

To show how language use affects the number of answers that a question receives, we perform significant tests on different linguistic features. The one-sided paired t-test with Bonferroni correction (for multiple comparisons) is adopted. For significant levels, we set $\alpha = .05, .01, .001, .0001$, which correspond with the number of arrows (Table 1). The direction of arrows show how the feature affects the number of answers: up arrows (↑) indicate that a large feature-value (e.g. a longer length, a higher perplexity) can lead to more answers, and down arrows (↓) means small feature values are preferred. Here are some interesting conclusions [2]:

Ask concise questions. The basic sanity check we perform is the length of questions. Table 2 indicates that questions with less words tend to get more answers. This is in accordance with Simmons et al. (2011) which shows that short version of memes are more likely to become popular. In contrast, Tan et al. (2014) found that longer versions of tweets are more likely to be popular. This indicates that attracting more answers is different from making a blog retweeting by more people.

Ask one thing a time and make it vivid. What kinds of words can help to get more answers? We test the proportion of different parts of speech (POS) that occurs (proportions are better than word counts since they can eliminate the effect of length). As Table 2 suggests, using less nouns, adjectives and prepositions is helpful. As nouns are often topic words (occurred with adjectives and prepositions), it is better to contain less topics and ask one thing a time. On the other hand, it is better to use more verbs and adverbs to make the question vivid. Besides, using less punctuation helps (this often leads to more concise questions).

data for training language models		ppl (word n-grams)	ppl (POS n-grams)
random sampled questions	3-gram	↓↓↓↓	↓↓↓↓
	2-gram	↓↓↓↓	↓↓↓↓
	1-gram	↓↓↓↓	↓↓↓↓
most answered questions	3-gram	↓↓↓↓	↓↓↓↓
	2-gram	↓↓↓↓	↓↓↓↓
	1-gram	↓↓↓↓	↓↓↓↓
news headlines	3-gram	↓↓↓↓	↓↓↓↓
	2-gram	↓↓↓↓	↑↑↑↑
	1-gram	↓↓↓↓	↑↑↑↑

Table 3: Significance tests on LM-based features. ppl stands for perplexity.

Interact with readers naturally. We check the proportions of personal pronouns (ppron), and find it helps to be interactive by using more second ppron, e.g. 你认为 (what do *you* think of). We also check the proportion of please-words, e.g. 请教 (could you please answering...). As Table 2 indicates, we should not use too many honorifics. Just interact with others naturally as if we are talking to our close friends.

Positive words help. Can we get more answers by picking words with sentiments? We check the occurrence of positive and negative words based on a word emotional polarity dictionary, NTUSD [3]. As shown in Table 2, more sentiment words can help, especially positive words.

Use familiar expressions. Distinctive expressions may attract attention, but using "common language" can make a question better understood. Intuitively, if more commonly-used words occurs, a question is easier to read. To this end, we collect 4K words with the highest frequency from OQRanD and measure their occurrence. Table 2 shows that it is better to use common words and make the question familiar.

[2] More details, (e.g. how we trained the language models) are listed in the appendix).

[3] `https://github.com/data-science-lab/sentimentCN/tree/master/dict`.

Model	Accuracy	
	traditional	traditional+ours
LR	78.61%	82.33%
RF	81.70%	87.74%
SVM	79.02%	**87.96%**
RNN	74.68%	-
CNN	**83.18%**	-

Table 4: Results for QR task. For LR (logistic regression), RF (random forest) and SVM (support vector machine), "traditional" means n-gram word and POS features. "+ours" means adding the 33 features that pass the significant test in Table 2 and Table 3. For LSTM (long-short term memory network) and CNN (convolution neural network), "traditional" means word and POS embeddings.

In addition, we randomly sample 134K questions that are not appeared in OQRanD to build six language models (LMs) based on 1, 2, 3 gram word and POS features, respectively. Table 3 indicates that questions with smaller perplexity (i.e. more familiar) are always better.

Imitate good questions. Since a number of questions have already aroused a large range of open discussion, can we get more answers by imitating them? We pick 80K questions that are not appeared in OQRanD with the highest answer number as "good questions" and train six LMs (similar to above). Table 3 shows that the less perplexity a question gets, the more answers it arouses. In conclusion, imitating good questions helps. We also explore if news headlines are worth imitating. On one hand, they are carefully-written concise texts. On the other hand, as pointed out by Wei and Wan (2017), a lot of Chinese news headlines are intentionally written to be attention-getting. From Table 3, it turns out that imitating their word use is useful.

