A Benchmark Suite of Japanese Natural Questions

Takuya Uematsu[1] **Hao Wang**[1] **Daisuke Kawahara**[1] **Tomohide Shibata**[2]
[1]Waseda University [2]LY Corporation
takuya1009@akane.waseda.jp conan1024hao@akane.waseda.jp
dkw@waseda.jp tomshiba@lycorp.co.jp

Abstract

To develop high-performance and robust natural language processing (NLP) models, it is important to have various question answering (QA) datasets to train, evaluate, and analyze them. Although there are various QA datasets available in English, there are only a few QA datasets in other languages. We focus on Japanese, a language with only a few basic QA datasets, and aim to build a Japanese version of Natural Questions (NQ) consisting of questions that naturally arise from human information needs. We collect natural questions from query logs of a Japanese search engine and build the dataset using crowdsourcing. We also re-define the dataset specification of the original NQ to construct Japanese Natural Questions (JNQ). Furthermore, we construct a Japanese version of BoolQ (JBoolQ), which is derived from NQ and consists of yes/no questions. JNQ consists of 16,871 questions, and JBoolQ consists of 6,467 questions. We also define two tasks from JNQ and one from JBoolQ and establish baselines using competitive methods drawn from related literature. We hope that these datasets will facilitate research on QA and NLP models in Japanese. We will make JNQ and JBoolQ publicly available.

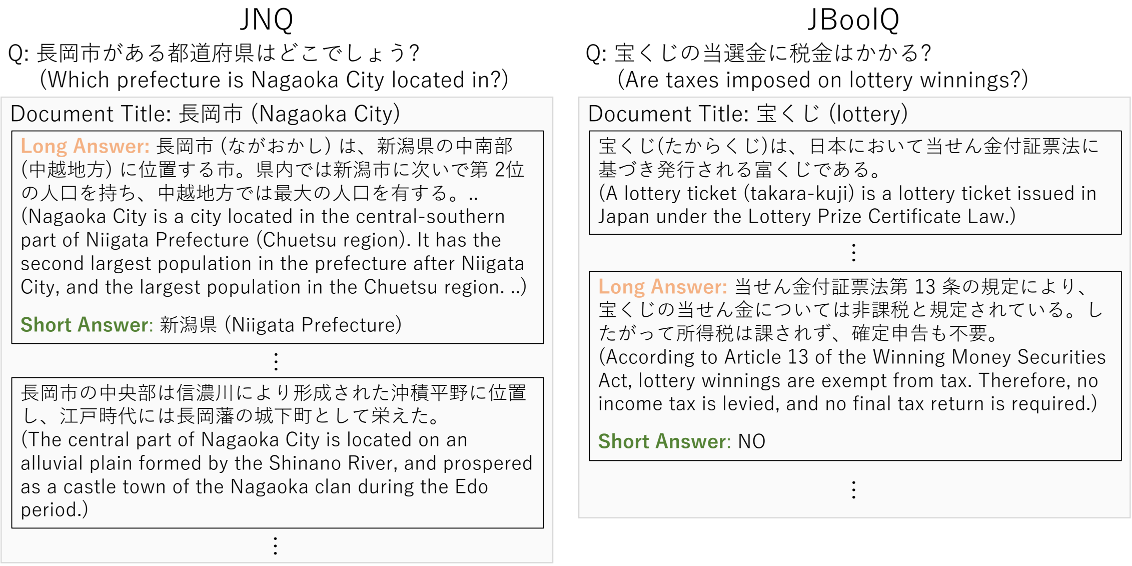
1 Introduction

To develop high-performance and robust natural language processing (NLP) models, it is important to have various question answering (QA) datasets to train, evaluate, and analyze them. There are diverse extractive and generative QA datasets that require many techniques and knowledge to solve, such as multi-hop inference (Yang et al., 2018) and real-world knowledge (Dua et al., 2019). There have been some studies to solve many QA tasks in an integrated manner, rather than solving them individually, such as Unified QA (Khashabi et al., 2020) and FLAN (Wei et al., 2022). However, such an integrated analysis is possible only in English but not in other languages because of the lack of QA datasets. This study focuses on Japanese, which has only a few basic QA datasets, such as JSQuAD (Kurihara et al., 2022), JaQuAD (So et al., 2022), and JAQKET (Suzuki et al., 2020).

