INTELLIGENCE IS ASKING THE RIGHT QUESTION: A STUDY ON JAPANESE QUESTION GENERATION

Lasguido Nio, Koji Murakami

Rakuten Institute of Technology, Rakuten Inc.
{lasguido.nio, koji.murakami}@rakuten.com

ABSTRACT

Traditional automatic question generation often requires hand-crafted templates or sophisticated NLP pipelines. Such approaches, however, require extensive labor and expertise to morphologically analyze the sentences and create the NLP framework. Our works aim to simplify these labors. We conduct a contrastive experiment between two types of sequence learning: statistical-based machine translation and attention-based sequence neural network. These models can be trained end-to-end, and it can capture the pattern between the input sequence and output sequence, thus diminishing the need to prepare a sophisticated NLP pipeline. Automatic evaluation results show that our system outperforms the state-of-the-art rule-based system, and also excels in terms of content quality and fluency according to a subjective human test.

Index Terms— question generation, neural sequence learning, data-driven question generation

1. INTRODUCTION

Question generation is the task of generating a natural question from a given input sentence. Naturally, humans ask questions to express informational needs [1]. Intuitively, somebody will ask a question if they aren't certain about something. Creating a good question is not a trivial task, and it is much harder to create a deep question rather than a factoid question. While factoid questions require an explicit memory response, deep questions require more profound thinking and recall [2]. In this study we will limit ourselves to questions of factoid type that can be answered explicitly, but are tedious to create.

Language generation is an interesting challenge because its applications involve a vast amount of domains such as a chatbot component in dialogue systems [3, 4, 5], question generation for reading comprehension [6, 7], Frequently Asked Questions (FAQ) generation [8], and question-answering database creation [9, 10]. To put it simply, the benefits of question generation are twofold. The first benefit, as we know, is to fulfill informational needs [2]. The second is to assess others' level of understanding or to provide additional information beyond the description [1]. For example, teachers ask questions not because they do not know the answer, but because they want to give a certain level of understanding to their students. On the other hand, an e-commerce website may show question-answer information to provide a potential buyer a good amount of information before making a purchase decision [11].

Previous approaches on question generation tasks relied either on rule-based approaches [12, 13], or complex NLP pipelines [2, 14]. Aiming to simplify and increase system robustness, some recent works have started to address this task with recent neural network-based machine learning algorithms [10, 15]. However some issues still remain.

Next, we point out two common challenges that often faced by this task: the first is the complexity of analyzing and building a pipeline that is able to generate a natural and relevant question, and the second is data collection. When it comes to statistical-approaches, many researchers utilize crowdsourcing which saves time, but is more expensive. In this research, we aim to design a system that is both efficient and robust, that is able to generate natural and relevant questions. On top of that, the utilization of machine learning enables us to do soft-matching [16, 17]. This feature helps with the robustness of the model, enabling it to deal with question patterns that are not available in the training dataset.

To address the above-mentioned challenges we utilize end-to-end machine learning approaches in order to learn the sentence-question pattern automatically. The recent success of neural machine translation (NMT) technology [18, 19] promises us a high-performance translation result. However, there have also been some reports of traditional statistical machine translation (SMT) approaches that boast better performance, especially in certain low-resource conditions [20]. Inspired by this finding, we utilize these two popular machine translation paradigms in our question generation task.

The power of NMT lies within the attention mechanism and bi-directional architecture properties [21, 22]. The attention mechanism allows us to selectively focus on parts of the source sentence during translation, while the bi-directional architecture enables us to capture information regarding long-term dependency structure of sentences. On the other hand, SMT gains an advantage from phrase-alignment that is statistically calculated during the learning process [23], this alignment is basically a mapping rule that is learned via the train-

ing data.

In this work we make the following contributions:

- To our knowledge, we are the first to utilize end-to-end machine learning translation techniques for question generation on Japanese sentences. And we are among the first to employ a deep sequence-to-sequence learning approach to generate questions.
- We perform a contrastive experiment to compare our proposed end-to-end approaches with the rule-based approach on Japanese text. We focus ourselves to yes-no and how-what type questions. Later we evaluate the models using both automatic and manual evaluation; deep analysis on the generated sentence using content quality and fluency metrics is also performed.

2. RELATED WORK

We consider our works to be aligned with the task of reading comprehension. This requires a machine to understand natural language and make use of world knowledge to answer questions. The recent growing interest on reading comprehension tasks has led some researchers to build some baseline datasets such as Stanford Question Answering Dataset[1] (SQuAD) [6], MS MARCO [7], and NewsQA [24]. Various works have been done in this field exploring various techniques ranging from answer re-ranking algorithms, reinforcement learning, sum reader networks, gated-attention structures, to domain adaptation techniques [25, 26, 27, 28, 29, 30, 31]. Our work follows in a similar vein, however from a slightly different perspective: instead of trying to answer questions directly, we are trying to generate questions given limited context sentences.