3.3 Question Evaluation Model

Based on OQRanD and our conclusions about how language use affects the answer that a question receives, we can train models to predict which question can receive more answers in each pair. Since questions in the same pair only differ in language use, models based on OQRanD can concentrate on linguistic facts to avoid being domain-specific.

Given pair (q_1, q_2), we label it as "1" if $n_1 > n_2$, otherwise we use label "0". In this way, our task turns into a binary classification task. We further train a model F_s which inputs a question and outputs a score. The larger $F_s(\cdot)$, the more answer is expected. By comparing $F_s(q_1), F_s(q_2)$, we can make the final prediction. Although we can also use both q_1, q_2 as inputs and train a model that directly outputs label 0 or 1, using F_s on q_1, q_2 respectively is more flexible when we need to rank more than two question. Besides, F_s can be directly used for getting rewards during the reinforcement QG process.

We use several models as F_s, and perform training based on the hinge loss. Table 4 shows the accuracy of different models (hyper-parameters and training details are provided in the appendix). When features in Section 3.2 are not used, the CNN model gets the best performance, which is not surprised. However, adding these features greatly improves the performance of all statistical models, making SVM and RF significantly surpass CNN. This illustrates the importance of linguistic factors.

4 Question Generation

In this section, we perform openQG. We construct OQGenD, the first dataset for openQG as far as we know, and propose a model based on CGAN. Especially, we use the question evaluation model based on OQRanD to introduce prior knowledge. Finally, we perform experiments and use multiple evaluation metrics (including our linguistic-based model) and reach to the conclusions.

4.1 Construction of OQGenD

Since real-world news are suitable for arousing open discussion, we built OQGenD from news and open-answered questions. We crawled news (published in the last three years) from Tencent News[4], and performed data cleaning (removing non-textual components and filtering out redundant data) to get 59K news at last. To make questions in OQGenD suitable for open discussion, we ranked the 11.5M questions mentioned in Section 3.1 by their number of answers from large to small and picked the first half (576K).

To match news and questions, we first used automatic ways to find a "candidate dataset" and then performed human labeling to build our final OQGenD dataset. To get the candidate dataset, three heuristic unsupervised methods were used to compute the distance between a piece of news

[4] `https://news.qq.com/`. It is one of the largest social media company in China

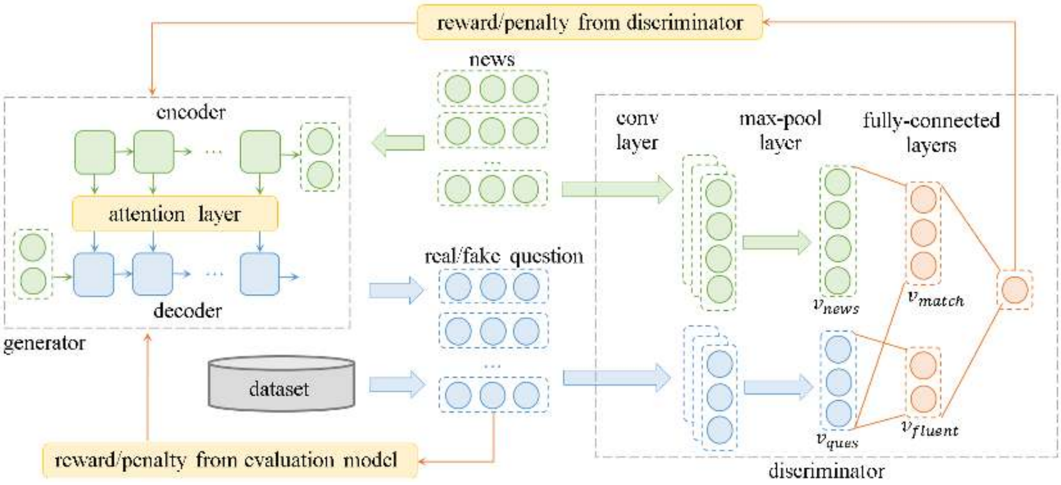

Figure 3: Architecture of our model.

and a question: (1) **term frequency-inverse document frequency (tf-idf)**, which first extracted 5 (10) key words from each question (news) by tf-idf values, and then measured distances by the number of intersected key words; (2) **cosine distance**, which is based on the bag-of-words model; (3) **weighted averaged word embeddings**, which was proposed by Arora et al. (2016). It first computed a weighted average of the word vectors in the sentence and then performed a "common component removal". For each piece of news, we picked out questions with the smallest two distances under each method.