In this paper, we focus on Natural Questions (NQ) (Kwiatkowski et al., 2019), which consist of questions that arise naturally from human information needs, as a critical QA dataset that does not exist in Japanese. QA datasets such as SQuAD (Rajpurkar et al., 2016) have the problem of annotation artifacts (Gururangan et al., 2018) because the questions are manually created by annotators, which are not natural. In contrast, NQ uses queries entered by users in a search engine, which are considered natural questions. One possible approach to creating a Japanese version of NQ is translating the original NQ dataset into Japanese. However, we do not use translation due to concerns about the unnaturalness of translated sentences, which can result from differences in grammar and other linguistic factors, as well as potential cultural differences between Japan and other countries. Instead, we build and publish Japanese Natural Questions (JNQ) using query logs from a Japanese search engine. We also re-define the dataset specification of the original NQ to obtain a better NQ dataset. Kwiatkowski et al. (2019) have hired trained annotators to build the NQ dataset, but for JNQ, we use crowdsourcing to reduce costs. This method can be applied to any language in which search engine query logs are available.

In addition to JNQ, we build JBoolQ, a Japanese version of BoolQ (Clark et al., 2019). BoolQ is derived from NQ and consists of yes/no questions. JBoolQ questions and yes/no answers are collected in the same way as JNQ. In the original BoolQ, there are only two options: “yes” or “no”. However, to make the setting more realistic, we add an option of “unable to answer” to JBoolQ, represented as “NONE”. This makes our dataset more challenging than the original BoolQ.

*Proceedings of the 13th Joint Conference on Lexical and Computational Semantics (*SEM 2024)*, pages 58 - 68
June 20-21, 2024 ©2024 Association for Computational Linguistics



JNQ

Q: 長岡市がある都道府県はどこでしょう？
(Which prefecture is Nagaoka City located in?)

Document Title: 長岡市 (Nagaoka City)

Long Answer: 長岡市 (ながおかし) は、新潟県の中南部 (中越地方) に位置する市。県内では新潟市に次いで第 2位の人口を持ち、中越地方では最大の人口を有する。..
(Nagaoka City is a city located in the central-southern part of Niigata Prefecture (Chuetsu region). It has the second largest population in the prefecture after Niigata City, and the largest population in the Chuetsu region. ..)

Short Answer: 新潟県 (Niigata Prefecture)

⋮

長岡市の中央部は信濃川により形成された沖積平野に位置し、江戸時代には長岡藩の城下町として栄えた。
(The central part of Nagaoka City is located on an alluvial plain formed by the Shinano River, and prospered as a castle town of the Nagaoka clan during the Edo period.)

⋮

JBoolQ

Q: 宝くじの当選金に税金はかかる？
(Are taxes imposed on lottery winnings?)

Document Title: 宝くじ (lottery)

宝くじ(たからくじ)は、日本において当せん金付証票法に基づき発行される富くじである。
(A lottery ticket (takara-kuji) is a lottery ticket issued in Japan under the Lottery Prize Certificate Law.)

⋮

Long Answer: 当せん金付証票法第 13 条の規定により、宝くじの当せん金については非課税と規定されている。したがって所得税は課されず、確定申告も不要。
(According to Article 13 of the Winning Money Securities Act, lottery winnings are exempt from tax. Therefore, no income tax is levied, and no final tax return is required.)

Short Answer: NO

⋮

Figure 1: Examples of JNQ and JBoolQ.

In consequence, JNQ contains 16,871 queries and 80,288 paragraphs. JBoolQ, combined with the JNQ yes/no questions, contains 6,467 queries and 31,677 paragraphs. Examples of JNQ and JBoolQ are shown in Figure 1.

Furthermore, we define three tasks using the two datasets as a new QA benchmark in Japanese: long answer extraction, short answer extraction, and yes/no answer identification (BoolQ). We also evaluate these tasks with their respective baselines. JNQ and JBoolQ will be available online.

2 Related Work

Existing QA datasets can be broadly categorized into those where the questions are natural and those where they are not.

QA datasets where the questions are not natural mainly include SQuAD (Rajpurkar et al., 2016) and SQuAD 2.0 (Rajpurkar et al., 2018). The questions in these datasets are not natural because annotators create them after reading a paragraph. Therefore, annotation artifacts in the created questions and lexical overlap between questions and paragraphs are problematic when using these datasets.

Natural Questions (Kwiatkowski et al., 2019) and BoolQ (Clark et al., 2019) are QA datasets that contain natural questions. To build these datasets, search engine query logs are used to collect natural questions arising from human information needs. The documents are Wikipedia articles, and the answers consist of long answers (e.g., paragraphs or tables) and short answers (spans or Yes/No). Other datasets that collect questions from query logs include WikiQA (Yang et al., 2015) and MS MARCO (Bajaj et al., 2018). In these datasets, the answer format differs from NQ and BoolQ, with a single sentence in the document or a hand-crafted summary.