Conventional approaches on question generation tasks often rely on a rule-based methodology [32, 33, 34, 12]. This kind of approach requires rules that map sentences or contextual information to a specific interrogative template. The success of this approach depends heavily on the quality of the designed rules and templates, which require a deep linguistic knowledge. As an improvement to rule-based approach, the ranking method is utilized [35, 13]. This method forces the hand-crafted rule or question generation pipeline to over-generate questions, and then ranks them with the trained unsupervised-based ranker. The ranking method helps increase the performance, however, this approach is still considered tedious because designing a complicated template and features is necessary.

Going further, other researchers applied more advanced methods that exploit NLP techniques, for example Labutov et al. [2] who utilized crowdsourcing to collect the templates and exploit low-dimensional ontology to match sentences into questions. Other works exploit NLP features such as entity detection, synonym, antonym, and sentence modifiers to create questions with a broad type coverage [14, 34, 12, 36]. Yao et al. [37] incorporated deep linguistic grammar to break down sentences into context grammar, and then transform them back into questions.

Du et al. [15] point out the importance of attention-based sequence learning adaptation to the generation of questions. Promoting a practical, easily trainable end-to-end model, this work utilizes attention-based Bi-directional Long Short-Term Memory (Bi-LSTM). In contrast with this approach, ours tackles the issue of training with a small dataset, which makes neural based learning much more challenging. Furthermore, we also applied some soft domain adaptation learning to our end-to-end model.

Another piece of related work is presented by Yang et al. [10]. This work utilizes generative models to generate questions from unlabeled paragraphs. This is done by harnessing linguistic tags to extract possible answers in an unlabeled paragraph, following which the generative networks treat the candidate answers and context paragraph as input feed training data. Later, the generated question and the human-created question will be verified through the discriminative model, the goal being to build a multipurpose robust question-answer database. This approach differs from ours in that we are investigating how to generate questions from the limited context sentences. Thus, we employ an end-to-end sequence learning based algorithm to learn the pattern between the context sentence and output question.

3. QUESTION GENERATION

We formulate our question prediction task as follows: Given an input context sentence c, we aim to generate a natural question q. Both input and output can be a sequence of arbitrary length $[c_1, ..., c_{|c|}]$ and $[q_1, ..., q_{|q|}]$. This question generation task can be defined as:

$$\hat{q} = \arg\max_{q} P(q|c) \tag{1}$$

where $\hat{q}$ is the system best-generated question, and $P(q|c)$ is the conditional probability of the predicted question sequence q, given the input c. Here, we aim to maximize $P(q|c)$ over all possible q. This conditional probability can be likened to a translation model in a statistical sequence learning approach, or a conditional log-likelihood in neural sequence learning approach.

In order to generate a question, we have 3 main approaches. First, we set up a rule-based question generator as a baseline approach. Second, we employ phrase-based statistical machine translation as the statistical sequence learning approach. And last, we utilize Bi-LSTM with attention mechanism as the neural sequence learning approach. Each will be presented in the next section.

3.1. Rule Based Question Generation

The Japanese language falls under the class of agglutinative languages. Verbal expression is fundamentally located at the

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

end of a sentence, and it can be added to various words such as nouns, particles, and auxiliary verbs with conjugations at the ending of the word stem. If we convert an affirmative sentence to yes-no question form, we need to add an appropriate sentence ending particle. For generating wh-type question, we need not only that particle, but also need to use an interrogative word.

We have extracted around 30,000 affirmative sentences from our datasets, which we explain in detail in Section 4. We shall focus on the last 3 words in these sentences, because in over 90% of the sentences in our data, an interrogative form can be generated by editing these 3 words and adding an interrogative expression at the end of the sentence. We analyzed the sentences which end with the following part-of-speech: verbs, adjectives, particles, and auxiliary verbs; we will also consider a few nouns that function similarly to adjectives, verbs, or auxiliary verbs. While creating rules, only "KA(か)" was used as the sentence ending particle in interrogative expressions, because this is the most popular and widely used with any part-of-speech. An interrogative expression and an edit flag (use or delete) of each 3 words are annotated to all sentences by the human annotator. If several sentences have the same interrogative expression and share the same words, they are merged. Finally, we obtained approximately 1,700 converting rules.

To generate wh-type questions, we apply our above described rules and randomly replace only one word with "how" or "what". The words which can be replaced are (1) a noun in an objective case, (2) an adjective, or (3) a verb.