We further hired five native speakers to label the candidate dataset. An NQ-pair was preserved only if it was appropriate for a human to raise the question given the piece of news. In other words, the question should be related to the given news while not mentioning extra information. In case that too many NQ-pairs were discarded, we allowed human labelers to perform two kinds of modifications on each question to preserve more data. First, we allowed them to modify the question in an NQ-pair by at most two entities, e.g. change it from "马克龙是怎样一个人？(What is Macron like?)" to "特朗普是怎样的一个人(What is Trump like?)". Second, we allowed them to use a meaningful substring to replace the original question. We ensured that each NQ-pair was labeled by three people, and it was preserved in OQGenD only if all of them agreed. In this way, we got 20K NQ-pairs. Among these pairs, there were 9K news, each corresponding with more than one questions. The average word numbers in each piece of news, question were 508, 12, respectively.

4.2 Model

As shown in Figure 3, our model is composed by a generator $G_{\boldsymbol{\theta}}$ and a discriminator $D_{\boldsymbol{\phi}}$. $G_{\boldsymbol{\theta}}$ outputs a question $\hat{Y} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_n\}$ from given news $X = \{x_1, x_2, ..., x_m\}$. It is a Seq2seq network with the attention mechanism (Luong et al., 2015). Both encoder and decoder are GRU (Chung et al., 2014) networks. $D_{\boldsymbol{\phi}}$ takes an NQ-pair (X, Y_D) as input, and predicts how likely it comes from real-world dataset. First, it embeds the X, Y_D into $\boldsymbol{v}_{news}, \boldsymbol{v}_{ques}$ respectively by two CNNs similar to Zhang and Wallace (2015). Based on the two representations, it computes

$$\begin{aligned} \boldsymbol{v}_{match} &= \boldsymbol{W}_m\,[\boldsymbol{v}_{news}; \boldsymbol{v}_{ques}] + \boldsymbol{b}_m \\ \boldsymbol{v}_{fluent} &= \boldsymbol{W}_f\,\boldsymbol{v}_{ques} + \boldsymbol{b}_f \end{aligned} \tag{4}$$

where $[\boldsymbol{v}_{news}; \boldsymbol{v}_{ques}]$ is the concatenation of the two vectors $\boldsymbol{v}_{news}, \boldsymbol{v}_{ques}$, and $\boldsymbol{W}_m, \boldsymbol{W}_f, \boldsymbol{b}_m, \boldsymbol{b}_f$ are parameters of our model. We expect $\boldsymbol{v}_{match}$ to measure if the question matches the news, and v_{fluent} to measure if the question is fluent enough (like human-written questions). The final prediction $D_{\boldsymbol{\phi}}(X, Y_D)$ is computed by

$$D_{\boldsymbol{\phi}}(X, Y_D) = \sigma(\boldsymbol{W}_{proj}\,[\boldsymbol{v}_{match}; \boldsymbol{v}_{fluent}] + \boldsymbol{b}_{proj}) \tag{5}$$

where σ is the *sigmoid* function and $\boldsymbol{W}_{proj}, \boldsymbol{b}_{proj}$ are parameters. As we can see, both $G_{\boldsymbol{\theta}}(X)$ and $D_{\boldsymbol{\phi}}(X, Y_D)$ are conditioned on X, thus our model can be viewed as a special type of CGAN (Mirza and Osindero, 2014), which provides more control to make generated questions closely related to input news.