QA datasets whose questions are not derived from query logs but are claimed to be natural include TyDi QA (Clark et al., 2020), Icelandic NQ (Snæbjarnarson and Einarsson, 2022), and Russian BoolQ (Glushkova et al., 2021). In these datasets, annotators are given a prompt consisting of a part or summary of a document and asked to think of a question that cannot be answered by reading only the prompt. These questions are claimed to be “natural” because they are derived from what humans wanted to know about the prompt. However, they are not naturally occurring questions because the authors ask them to think of a question. Thus, we consider that they are not truly natural questions.

For non-English QA datasets, there are several multilingual QA datasets, such as TyDi QA (Clark et al., 2020), MLQA (Lewis et al., 2020), XOR QA (Asai et al., 2021), and XQuAD (Artetxe et al., 2020). However, only approximately half of them include Japanese. Due to the lack of diverse datasets in Japanese, we construct Japanese Natural Questions from scratch.

3 Japanese Natural Questions

Natural Questions (NQ) (Kwiatkowski et al., 2019) is a dataset that focuses on the ability to answer natural questions by reading documents. Each instance consists of a quadruple of a question, a document, a long answer, and a short answer. The

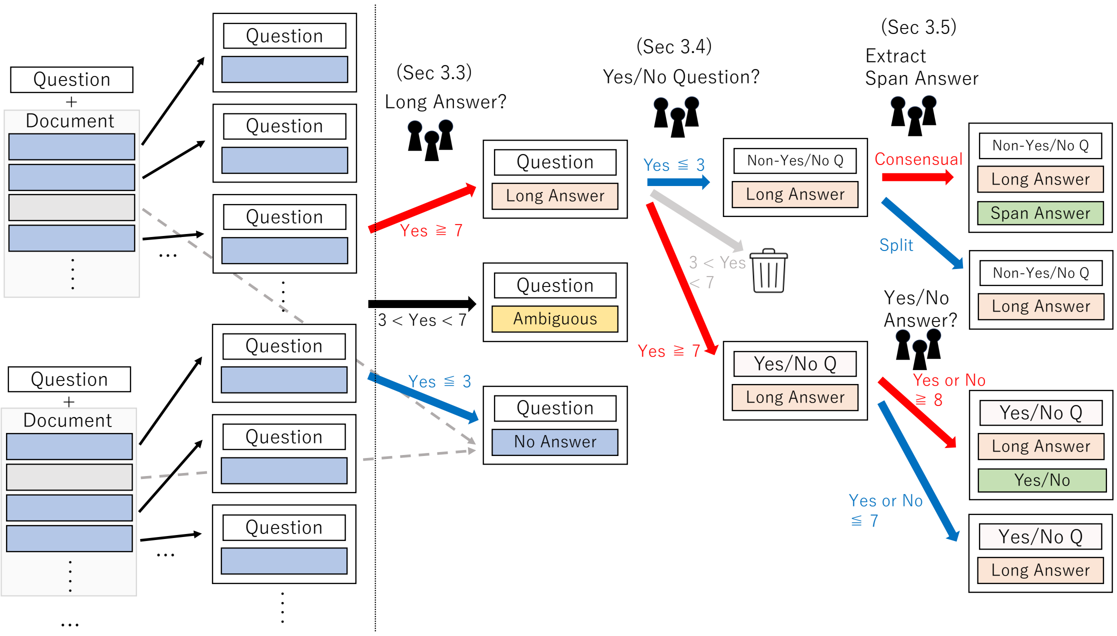


Figure 2: Construction flow of Japanese Natural Questions.

questions are collected from search engine query logs. The documents are Wikipedia articles, with one document provided for each question. The long answer is a paragraph or table in a document containing enough information to infer the answer. The short answer is the shortest possible answer to the question and is a span in the document.

Japanese Natural Questions (JNQ), like NQ, consists of quadruples of a question, a document, long answer(s), and short answer(s). The questions are extracted from search engine query logs, and the documents are Japanese Wikipedia articles. The long answers and short answers are obtained using crowdsourcing. By using crowdsourcing, it is possible to construct a dataset at a low cost and with some quality level without expert annotators. We limit the long answers only to paragraphs to simplify the task, considering that dataset construction is conducted using crowdsourcing. Although NQ has a strict restriction that there is at most one long answer in a document, there are often multiple paragraphs containing answers. Therefore, JNQ allows for scenarios with multiple long answers to a single question.