3.2. Statistical Question Generation

Here we utilize phrase-based statistical machine translation (SMT) to capture the pattern between context sentence input and question output. SMT has been proven successful to address various NLP tasks, ranging from sequence generation for Twitter[2] to chat-oriented dialog system response [38, 39, 40].

We treat the sentence-question pair as a parallel corpus to train the translation model, which is based on the noisy channel model. Considering the general question generation task, we reformulate Equation 1 with Bayes rule as

$$\hat{q} = \arg\max_{q} P(c|q)P_{LM}(q). \tag{2}$$

This way we can obtain the language model $P_{LM}(q)$ and separate the translation model $P(c|q)$.

Going deeper, we can decompose the translation model $P(c|q)$ into

$$P(\overline{c}_i^I|\overline{q}_i^I) = \prod_{i=1}^{I} \phi(\overline{c}_i|\overline{q}_i) d(start_i - end_{i-1}). \tag{3}$$

[2]https://twitter.com

In the decoding stage we segment the input sentence c into the phrase sequence $\overline{c}_1^I$. This input phrase sequence $\overline{c}_1^I$ contains input sentence phrase $\overline{c}_i$ that will be mapped to target question phrase $\overline{q}_i$. This phrase translation is portrayed as the probability distribution $\phi(\overline{c}_i|\overline{q}_i)$.

Later, Expectation Maximization learning was used [41] to obtain the phrase-based joint probability model on the parallel corpus. This model was then used to train the relative distortion probability distribution $d(start_i - end_{i-1})$ [23]. Where $start_i$ denotes the beginning chunk of the phrase in the input sentence that corresponds to the ith phrase in the output question, and end_{i-1} denotes the last chunk of the phrase in the input sentence that corresponds to the $(i-1)$th phrase in the output question.

Applying SMT to the sequence learning, we model the probability of a generated question q given the input sentence context c using the log-linear of phrase-translation probabilities, securing q as a well-formed question. To find the best response, beam search decoder [42] is employed at the end.

3.3. Neural Question Generation

In this section, we describe our approach in generating a question with the neural machine translation (NMT) technique. In contrast with the SMT, the power of NMT lies on the bi-directional recursive architecture and global attention mechanism. The bi-directional architecture enables the model to learn in both a forward and backward context. The attention mechanism allows the model to put emphasis on a certain part of the sentence, imitating the way humans think to solve a task.

Taking into account the question generation task formulation on Equation 1, we can factorize the conditional probability $p(q|c)$ as

$$P(q|c) = \prod_{t=1}^{|q|} P(q_t|c, q_{1..t-1}). \tag{4}$$

Where q_t are word candidates that combine to give output question q, we treat the conditional probability $P(q|c)$ as a product of word-level prediction. The probability of q_t is predicted based on the input context sentence c, and all the words that have been previously generated $q_{1..t-1}$.

Furthermore, we can break down the word-level conditional probability into

$$P(q_t|c, q_{1..t-1}) = softmax(W_q \tanh(W_b[b_t; a_t])). \tag{5}$$

Where b_t portrays the bi-directional recursive network state variable at the time step t, a_t is the attention based encoding of input sentence context c at decoding time step t, and W_q and W_b are the parameters to be learned.

3.3.1. Bi-directional Long Short-Term Memory

Before we discuss Bi-directional Long Short-Term Memory (Bi-LSTM), we present a brief explanation on Long Short-

Term Memory (LSTM). LSTM takes words from an input sentence in a distributed word representation format. Distributed word representation is a *n*-dimensional vector of continuous values used to represent a word in the vocabulary [43, 44, 45]. Each word in the dictionary ($w \in W$) is embedded into *n*-dimensional space ($L \in \mathbb{R}^{n \times |W|}$). Finally, a word vector can be seen as a single vector in the column of L.

The LSTM architecture consists of a set of recurrently connected memory blocks. These memory blocks enable the LSTM to store and access information over a period of time. Each block contains one memory cell c and three multiplicative units (gates). These gates allow the LSTM memory cell to perform write, read, and reset operation. They are the so-called input gate i, forget gate f, and output gate o. Mathematically one block of LSTM can be viewed as:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i), \quad (6)$$

$$f_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_f), \quad (7)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \quad (8)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o), \quad (9)$$

$$h_t = o_t \tanh(c_t), \quad (10)$$

where x_t is a single distributed vector L of word w from the input context sentence c, σ is a logistic sigmoid function, and h is a hidden vector. The weight W and b represent the edge connection matrix and bias vector.

One drawback of the LSTM architecture is that they only consider the previous context. In order to make LSTM aware of both previous and subsequent context, it needs to process data in both directions with two separate hidden layers [22]. Later these two hidden layers are combined into the same output layer. This architecture is the so-called Bi-LSTM.