Algorithm 1 Training process.

```
Input: NQ-pairs $(X, Y)$ from OQGenD; Generator $G_{\boldsymbol{\theta}}$; Discriminator $D_{\boldsymbol{\phi}}$; Evaluator $Q$;
Output: Well-trained generator.
Initialize $G_{\boldsymbol{\theta}}, D_{\boldsymbol{\phi}}$ ($Q$ is frozen);
Pre-train $G_{\boldsymbol{\theta}}$ on $(X, Y)$ by MLE;
repeat
    for d-steps do
        Sample $\hat{Y} \sim G_{\boldsymbol{\theta}}(\hat{Y}|X)$;
        Use $X, Y, \hat{Y}$ to generate fake NQ-pairs $(X_f, Y_f)$;
        Train $D_{\boldsymbol{\phi}}$ on real NQ-pairs $(X, Y)$ and fake NQ-pairs $(X_f, Y_f)$ by Eq. 6;
    end for
    for g-steps do
        Sample $\hat{Y} \sim G_{\boldsymbol{\theta}}(\hat{Y}|X)$;
        Compute rewards for $\hat{Y}$ by Eq. 10;
        Update $G_{\boldsymbol{\theta}}$ on $(X, \hat{Y})$ by Eq. 9;
    end for
until $G$, $D$ converge
```

4.3 Adversarial Training

The training process of GAN is formalized as a game in which the generative model is trained to generate outputs to fool the discriminator (Goodfellow et al., 2014). For our model, the training process is described in algorithm 1.

Before adversarial training, we pre-train $G_{\boldsymbol{\theta}}$ by maximizing the log probability of a question Y given X (X, Y come from OQGenD), i.e. Maximum Likelihood Estimate (MLE), as described in Sutskever et al., 2014. This is helpful for making the adversarial training process more stable. Besides, the parameters of our question evaluation model Q is frozen during the whole process.

We iteratively perform d-steps and g-steps to train $D_{\boldsymbol{\phi}}$, $G_{\boldsymbol{\theta}}$ respectively during the adversarial traing process. In d-steps, we fix the parameters of $G_{\boldsymbol{\theta}}$, and the inputs for $D_{\boldsymbol{\phi}}$ are three-folds: (1) NQ-pairs (X, Y) from OQGenD. (2) News and questions generated by $G_{\boldsymbol{\theta}}$, i.e. $(X, \hat{Y})$. (3) Unmatched NQ-pairs created from OQGenD. We label "real data" (1) as "1"; and regard both (2), (3) as "fake data" with label "0". It is worth mentioning that the unmatched NQ-pairs are used to keep $D_{\boldsymbol{\phi}}$ from only focusing on the questions. To train $D_{\boldsymbol{\phi}}$, we minimize the objective function:

$$\begin{aligned} J_D(\boldsymbol{\phi}) = &-\mathbb{E}_{(X,Y)\sim P_{\text{real data}}} \log D_{\boldsymbol{\phi}}(X, Y) \\ &-\mathbb{E}_{(X,Y)\sim P_{\text{fake data}}} \log(1 - D_{\boldsymbol{\phi}}(X, Y)) \end{aligned} \tag{6}$$

Since text-generation is a discrete process, we cannot directly use $D_{\boldsymbol{\phi}}(X, \hat{Y})$ to update $\boldsymbol{\theta}$ in $G_{\boldsymbol{\theta}}$. A commonly-used idea (Yu et al., 2017; Li et al., 2017) is to train $G_{\boldsymbol{\theta}}$ based on policy gradient (Sutton et al., 2000). In this case, $G_{\boldsymbol{\theta}}$ is regarded as a policy network. At time-step t, state s_t is the generated text $\hat{Y}_{[1:t]}$, and action a_t is generating the next word $\hat{y}_{t+1}$ with a probability $\pi_G(a_t|s_t) = p_G(\hat{y}_{t+1}|\hat{Y}_{[1:t]}, X)$. To get reward r_t, we perform Monte-Carlo search, i.e. sample $\hat{Y}_{[1:t]}$ into a complete sentence $\hat{Y}_{MC}$ for k times, and perform:

$$r_t = \frac{1}{k}\sum_{i=1}^{k} D_{\boldsymbol{\phi}}(\hat{Y}_{MC}^{(i)}, X) \tag{7}$$

After getting r_t, $\boldsymbol{\theta}$ is updated by minimizing

$$J_G(\boldsymbol{\theta}) = -\mathbb{E}[\sum_t r_t \cdot \log \pi(a_t|s_t)] \tag{8}$$

We can also change Eq 8 into a penalty-based version:

$$\begin{aligned} J'_G(\boldsymbol{\theta}) &= \mathbb{E}[\sum_t (1 - r_t) \cdot \pi(a_t|s_t)] \\ &= J_G(\boldsymbol{\theta}) + \mathbb{E}[\sum_t \pi(a_t|s_t)] \end{aligned} \tag{9}$$

where $\mathbb{E}[\sum_t \pi(a_t|s_t)]$ can be viewed as a regularization term. It forces the generator to prefer a smaller $\pi(a_t|s_t)$. In this way, it can generate more diversified results.