We describe each stage of building JNQ below. In crowdsourcing, 10 crowdworkers are assigned to deal with a task to build a high-quality benchmark. In cases where ambiguity is detected due to diverging opinions among crowdworkers at each stage, such instances are not incorporated into JNQ. We illustrate the construction flow in Figure 2.

3.1 Question and Document Collection

Question candidates of JNQ are taken from the search query logs accumulated by a company[1]. When people search, they sometimes use word sequences instead of full sentences. Such queries are specific to search engines and may include non-questions. Therefore, queries with spaces are excluded from the pool of question candidates[2]. Furthermore, short queries are often not in the form of questions; therefore, only queries composed of eight or more words are extracted[3]. Subsequently, we prepare the following question patterns and extract queries that match any of them.

1. Contains "は" (Japanese topic marker) + an interrogative word
2. The final character is "?"
3. Contains the specific word such as "意味" (meaning), "方法" (method), and "理由" (reason).

We perform a Google search with the question candidates obtained above. If there is a Wikipedia article within the top five search results, we select the top-ranked article as the document. Question candidates for which there are no Wikipedia articles within the top five search results are excluded.

[1] The name of the company will remain anonymous until the paper is accepted.

[2] Japanese is a language that does not use spaces between words, and Japanese sentences usually do not contain spaces.

[3] Word segmentation is performed using the morphological analyzer Juman++: https://github.com/ku-nlp/jumanpp.

3.2 Good Question Identification

The extracted question candidates contain non-questions and inappropriate questions. Therefore, we use crowdsourcing to obtain good questions. A good question is one that inquires about facts, methods, causes, or reasons. A bad question is ambiguous, based on incorrect assumptions, soliciting opinions, asking about the title of a work, or posing questions with answers that vary depending on the timing. 10 crowdworkers judge whether the given question is good or bad. Among the 10 crowdworkers, question candidates that are judged as good questions by six or more workers are adopted as questions for JNQ. Examples of good questions are provided in Appendix A. Examples judged as bad questions are "今日はどこに行こうか？" (Where shall we go today?) and "Amazon支払い方法が承認されません" (The Amazon payment method is not approved).

3.3 Long Answer Identification

Through crowdsourcing, we extract paragraphs from the document that contain sufficient information to answer a question and designate them as long answers. We provide crowdworkers with a maximum of five paragraphs to reduce annotation costs. These five paragraphs consist of the document's first paragraph and four paragraphs (excluding the first one) that have high relevance to the snippet obtained from the Google search conducted in Section 3.1. This is because the first paragraph, which usually provides an overview, and the paragraphs with high relevance to the snippet are likely to contain the answer. The paragraphs that are not included in these five paragraphs are identified as not containing the long answer and are accordingly labeled as "NONE". The relevance is calculated by the cosine similarity between the snippet and a paragraph, with both represented as bag-of-words vectors. We illustrate the paragraph selection process in Figure 3.

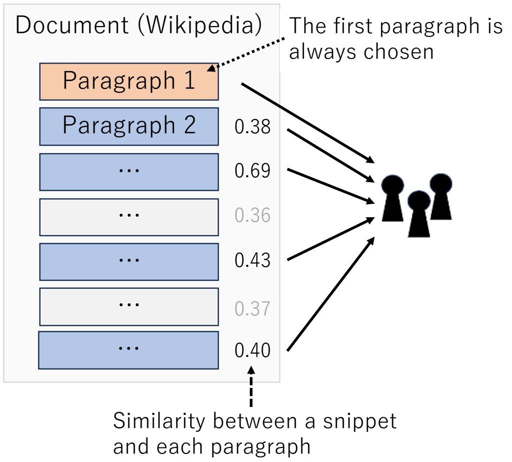

Document (Wikipedia)
The first paragraph is always chosen
Paragraph 1
Paragraph 2
0.38
···
0.69
···
0.36
···
0.43
···
0.37
···
0.40
Similarity between a snippet and each paragraph

Figure 3: An illustration for choosing paragraphs from documents to ask crowdworkers whether they qualify as long answers.