Bi-LSTM computes forward $\overrightarrow{h_t}$ and backward $\overleftarrow{h_t}$ hidden sequence from the LSTM output result. Then it produces the output h_t by iterating from $t = 1$ to T for the forward layer, and from $t = T$ to 1 for the backward layer, thus we can obtain bi-directional recursive network state variable by concatenating $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$

$$b_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]. \quad (11)$$

Recently, Bi-LSTM has been used intensively for real-world applications, ranging from signal processing tasks [46, 47] to text processing tasks [48, 49].

3.3.2. Attention Mechanism Encoder

Attention-based encoding of input context sentence c at time step t (a_t) is obtained by taking the sum weighted average over b_t ($t = 1, .., |c|$),

$$a_t = \sum_{i=1,..,|c|} aw_{i,t} b_i. \quad (12)$$

Similar with Luong et al. [21], we also define attention weight $aw_{i,t}$ as a comparison between the current target hidden state b_{target} with each source hidden state $\bar{b}_{source}$,

$$aw_{i,t} = \frac{\exp(b_{target}^T W_b \bar{b}_{source})}{\sum_{source'} \exp(b_{target}^T W_b \bar{b}_{source'})}. \quad (13)$$

3.3.3. Training and Inference

We train our model to minimize the negative log-likelihood of the training data $\mathcal{S} = \{(c^{(i)}, q^{(i)})\}_{i=1}^{S}$, with respect to all parameters θ,

$$\mathcal{L} = -\sum_{i=1}^{S} \sum_{j}^{|q|} \log P(q_j^{(i)} | c^{(i)}, y_{1..j-1}^{(i)}; \theta). \quad (14)$$

Later, during inference, the generated result is obtained by approximately maximizing the conditional probability given by the model, commonly known as the beam-search strategy [50].

4. DATASETS

The sentence-question pairs dataset that we constructed in this study is based on the user-merchant review pages [51] on the Rakuten Japan website[3]. We collect the review data from the selected wine genre products from 2013 to 2017. Overall we obtain 241,794 reviews and segment it into 673,963 sentences.

From there, we run our rule-based question generation on all of the review sentences, and employ annotators to check and correct the generated questions. In the end, we managed to gather around 30k data points. We split this dataset into two different categories YN and WH, which consecutively portray yes-no and how-what type question, and then further randomly subdivide each of these categories into training, development, and test sets.

	Training	Development	Test	Total
YN	15,755	300	300	16,355
WH	14,571	600	300	15,471

Table 1. Dataset statistics.

However, due to the limited amount of data, we only take 300 pairs of sentence-question as a test set. The results of our experiments will later be presented in the context of this test set.

The details of this dataset are provided in Table 1. There are around 15k training set for each YN and WH category, making total 30k dataset for training. We train the model separately with the different dataset for each YN and WH. Furthermore, we are also employing domain adaptation technique, making a compound model that trained from both YN and WH dataset.

[3]https://www.rakuten.co.jp

	SMT	SMT_CMP	NMT	NMT+PRE	NMT_CMP	NMT_CMP+PRE	RULE
YN	91.70	64.25	38.30	33.38	42.32	42.66	71.51
WH	75.66	81.75	66.45	69.74	71.70	73.37	83.53

Table 2. Automatic evaluation result with BLEU score given various approaches.

5. EXPERIMENTAL SETUP AND EVALUATION

5.1. Experimental Setup

In this paper, we employ two types of approaches: statistical question generation, and neural question generation. We employ Moses[4] toolkit [42] and OpenNMT system [52] for SMT and NMT implementation, respectively.

During SMT training, we employ GaCha filtering [53] to remove noisy sentence level alignment, with the GaCha filtering threshold set to 0.8. In our experiment, the SMT model is denoted by SMT.

As for the NMT training, we employ the Japanese Wikipedia[5] dataset provided by Polyglot Project [54] to enrich the model word embedding. Using this dictionary, we learned the word representation with FastText [55]. The NMT model that utilizes this embedding is indicated by +PRE.

In the NMT-based approach, we follow the same configuration used by Du et al. [15]. Therefore, the model denoted by NMT is treated as the state-of-the-art baseline.

Going a step further, we build a compound model using a combined dataset. This model is denoted by CMP (SMT_CMP, NMT_CMP, and NMT_CMP+PRE). This compound model is trained using a compound dataset (YN and WH). In contrast, the non-compound models (SMT, NMT, and NMT+PRE) are trained with the corresponding training and test sets. For example, in a non-compound model, if we evaluate the model with the YN dataset, we train the model only with the YN dataset. From this compound model, we would like to assess how the model works given a complex and noisy training set.