Since we have already trained a question evaluation model $F_s(\cdot)$ in Section 3.3, we can use:

$$r_t = \frac{1}{k}\sum_{i=1}^{k}(\gamma D_{\boldsymbol{\phi}}(\hat{Y}_{MC}^{(i)}, X) + (1-\gamma)F_s(\hat{Y}_{MC}^{(i)})) \tag{10}$$

to replace Eq. 7. In Eq. 10, we add prior knowledge about "how language use affects the number of answers" into the adversarial training process through reinforcement learning, and expect the linguistic affects that we have discovered can throw light on the text generation process.

4.4 Experiments

We choose several typical text-generation models as baselines. We apply a Seq2seq model similar to Du et al. (2017), and use a CopyNet similar to Kumar et al. (2018b). As adversarial training has become a new trend in QG, we also adopt the SeqGAN proposed by Yu et al. (2017) and SentiGAN by Wang et al. (2018). For our model, the "vanilla"

Models	BLEU				ROUGE$_L$	METEOR	F_s (SVM)
	1	2	3	4			
Seq2seq	$36.35^{*}_{\diamond}$	$20.25^{*}_{\diamond}$	$14.90^{*}_{\diamond}$	$13.22^{*}_{\diamond}$	$36.72^{*}_{\diamond}$	$21.57^{*}_{\diamond}$	$-2.28_{\diamond}$
CopyNet	$37.89^{*}_{\diamond}$	$21.09^{*}_{\diamond}$	$15.77^{*}_{\diamond}$	$14.07^{*}_{\diamond}$	$38.05^{*}_{\diamond}$	$22.63^{*}_{\diamond}$	$-1.80^{*}_{\diamond}$
SeqGAN	$38.51_{\diamond}$	$22.29_{\diamond}$	$16.97^{*}_{\diamond}$	$14.92^{*}_{\diamond}$	$38.40_{\diamond}$	$23.13^{*}_{\diamond}$	$-1.67^{*}_{\diamond}$
SentiGAN	$37.25^{*}_{\diamond}$	$21.52^{*}_{\diamond}$	17.24^{*}	15.60	$36.85^{*}_{\diamond}$	23.57	$-2.42^{*}_{\diamond}$
Ours (vanilla)	**39.67**	**23.62**	$18.01_{\diamond}$	$16.00_{\diamond}$	$\mathbf{39.87}_{\diamond}$	$24.52_{\diamond}$	$-1.89_{\diamond}$
Ours (full)	39.35	23.25	**18.62**	**16.44**	39.10	**24.96**	**-1.54**

Table 5: Results for openQG. $*$ ($\diamond$) denotes that our vanilla (full) model differs from the baseline significantly based on one-side paired t-test with $p < 0.05$.

version uses Eq. 7 to compute rewards, and the "full" version uses Eq. 10 (the SVM model which gets the best performance in Table 4 are adopted as F_s). More details about hyper-parameters and training process are provided in the appendix.

We adopt the commonly-used BLEU, ROUGE-L and METEOR for question evaluation. Besides, our score function F_s based on OQRanD is also used. Similarly, we choose the the SVM model which gets the best performance in Table 4. We compute $F_s(\hat{Y})$ for each generated question $\hat{Y}$, and report the average value in "F_s-SVM" column of Table 5. As mentioned above, F_s shows if the generated questions are expected to receive more answers thus are more suitable for open discussion. The higher F_s a model gets, the better performance it has.