We provide a question and each paragraph to 10 crowdworkers, prompting them to make a binary choice on whether the paragraph contains "sufficient information to infer an answer to the question" or not. We classify the paragraphs into three groups based on the votes of the 10 workers. If seven or more "Yes" votes are collected, we categorize the paragraph as a long answer and assign it the label "EXIST". If four to six "Yes" votes are collected, we categorize the paragraph as ambiguous in terms of being a long answer and label it as "AMBIGUOUS". Excluding this paragraph during the training process can help reduce noise. If three or fewer votes are collected, we categorize the paragraph as lacking a long answer and label it as "NONE". Since the judgment is done on a per-paragraph basis, multiple paragraphs may be classified as long answers for a single question, or there may be no long answer at all. If none of the paragraphs within these five paragraphs qualifies as the long answer, we infer that the document does not contain a long answer to the question.

3.4 Yes/No Question Identification

In the following step, detailed in Section 3.5, we extract short answers from paragraphs designated as long answers. The task of short answer extraction varies depending on whether the question is a yes/no question. Therefore, we first crowdsource the judgment of whether the question is a yes/no question. If seven or more crowdworkers judge the question to be a yes/no question, the question is considered as a yes/no question. If a question receives between four and six votes, we remove it from the dataset due to its ambiguity.

3.5 Short Answer Identification

We categorize the cases based on whether the question is a yes/no question. For each category, we obtain a short answer, i.e., a yes/no answer or a span answer, using the following procedure.

Yes/No Answer Identification If the question is a yes/no question, crowdworkers judge whether the answer is "YES" or "NO" based on the paragraph of a long answer. If more than seven crowdworkers judge the answer as either "YES" or "NO", the

	Number	Length (# of chars)		
		Mean	Max	Min
Question	16,871	17.7	50	8
Paragraph	192,514	159.0	999	10
Span answer	5,463	9.6	180	1

Table 1: Numbers and lengths of questions and paragraphs, and short answers in JNQ. The paragraphs in this table refer to all paragraphs, including unannotated paragraphs (i.e., considered as no long answer).

Long →	EXIST			AMBIGUOUS	NONE
Short →	Span	Yes/No	NONE		
	5,463	143	2,280	10,866	61,536

Table 2: Statistics of paragraphs in JNQ. The total number of paragraphs is 80,288.

answer is considered as a short answer. Paragraphs with seven or fewer "YES" or "NO" votes are considered ambiguous paragraphs, and a "NONE" label is assigned to the short answer. In other words, this paragraph is judged to have only a long answer.

Span Answer Identification If the question is not a yes/no question, we ask 10 crowdworkers to extract a span answer from the paragraph. If there is no span answer in the paragraph, crowdworkers judge it as "NONE". We aggregate the 10 answers by majority voting. As a pre-process, if one answer is subsumed by another, the votes are added to the shorter one. If there is a tie with multiple short answers receiving the most votes, the shortest one is chosen. Furthermore, answers that receive only one vote are considered unreliable and are not adopted.

4 Japanese BoolQ

BoolQ (Clark et al., 2019) is a QA dataset focusing on natural yes/no questions. It contains many non-factoid questions that require a wide range of inferential abilities to answer. Each instance consists of a question, a paragraph (equivalent to a long answer in NQ), and an answer (yes/no). The questions and paragraphs are extracted from search engine query logs and Wikipedia articles, like NQ. BoolQ adopts only the questions with either yes or no answers and pairs them with not a whole document but a paragraph to simplify the specification.

Japanese BoolQ (JBoolQ) consists of a question, a document, a long answer, and a yes/no answer, like yes/no questions in JNQ. Unlike BoolQ, each question may have multiple long answers, and the answers can include "NONE", which means unanswerable, in addition to yes/no. Therefore, it is more challenging than BoolQ, and a deeper understanding of the documents is required to answer the questions.

# of long answers	Number	Ratio
0	11,126	65.9%
1	4,117	24.4%
2	1,203	7.1%
3	344	2.0%
4	74	0.4%
5	7	0.04%
Total	16,871	100%

Table 3: Distribution of the number of long answers per question in JNQ.

We construct JBoolQ using basically the same procedure as JNQ. Since the ratio of yes/no questions in JNQ is only around 1%, for JBoolQ, we collect questions from a larger query log pool than JNQ. The construction procedure is as follows. The details of each step are described in Section 3.

1. Question and document collection[4]
2. Good question identification
3. Yes/No question identification
4. Long answer identification
5. Yes/No answer identification

Compared to JNQ, the order of yes/no question identification and long answer identification is reversed to narrow down the candidates to the target yes/no questions at an early stage and reduce the annotation cost later. Finally, we merge the yes/no questions in JNQ into JBoolQ.

5 Analysis

In this section, we analyze JNQ and JBoolQ.