5.2. Automatic and Human Subjective Evaluation

At the end of this experiment, we evaluated our system response both automatically by calculating the generated response $\hat{q}$ with the question in reference database q, and subjectively by giving out a survey to humans.

Automatic evaluation of our models are done by the BLEU-4 metric. This metric calculates how well the generated question compares to the reference question. The results of the BLEU-4 evaluation is presented above in Table 2. More analysis on these results will be discussed in Section 6.

Next, the human evaluation studies are performed to measure the quality of questions. We conduct the evaluation on rule-based approach (RULE), both SMT approaches (SMT, SMT_CMP), and the best model in NMT approaches according to the automatic evaluation results (NMT_CMP+PRE). We apply two metrics, content quality and fluency. Content quality indicates grammaticality and consistency with the reference, and the results of this evaluation can be seen in Fig. 1. Fluency focuses on only readability of the generated questions, with the results of this is shown in Fig. 2. If a generated question is neither of interrogative form or question type, conflict happens between YN and WH, and so both scores should be lower. We follow the scaling criteria guideline defined by Japanese Patent Office[6].

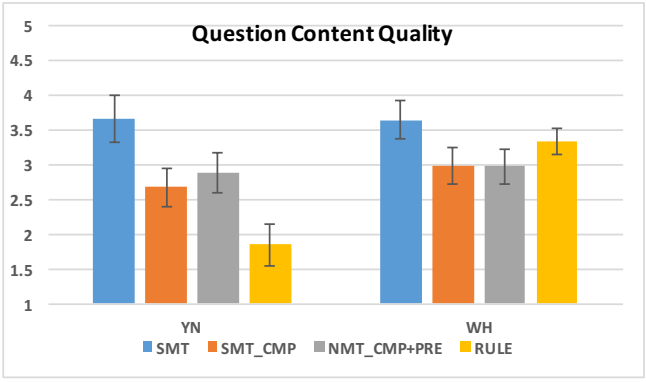

Fig. 1. Subjective evaluation result on content quality.

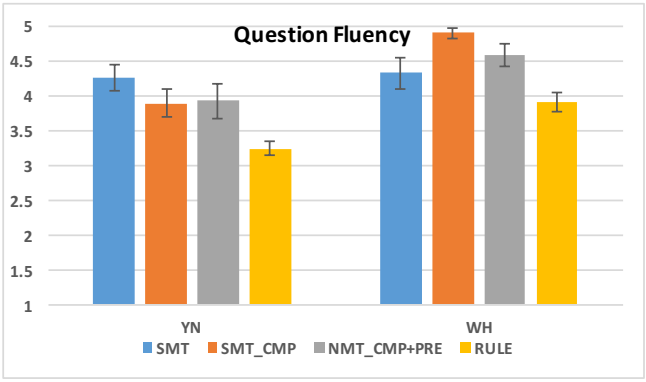

Fig. 2. Subjective evaluation result on question fluency.

It is easy to imagine that the rule-based approach performs with higher precision and lower recall because rules will replace a few words in a sentence. If we evaluate samples, in which the rule-based system outperformed other metrics, the evaluation might be unfair for SMT and NMT-based approaches. So we focus on samples in which the rule-based system failed to generate questions out of rules in YN, and those in which BLEU scores on the rule-based approach are

[4]https://www.statmt.org/moses
[5]https://ja.wikipedia.org
[6]https://www.jpo.go.jp/shiryou/toushin/chousa/pdf/tokkyohonyaku_hyouka/01.pdf (In Japanese)

Sentence Context	SMT / SMT_CMP	NMT_CMP+PRE	RULE
Yawarakana [suave] kajitsumi-to [fruity flavor and] hodoyoi san-ga [modestly sized acids] baransu yoku [with good balance] totonotte imasu [being settled]	**Donna** kajitsumi-to [**what sort of** fruity flavor and] hodoyoi san-ga [modestly sized acids] baransu yoku [with good balance] totonotte imasu-ka? [being settled?]	Kajitsumi-to [fruity flavor and] hodoyoi san-ga[modestly sized acids] baransu yoku [with good balance] totonotte imasu-ka? [being settled?]	Yawarakana [suave] Kajitsumi-to [fruity flavor and] hodoyoi san-ga[modestly sized acids] baransu yoku [with good balance] **dou** [**how**] imasu-ka? [being settled?]

Table 3. Comparison of generated question given various approaches for sentence ”柔らかな果実味と程好い酸がバランスよく整っています”

lower in WH. We randomly sampled 50 sentence-question pairs for YN and WH from these two cases, and evaluated by asking two Japanese native speakers to rate the pairs in terms of the metrics above on a 1-5 scale (5 for the best).