The results of our experiments are listed in Table 5. When it comes to BLEU, ROUGE-L and METEOR, our models get the best performance. This shows the advantage of making both of the generator and discriminator conditioned on input news. Besides, the full version of our model gets the best BLEU-3, BLEU-4 and METEOR values by introducing the linguistic-based question evaluation model during adversarial training. Of all the baselines, SentiGAN gets the best performances on BLEU-3 and BLEU-4, which is largely contributed by its penalty based objective function. Since the same piece of news always corresponds with multiple questions (and these questions may differ a lot) in OQGenD, models based on adversarial training (SeqGAN, SentiGAN and ours) always get better results than others (Seq2seq and CopyNet).

When it comes to F_s, the full version of our model gets the best performance, which illustrates that information from the SVM model is useful to generate questions with better quality. Besides, we can also use the conclusions in Section 3.2 to compare different models, e.g. questions generated by our full version model are the most concise (9.68 words per question). On the other hand, SentiGAN generates the longest questions (11.54 words per question).

5 Conclusion and Future Work

In this paper, we take the first step on teaching machines to ask open-answered questions from news for open discussion. To generate high-qualified questions, we analysis how language use affects the number of answers that a question receives based on OQRanD, a dataset created by variable control. These conclusions help us to build question evaluation models, and can also used to compare results of different question generation models. For question generation, we propose a model based on CGAN using reinforcement learning to introduce information from our evaluation model. Experiments show that our model outperforms commonly-used text generation methods.

There are many future works to be done. First, we will explore more powerful QG structure to deal with the huge difference between the length of input and output texts. Besides, how to better leverage prior knowledge during openQG (like human often do) is also interesting. Finally, combining openQG with its reverse task, openQA, is also worth exploration.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61772036) and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We thank the anonymous reviewers for their helpful comments. Xiaojun Wan is the corresponding author.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016. A simple but tough-to-beat baseline for sentence embeddings.

Youmna Borghol, Sebastien Ardon, Niklas Carlsson, Derek Eager, and Anirban Mahanti. 2012. The untold story of the clones: content-agnostic factors that impact youtube video popularity. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1186–1194. ACM.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Kenneth Mark Colby, Sylvia Weber, and Franklin Dennis Hilf. 1971. Artificial paranoia. *Artificial Intelligence*, 2(1):1–25.

Cristian Danescu-Niculescu-Mizil, Justin Cheng, Jon Kleinberg, and Lillian Lee. 2012. You had me at hello: How phrasing affects memorability. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 892–901. Association for Computational Linguistics.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. *arXiv preprint arXiv:1705.00106*.

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

Hady Elsahar, Christophe Gravier, and Frederique Laforest. 2018. Zero-shot question generation from knowledge graphs for unseen predicates and entity types. *arXiv preprint arXiv:1802.06842*.

Zhihao Fan, Zhongyu Wei, Siyuan Wang, Yang Liu, and Xuanjing Huang. 2018. A reinforcement learning framework for natural question generation using bi-discriminators. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1763–1774.

Alejandro Figueroa and Günter Neumann. 2013. Learning to rank effective paraphrases from query logs for community question answering. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Yifan Gao, Jianan Wang, Lidong Bing, Irwin King, and Michael R Lyu. 2018. Difficulty controllable question generation for reading comprehension. *arXiv preprint arXiv:1807.03586*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Marco Guerini, Alberto Pepe, and Bruno Lepri. 2012. Do linguistic style and readability of scientific abstracts affect their virality? In *Sixth International AAAI Conference on Weblogs and Social Media*.

Marco Guerini, Carlo Strapparava, and Gozde Ozbal. 2011. Exploring text virality in social networks. In *Fifth International AAAI Conference on Weblogs and Social Media*.

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. Association for Computational Linguistics.

Wenpeng Hu, Bing Liu, Jinwen Ma, Dongyan Zhao, and Rui Yan. 2018. Aspect-based question generation.

Hafedh Hussein, Mohammed Elmogy, and Shawkat Guirguis. 2014. Automatic english question generation system based on template driven scheme. *International Journal of Computer Science Issues (IJCSI)*, 11(6):45.

Vishwajeet Kumar, Kireeti Boorla, Yogesh Meena, Ganesh Ramakrishnan, and Yuan-Fang Li. 2018a. Automating reading comprehension by generating question and answer pairs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 335–348. Springer.

Vishwajeet Kumar, Ganesh Ramakrishnan, and Yuan-Fang Li. 2018b. A framework for automatic question generation from text using deep reinforcement learning. *arXiv preprint arXiv:1808.04961*.