5.1 JNQ

Statistics JNQ contains 16,871 questions. Table 1 shows the average, maximum, and minimum numbers of characters in the questions, paragraphs, and short answers. Statistics on the paragraphs are shown in Table 2. In JNQ, multiple paragraphs can be a long answer to a single question. The distribution of the number of long answers per question is shown in Table 3. Questions with multiple long answers account for approximately 10% of all questions and 28% of the questions with long answers.

[4] We change the conditions of JNQ to extract yes/no questions as follows: more than six words and ending with "?" or "か" (Japanese interrogative particle).

Type	Example
What	歌手「矢沢永吉」が1978年にヒットした曲は?
(39%)	What was the song that the singer Eikichi Yazawa had a hit with in 1978?
Where	「伯方の塩」で知られる伯方島があるのはどこ?
(12%)	Where is Hakata Island, known for “Hakata Salt”?
When	パスポートに菊が描かれたのはいつ
(4%)	When was the chrysanthemum depicted on passports?
Why	日本にはなぜ四季があるのか
(4%)	Why does Japan have four seasons?
Who	「青の時代」といった、20世紀を代表する画家は誰でしょう?
(3%)	Who is the iconic painter of the 20th century known for the ’Blue Period’?
How	スマートフォンでqrコードを読み取る方法
(31%)	How to read qr code with smartphone
Yes/No	源泉徴収票は市役所でもらえる?
(3%)	Can I obtain a withholding slip at the city hall?
Other	冬に卵を生で食べられる期間は何日
(4%)	How long can eggs be eaten raw in winter?

Table 4: Question types of JNQ.

	Number	Length (# of chars)		
		Mean	Max	Min
Question	6,467	11.4	48	6
Paragraph	27,954	171.7	988	21

Table 5: Numbers and lengths of questions and paragraphs in JBoolQ.

Long →	EXIST		AMBIGUOUS	NONE
Short →	Yes/No	NONE		
	1,742	833	3,723	25,379

Table 6: Statistics of paragraphs in JBoolQ. The total number of paragraphs is 31,677.

# of long answers	Number	Ratio
0	4,649	71.9%
1	1,252	19.4%
2	414	6.4%
3	117	1.8%
4	31	0.5%
5	4	0.06%
Total	6,467	100%

Table 7: Distribution of the number of long answers per question in JBoolQ.

Question Type We sampled 100 questions from JNQ and classified them according to which wh-word they begin with when translated into English. The results are shown in Table 4. The most common question type is “What”, accounting for 39%. The next most common question is “How”, accounting for 31%. Of the questions asking “How”, 84% of the questions are about “How to”. In NQ, questions starting with “How to” account for less than 1% of the total, and thus there are more ”How to” questions in JNQ, which can be considered more difficult to answer than fact-seeking ones.

Lexical Overlap We investigated lexical overlap. Lexical overlap refers to the ratio of overlapping words between a paragraph and a question. It is reported that when this ratio is high, the model can easily provide answers (Clark et al., 2020). Each question and paragraph pair of JNQ was segmented at the word level[5], and lexical overlap was calculated. Lexical overlap of JNQ is 59.4%, which is much lower than 79.5% observed in Japanese SQuAD (JSQuAD). This result indicates that we address, to some extent, the issue of annotation artifacts, which are common in datasets such as SQuAD, where an annotator is asked to create a question after reading a paragraph.

5.2 JBoolQ

Statistics JBoolQ contains 6,467 questions. Table 5 shows the average, maximum, and minimum numbers of characters in the questions and paragraphs. The average length of the questions is shorter than JNQ. This is because when extracting candidate questions from query logs, JNQ extracted queries with eight or more words, while JBoolQ extracted queries with six or more words to obtain more yes/no questions. Statistics on the paragraphs are shown in Table 6. The distribution of the number of long answers, shown in Table 7, is similar to JNQ.

Question Type We sampled 100 questions from JBoolQ and classified them according to their question types. We basically adopted the classification method used in BoolQ but added two categories: “Possibility” and “Necessity”. The results are shown in Table 8. Questions asking facts about a specific entity occupy 31%, which is the most

[5] We used MeCab + IPAdic (`https://taku910.github.io/mecab/`) for word segmentation.

Type	Example
Possibility (23%)	新幹線で携帯充電できる? Can I charge my cell phone on the Shinkansen?
Necessity (11%)	履歴書に印鑑は必要か Do I need a seal on my resume?
Definitional (7%)	ナショナルとパナソニックは同じ? Are "National" and "Panasonic" the same?
Existence (4%)	国会議事堂の中に保育園ある? Is there a daycare center in the Capitol?
Other General Fact (24%)	疲れで熱は出る? Does fatigue cause fever?
Other Entity Fact (31%)	久能山東照宮は神社? Kunouzan Toshogu is a shrine?