6. RESULTS AND DISCUSSION

From the automatic evaluation results (Table 2) we can see that the SMT approach (SMT and SMT_CMP) in general performs better than NMT (NMT and NMT_CMP) and the rule-based RULE approach, except on the WH type dataset, where it seems that the rule-based approach performs a little bit better. However, during the subjective evaluation, we found out that the content quality and fluency of the SMT approach is much better. The soft-matching feature of machine learning allows the model to become more flexible in generating natural questions, while the rule-based generated question looks like a canned question.

Fig.1 and 2 show the results of the human evaluation. We see that our SMT approaches (SMT and SMT_CMP) outperform NMT_CMP+PRE and RULE in our experimental settings while NMT_CMP+PRE performs moderately well. An inter-rater agreement of weighted Kappa values of 0.952 to WH and 0.972 to YN are achieved for rating.

Overall, the SMT approach achieves highest scores in content quality in both YN and WH settings. According to our error analysis, the main reason is that SMT did not confuse question types between YN and WH. The rest of the systems perform on par on the WH type dataset, while the RULE approach performs poorly on the YN scenario.

As for the compound approaches, SMT_CMP and NMT_CMP show better scores on fluency for WH. Both of these approaches were trained on all sentence-question pairs in our dataset, while SMT was only trained on single type sentence-question pairs (YN or WH). This implies that the quantity of training data could be crucial for training a better model. Koehn et al. showed the quality of NMT starts much lower and finally beats the SMT with sufficient of training data [20], so there is much possibility that NMT could beat SMT-based approaches if we had over a hundred million sentence pairs.

Both content quality and fluency on YN questions from the rule-based approach are lower than others because 34 out of 50 sentences are outside of the rules and it failed to generate questions. For WH question generation, only a few words are replaced in the rule-based approach so that it contributes to content, while fluency is lower than others.

Table 3 shows an example question generated by each approach. We focus only on WH here. The sentences in the table are separated by each phrase with English words as a translation. For this sentence, 2 kinds of questions can be produced with interrogative words “どんな(what)” or “どう(how)”. The example indicates that both of the SMT-based approaches successfully produced an interrogative word (“どんな(what)”), however, no interrogative word was generated by the NMT-based approach even if the model has been trained. The rule-based approach has produced the question by replacing a verb “整う” with “どう(how)” but this question is not good enough grammatically. While the rules could focus on replacing only a verb, the replacement should be done with inserting a non-volitional verb “natte”, as “dou/**natte**/imasuka” because “imasuka” represents a state. However, it is difficult to create complicated rules for capturing multiple verbs at the same time.

7. CONCLUSION

In this paper we presented a data-driven automatic question generation approach for Japanese text. Two approaches were investigated, first was statistical question generation with SMT, and the second was neural question generation with Bi-LSTM on a NMT framework. Our experimental evaluation shows that the statistical question generation model achieves state-of-the-art performance on Japanese text in both automatic and subjective evaluation.

From this work we can see several interesting future research directions. One is to employ a Generative Adversarial Network for question generating problems. The learning process can also be done not only in the small sentence context, but also paragraph-level (such as item descriptions, or review paragraph). Another future possibility is to utilize this generation technique for another task such as a dialogue chatbot system or automatic QA generation system.

8. REFERENCES

[1] V. Rus, P. Piwek, S. Stoyanchev, B. Wyse, M. Lintean, and C. Moldovan, "Question generation shared task and evaluation challenge: Status report," in *Proceedings of the 13th European Workshop on Natural Language Generation*. 2011, ENLG '11, pp. 318–320, Association for Computational Linguistics.

[2] I. Labutov, S. Basu, and L. Vanderwende, "Deep questions without deep understanding," in *Proceedings of the 7th International Joint Conference on Natural Language Processing*. July 2015, IJCNLP '15, pp. 889–898, Association for Computational Linguistics.

[3] P. Piwek, H. Hernault, H. Prendinger, and M. Ishizuka, "T2D: Generating dialogues between virtual agents automatically from text," in *Proceedings of the 7th International Conference on Intelligent Virtual Agents*. 2007, IVA '07, pp. 161–174, Springer Berlin Heidelberg.

[4] N. Mostafazadeh, I. Misra, J. Devlin, M. Mitchell, X. He, and L. Vanderwende, "Generating natural questions about an image," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. 2016, ACL '16, pp. 1802—-1813, Association for Computer Linguistics.

[5] K. M. Colby, S. Weber, and F. D. Hilf, "Artificial paranoia," *Artificial Intelligence*, vol. 2, no. 1, pp. 1–25, 1971.

[6] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2016, EMNLP '16, pp. 2383–2392, Association for Computational Linguistics.

[7] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, "Ms marco: A human generated MAchine Reading COmprehension dataset," *arXiv preprint arXiv:1611.09268*, 2016.