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)*, volume 1, pages 889–898.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231. Association for Computational Linguistics.

Joseph Lee Rodgers and W Alan Nicewander. 1988. Thirteen ways to look at the correlation coefficient. *The American Statistician*, 42(1):59–66.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

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*, pages 105–114.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

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)*, volume 2, pages 321–326.

Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Sudha Rao and Hal Daumé III. 2018. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. *arXiv preprint arXiv:1805.04655*.

Mengye Ren, Ryan Kiros, and Richard Zemel. 2015. Exploring models and data for image question answering. In *Advances in neural information processing systems*, pages 2953–2961.

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. *arXiv preprint arXiv:1603.06807*.

Matthew P Simmons, Lada A Adamic, and Eytan Adar. 2011. Memes online: Extracted, subtracted, injected, and recollected. In *Fifth international AAAI conference on weblogs and social media*.

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

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063.

Chenhao Tan, Lillian Lee, and Bo Pang. 2014. The effect of wording on message propagation: Topic- and author-controlled natural experiments on twitter. *arXiv preprint arXiv:1405.1438*.

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

Ke Wang and Xiaojun Wan. 2018. Sentigan: Generating sentimental texts via mixture adversarial networks. In *IJCAI*, pages 4446–4452.

Tong Wang, Xingdi Yuan, and Adam Trischler. 2017. A joint model for question answering and question generation. *arXiv preprint arXiv:1706.01450*.

Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to ask questions in open-domain conversational systems with typed decoders.

Wei Wei and Xiaojun Wan. 2017. Learning to identify ambiguous and misleading news headlines. *arXiv preprint arXiv:1705.06031*.

Joseph Weizenbaum et al. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Kaichun Yao, Libo Zhang, Tiejian Luo, Lili Tao, and Yanjun Wu. 2018. Teaching machines to ask questions. In *IJCAI*, pages 4546–4552.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*.

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

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

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

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671. Springer.

A Details of Language Model

In this section, we introduce the details of our language models described in section 3.2.

We used the HanLP toolkit [5] perform word segmentation. The toolkit was also used to get the POS of each word. To train language models, we adopted the SRILM toolkit [6]. During this process, we used modified kneser-ney smoothing for all the language models based on word n-grams and witten-bell smoothing for language models based on POS n-grams.

B Details of Question Evaluation Models

In this section, we introduce the details of our question evaluation models described in section 3.3.

We adopted the Ranklib toolkit [7] to train the random forest model. For the SVM model, we used the SVM-rank toolkit [8]. More specifically, we set the trade-off between training error and margin of SVM to 3 and chose the linear kernel function.

For CNN and RNN models, the word embedding size is 128, and the size of POS embedding is 32. The RNN model is a single-layer bidirectional LSTM network with 128 hidden units. As for the CNN model, the convolution layer contains filters whose sizes are 160 × 1, 160 × 2, 160 × 3, 160 × 4. The counts for each kind of filters are 64, 64, 64, 64, and the stride for each of them is 1. After the convolution layer, there is a max-pooling layer and a fully connected layer with the sigmoid activation to get the final result.

C Details of Question Generation models

In this section, we introduce the details of our question generation model described in section 4.2.

Our model is composed by a generator and a discriminator. The generator is a typical seq2seq model. It has three components: an encoder network, a decoder network and an attention network. The encoder is a single-layer bidirectional GRU with 64 hidden units while the decoder is a single-layer unidirectional GRU with 128 hidden units. The CNN of discriminator for news contains filters whose sizes are 128 × 1, 128 × 2, 128 × 3, 128 × 4, 128 × 5. The counts for each kind of filters are 32, 64, 64, 32, 16, and the stride for each of them is is set to 1. The CNN of discriminator for questions contains filters whose sizes are 128 × 1, 128 × 2, 128 × 3, 128 × 4. The counts for each kind of filters are 32, 64, 64, 32, and the stride for each of them is set to 1.

D Examples of Our Datasets

As mentioned above, we controlled the effect of topic, time and author to get OQRanD. During this process, we divided all the questions into 37 subsets according to manually-tagged topics. These topics are listed in Table 6. The examples of OQRanD are shown in Table 7. The examples of OQGenD are shown in Table 8 (in case that the original news are too long, we omit the sentences that is not related to the qestions).