Table 8: Question types of JBoolQ.

Task	Train	Dev	Test
Long Answer Extraction	13,496	1,687	1,688
Short Answer Extraction	6,158	789	761
Yes/No Answer Identification	22,357	2,791	2,806

Table 9: Statistics of the three tasks. The number of long answer extraction refers to the number of questions, and the numbers of the other tasks refer to the number of instances.

common. Questions asking about "Possibility" and "Necessity", newly added categories in JBoolQ, account for 23% and 11%, respectively, corresponding to a total of 1/3 of the whole dataset.

6 Experiments

6.1 Experimental Setup

We define three tasks to use JNQ and JBoolQ as a benchmark for evaluating QA systems. From JNQ, we introduce the following two tasks: long answer extraction, short answer extraction. From JBoolQ, we introduce the task of yes/no answer identification. We also establish baselines using competitive methods drawn from related literature. We implement hyperparameter searches and report the best scores. We list the statistics of the tasks in Table 9.

Long Answer Extraction Unlike NQ, in our dataset, there can be multiple long answers or no long answer in a document. Thus, we consider long answer extraction as a paragraph-based multi-label classification task. *Given a question and a document, a system tries to select all paragraphs with long answers.* We use precision, recall, and F1 scores for evaluation metrics.

We introduce a baseline that considers the task a binary classification problem. For each paragraph in the document, we input the question-paragraph pair into the model and binarily decide whether the paragraph is a long answer. We use Japanese BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) as base models[6]. We use two kinds of training sets in our experiments: (1) the paragraphs collected in Section 3.3, which contain positive examples and hard negative examples (challenge candidates, which have high relevance to the snippet but are considered as no long answer), and (2) all paragraphs in the documents. The ambiguous paragraphs are excluded from both. For testing, we use all paragraphs in the documents, aiming to be close to real extraction scenarios.

We also evaluate human performance using crowdsourcing in the same way as the dataset construction process. We asked 10 annotators to answer. If seven or more annotators agree, it is considered that the paragraph is a long answer; otherwise, it is not. Due to cost reasons, we sampled 100 questions for human evaluation instead of using the whole test set.

Short Answer Extraction For short answer extraction, we target question-paragraph pairs labeled as being present for long answers. Following NQ, we exclude yes/no questions. In practice, we treat this task as a SQuAD 2.0 (Rajpurkar et al., 2018) like task. *Given a question-paragraph pair, a system tries to extract a span as the short answer from the paragraph.* If the paragraph has no short answer, we regard this question as unanswerable and make the target span an empty string. We use exact match (EM) and character-based F1 scores for evaluation metrics.

We treat short answer extraction as a classification problem of whether each token in a paragraph is an answer span's start/end position. We use BERT and RoBERTa as base models.

We also evaluate human performance using crowdsourcing on the whole test set. We asked three annotators to answer and average their scores.

Yes/No Answer Identification As described in Section 4, unlike BoolQ, our JBoolQ dataset contains three kinds of labels: "YES", "NO", and "NONE". This makes our task a multiclass classification problem. *Given a question-paragraph pair, a system tries to answer Yes/No/None.* We use precision, recall, and F1 scores on labels "YES" and "NO" for evaluation metrics.

We use BERT and RoBERTa as base models. Since the instances with yes/no answers are scarce, we oversample these instances five times.

[6] We use the transformers library provided by Hugging Face. `https://github.com/huggingface/transformers`

	Trained on Only Hard Negatives						Trained on All Data					
	Dev			Test			Dev			Test		
Model	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Tohoku-BERT-base[1]	36.6	74.6	49.1	35.0	72.3	47.2	53.1	67.4	59.4	51.2	68.3	58.5
Tohoku-BERT-large[2]	39.6	70.9	50.8	42.1	72.1	53.2	53.9	67.5	59.9	56.8	66.2	61.2
Waseda-RoBERTa-base[3]	42.5	73.8	53.9	44.6	74.8	55.9	**63.7**	**73.0**	**68.0**	**64.2**	**73.4**	**68.5**
Waseda-RoBERTa-large[4]	**47.1**	**76.2**	**58.2**	**48.6**	**80.9**	**60.7**	57.9	51.4	54.5	57.9	48.3	52.7
Human	-	-	-	-	-	-	-	-	-	46.3	75.8	57.5

Table 10: Performance on long answer extraction. We list precision (P), recall (R), and F1 of baselines and human annotators. Human evaluation was conducted by sampling 100 questions from the test set.