[8] D. Tang, N. Duan, T. Qin, and M. Zhou, "Question answering and question generation as dual tasks," *arXiv preprint arXiv:1706.02027*, 2017.

[9] H. Ali, Y. Chali, and S. A. Hasan, "Automation of question generation from sentences," in *Proceedings of the 3rd Workshop on Question Generation*, 2010.

[10] Z. Yang, J. Hu, R. Salakhutdinov, and W. Cohen, "Semi-supervised qa with generative domain-adaptive nets," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 2017, ACL '17, pp. 1040–1050, Association for Computational Linguistics.

[11] E. J. Lee and S. Y. Shin, "When do consumers buy online product reviews? effects of review quality, product type, and reviewer' s photo," *Computers in Human Behavior*, vol. 31, pp. 356 – 366, 2014.

[12] J. C. Brown, G. A. Frishkoff, and M. Eskenazi, "Automatic question generation for vocabulary assessment," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. 2005, HLT '05, pp. 819–826, Association for Computational Linguistics.

[13] Chali Y. and S. A. Hasan, "Towards automatic topical question generation," in *Proceedings of the 24th International Conference on Computational Linguistics*. 2012, COLING '12, pp. 475—-492, Association for Computational Linguistics.

[14] O. Rokhlenko and I. Szpektor, "Generating synthetic comparable questions for news articles," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. 2013, ACL '13, pp. 742–751, Association for Computer Linguistics.

[15] X. Du, J. Shao, and C. Cardie, "Learning to ask: Neural question generation for reading comprehension," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 2017, ACL '17, pp. 1342–1352, Association for Computational Linguistics.

[16] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *The Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[17] L Nio, S. Sakti, G. Neubig, K. Yoshino, and S. Nakamura, "Neural network approaches to dialog response retrieval and generation," *IEICE Transactions*, vol. 99-D, no. 10, pp. 2508–2517, 2016.

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the 3rd International Conference on Learning Representations*, 2015, ICLR '15.

[19] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. 2014, NIPS '14, pp. 3104–3112, MIT Press.

[20] P. Koehn and R. Knowles, "Six challenges for neural machine translation," in *Proceedings of the 1st Workshop on Neural Machine Translation*. 2017, pp. 28–39, Association for Computational Linguistics.

[21] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 1412–1421, Association for Computational Linguistics.

[22] M. Schuster and K.K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[23] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. 2003, NAACL '03, pp. 48–54, Association for Computational Linguistics.

[24] A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, and K. Suleman, "Newsqa: A machine comprehension dataset," in *Proceedings of the 2nd Workshop on Representation Learning for NLP*. 2017, pp. 191–200, Association for Computational Linguistics.

[25] Rudolf K., Martin S., Ondrej B., and Jan K., "Text understanding with the attention sum reader network," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. 2016, ACL '16, pp. 908–918, The Association for Computer Linguistics.

[26] Y. Shen, P. S. Huang, J. Gao, and W. Chen, "ReasoNet: Learning to stop reading in machine comprehension," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2017, KDD '17, pp. 1047–1055, Association for Computing Machinery.

[27] C. Chu, R. Dabre, and S. Kurohashi, "An empirical comparison of domain adaptation methods for neural machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 2017, ACL '17, pp. 385–391, Association for Computational Linguistics.

[28] B. Dhingra, H. Liu, Z. Yang, W. Cohen, and R. Salakhutdinov, "Gated-attention readers for text comprehension," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. 2017, ACL '17, pp. 1832–1846, Association for Computational Linguistics.

[29] S. Wang, M. Yu, J. Jiang, W. Zhang, X. Guo, S. Chang, Z. Wang, T. Klinger, G. Tesauro, and M. Campbell, "Evidence aggregation for answer re-ranking in open-domain question answering," in *Proceedings of the 6th International Conference on Learning Representations*, 2018, ICLR '18.

[30] A. Abujabal, R. R. Saha, M. Yahya, and G. Weikum, "Never-ending learning for open-domain question answering over knowledge bases," in *Proceedings of the 2018 World Wide Web Conference*. 2018, WWW '18, pp. 1053–1062, International World Wide Web Conferences Steering Committee.

[31] S. Wang, M. Yu, X. Guo, Z. Wang, T. Klinger, W. Zhang, S. Chang, G. Tesauro, B. Zhou, and J Jiang, "R^3: Reinforced ranker-reader for open-domain question answering," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. 2018, AAAI '18, pp. 5981–5988, AAAI Press.

[32] R. Mitkov and L. A. Ha, "Computer-aided generation of multiple-choice tests," in *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*. 2003, HLT-NAACL-EDUC '03, pp. 17–22, Association for Computational Linguistics.