[5] http://hanlp.linrunsoft.com

[6] http://www.speech.sri.com/projects/srilm/

[7] http://www.lemurproject.org/ranklib.php

[8] http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

Topics	
宗教(Religion)	自然科学(Science)
职场(Workplace)	政治(Politics)
运动健身(Physical Exercise)	娱乐(Entertainment)
游戏(Game)	影视(Film and Television)
音乐(Music)	艺术(Art)
心理学(Psychology)	体育(Sports)
时尚(Fashion)	社会科学(Social Sciences)
设计(Design)	商业(Business)
人文(Humanity)	情感(Emotion)
汽车(Car)	美食(Food)
旅行(Travel)	科技(Science and Technology)
军事(Military)	经济(Economics)
金融(Finance)	教育(Education)
健康(Health)	家居(Home Furnishing)
工程学(Engineering)	法律(Law)
宠物(Pets)	财务(Finance)
动漫(Comic)	母婴(Mother and Child)
其他(Other)	两性(Bisexual)
ACG	

Table 6: Topics of our questions.

	Questions	#Ans
1	有什么有趣且有知识的书推荐? (What interesting and knowledgeable books can you recommend?)	10
	2015 年你读过最好的书有哪些？为什么? (What are the best books you have read in 2015? Why?)	45
2	你的家乡有什么初次尝试不太容易接受的美食吗? (Is there any food that is hard to accept for the first time in your hometown?)	1
	有哪些在自己家乡很正常但在外地人眼里是黑暗料理的美食? (Which foods are normal in your hometown but are dark cuisine in the eyes of foreigners?)	89
3	请推荐值得一看的电影(列表)? (Please recommend some movies that are worthy of watching (make a list)?)	4
	你会推荐哪些值得一看的电影? What movies do you think are worthy of watching?)	24
4	如何判断自己得了抑郁症? (How to judge that if I am suffering from depression?)	5
	抑郁症有哪些症状表现? (What are the symptoms of depression?)	38
5	能帮我推荐一支送女生的口红吗? (Can you recommend me a lipstick as a gift for a girl?)	3
	有什么适合女生的平价口红? (Is there any cheap lipstick for girls?)	1062

Table 7: Examples of OQRanD. “#Ans” denotes for “the number of answers”.

news	最后一次世界杯，C罗和梅西谁会赢。C罗和梅西谁更强？这个问题自两人出道就争论至今。2018年俄罗斯世界杯，…… (Who will win the last World Cup between Ronaldo and Messi? Who is stronger, Ronaldo or Messi? This issue has been debated since the beginning of their career. The 2018 World Cup in Russia ...)
gold questions	最后一次世界杯，C罗和梅西谁会赢? (Who will win the last World Cup between Ronaldo and Messi?)
	最后一次世界杯，C罗会战胜梅西吗? (Will Ronaldo defeat Messi in the last World Cup?)
	最后一次世界杯，C罗会输给梅西吗? (Will Ronaldo lose to Messi in the last World Cup?)
	最后一次世界杯，梅西会输给C罗吗? (Will Messi lose to Ronaldo in the last World Cup?)
	最后一次世界杯，梅西会战胜C罗吗? (Will Messi defeat Ronaldo in the last World Cup?)
news	欧盟支持科威特出面"斡旋"卡塔尔断交风波。中新社布鲁塞尔6月19日电(记者沈晨) 欧盟外交与安全政策高级代表莫盖里尼19日在欧盟外长例行会议上表态，支持科威特出面“斡旋”卡塔尔断交风波，…… (EU supports Kuwait to “mediate” Qatar’s tumult of break-up of diplomatic relations. China News Service report in Brussels(reporter shen chen). Federica Mogherin, the European Union’s foreign-policy chief, spoke at the routine meeting of EU foreign ministers on the 19th to support Kuwait to “mediate” Qatar’s tumult of break-up of diplomatic relations ...)
gold questions	如何看待埃及、沙特、巴林几乎同时宣布与卡塔尔断交? (How do you think that Egypt, Saudi Arabia and Bahrain almost simultaneously announced the break-up of diplomatic relations with Qatar?)
	国家之间断交意着什么? (What does it mean when countries break off?)

Table 8: Examples of OQGenD.