[33] V. Rus, B. Wyse, P. Piwek, M. Lintean, S. Stoyanchev, and C. Moldovan, "The first question generation shared task evaluation challenge," in *Proceedings of the 6th International Natural Language Generation Conference*. 2010, INLG '10, pp. 251–257, Association for Computational Linguistics.

[34] H. Kunichika, T. Katayama, T. Hirashima, and A. Takeuchi, "Automated question generation methods for intelligent english learning systems and its evaluation," in *Proceedings of the 11th International Conference on Computers in Education*, 2003, ICCE '03.

[35] M. Heilman and N. A. Smith, "Good question! statistical ranking for question generation," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 2010, HLT '10, pp. 609–617, Association for Computational Linguistics.

[36] L. Vanderwende, "The importance of being important: Question generation," in *Workshop on the Question Generation Shared Task and Evaluation Challenge*, 2008.

[37] X. Yao, G. Bouma, and Y. Zhang, "Semantics-based question generation and implementation," *Dialogue and Discourse*, vol. 3, pp. 11–42, 2012.

[38] A. Ritter, C. Cherry, and W. B. Dolan, "Data-driven response generation in social media," in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. 2011, EMNLP '11, pp. 583–593, Association for Computational Linguistics.

[39] Y. W. Wong and R. J. Mooney, "Learning for semantic parsing with statistical machine translation," in *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. 2006, HLT-NAACL '06, pp. 439–446, Association for Computational Linguistics.

[40] L Nio, S. Sakti, G. Neubig, T. Toda, and S. Nakamura, "Utilizing human-to-human conversation examples for a multi domain chat-oriented dialog system," *IEICE Transactions*, vol. 97–D, no. 6, pp. 1497–1505, 2014.

[41] D. Marcu and W. Wong, "A phrase-based, joint probability model for statistical machine translation," in *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*. 2002, EMNLP '02, pp. 133–139, Association for Computational Linguistics.

[42] P. Koehn, H. Hoang, A. Birch, B. C. Callison, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. 2007, ACL '07, pp. 177–180, Association for Computational Linguistics.

[43] Ronan Collobert and Jason Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. of the 25th International Conference on Machine Learning*. 2008, ICML '08, pp. 160–167, ACM.

[44] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. of the 26th International Conference on Neural Information Processing Systems*. 2013, NIPS '13, pp. 3111–3119, Curran Associates Inc.

[45] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin, "A neural probabilistic language model," *The Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, mar 2003.

[46] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *Proc. of the 2013 Workshop on Automatic Speech Recognition and Understanding*, 2013, ASRU'13.

[47] Yuchen Fan, Yao Qian, Feng-Long Xie, and Frank K. Soong, "TTS synthesis with bidirectional LSTM based recurrent neural networks," in *The 15th Annual Conference of the International Speech Communication Association*, 2014, INTERSPEECH'14, pp. 1964–1968.

[48] Shotaro Misawa, Motoki Taniguchi, Yasuhide Miura, and Tomoko Ohkuma, "Character-based bidirectional lstm-crf with words and characters for japanese named entity recognition," in *Proc. of the First Workshop on Subword and Character Level Models in NLP*. 2017, pp. 97–102, ACL.

[49] Bill Y. Lin, Frank Xu, Zhiyi Luo, and Kenny Zhu, "Multi-channel bilstm-crf model for emerging named entity recognition in social media," in *Proc. of the 3rd Workshop on Noisy User-generated Text*. 2017, pp. 160–165, ACL.

[50] M. Freitag and Y. Al-Onaizan, "Beam search strategies for neural machine translation," in *Proceedings of the 1st Workshop on Neural Machine Translation*. 2017, pp. 56–60, Association for Computational Linguistics.

[51] K. Shinzato and Y. Oyamada, "What do people write in reviews for sellers? investigation and development of an automatic classification system," *Journal of Natural Language Processing*, vol. 25, no. 1, 2018.

[52] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, "Opennmt: Open-source toolkit for neural machine translation," in *Proceedings of the 55th Annual Meeting of the ACL on Demonstration Sessions*. 2017, ACL '17, pp. 67–72, Association for Computational Linguistics.

[53] L Tan and S. Pal, "Manawi: Using multi-word expressions and named entities to improve machine translation," in *Proceedings of the 9th Workshop on Statistical Machine Translation*. 2014, pp. 201–206, Association for Computational Linguistics.

[54] R. Al-Rfou, B. Perozzi, and S. Skiena, "Polyglot: Distributed word representations for multilingual nlp," in *Proceedings of the 17th Conference on Computational Natural Language Learning*. 2013, CoNLL '13, pp. 183–192, Association for Computational Linguistics.

[55] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.