	Dev		Test	
Model	EM	F1	EM	F1
Tohoku-BERT-base	23.3	33.4	23.1	31.3
Tohoku-BERT-large	23.1	32.9	23.3	31.0
Waseda-RoBERTa-base	41.1	49.9	41.7	50.1
Waseda-RoBERTa-large	**45.5**	**53.4**	**45.7**	**53.9**
Human	-	-	51.1	62.5

Table 11: Performance on short answer extraction.

	Dev			Test		
Model	P	R	F1	P	R	F1
Tohoku-BERT-base	63.4	**59.6**	61.4	62.5	52.5	57.0
Tohoku-BERT-large	66.0	54.1	59.5	65.1	50.6	56.9
Waseda-RoBERTa-base	58.1	56.8	57.5	59.5	56.2	57.8
Waseda-RoBERTa-large	**68.4**	57.9	**62.7**	**65.5**	**57.4**	**61.2**
Human	-	-	-	75.8	73.0	74.4

Table 12: Performance on yes/no answer identification.

We also evaluate human performance by asking 10 crowdworkers to conduct the following two tasks. First, they check if a paragraph is a long answer in a similar way to long answer extraction. Second, the workers judge “YES”, “NO”, or “NONE” for a paragraph that is judged to be a long answer. The answer with the most votes is adopted, and if the number of the most votes is the same, “NONE” is adopted.

6.2 Results

Long Answer Extraction We show the results of long answer extraction in Table 10. The models show high recall but low precision when trained on only hard negative examples. The models’ precision becomes much higher when trained on all data, indicating unlabeled negative examples are also helpful to training.

Human annotators performed poorly in precision for this task. This also indicates the possibility of there being a few paragraphs with a long answer within the unlabeled paragraphs (except five paragraphs given to the crowdworkers). To tackle this problem, a possible way is to provide the crowdworkers with paragraphs except for the five paragraphs judged as “long answers” by the models and ask them to determine whether they are long answers. We leave this exploration for future work.

Short Answer Extraction We show the results of short answer extraction in Table 11. Waseda-RoBERTa-base and Waseda-RoBERTa-large perform well, but the scores are very inferior to the human performance. Tohoku-BERT-base and Tohoku-BERT-large perform poorly. When examining the outputs, we found that Tohoku-BERTs sometimes extract the entire paragraph as predictions, which leads to underperformance. Since the paragraph is a long answer, extracting the entire paragraph could also be considered correct, but it is wrong according to our task definition. We speculate that insufficient data caused this phenomenon, considering our data is only one-tenth of JSQuAD (Kurihara et al., 2022).

Yes/No Answer Identification We show the results of yes/no answer identification in Table 12. The models show high precision and relatively low recall scores, indicating that they predict a large proportion of yes/no instances as “NONE”. “NONE” instances make our task more challenging than the original BoolQ, which makes our benchmark more valuable since advanced training techniques are needed to overcome the unbalanced data distribution and improve model performance.

Human annotators could recognize more yes/no answers correctly than the models. This leads to a higher recall.

7 Conclusion

We constructed two QA datasets: Japanese Natural Questions (JNQ) and Japanese BoolQ (JBoolQ). The questions in these datasets are collected from query logs from a Japanese search engine and are natural, derived from human information needs.

The annotation process was conducted through crowdsourcing. We also defined a total of three tasks, including long answer extraction, short answer extraction, and yes/no answer identification. We evaluated the performance of the baseline models. The constructed datasets can be used for training, evaluating, and analyzing QA and NLP models and are expected to facilitate these studies in Japanese.

Acknowledgements

This work was carried out in collaboration with LY Corporation and Waseda University.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Akari Asai, Jungo Kasai, Jonathan Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. 2021. XOR QA: Cross-lingual open-retrieval question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 547–564, Online. Association for Computational Linguistics.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset.

Xilun Chen, Kushal Lakhotia, Barlas Oguz, Anchit Gupta, Patrick Lewis, Stan Peshterliev, Yashar Mehdad, Sonal Gupta, and Wen-tau Yih. 2022. Salient phrase aware dense retrieval: Can a dense retriever imitate a sparse one? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 250–262, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Taisia Glushkova, Alexey Machnev, Alena Fenogenova, Tatiana Shavrina, Ekaterina Artemova, and Dmitry I. Ignatov. 2021. DaNetQA: A yes/no question answering dataset for the russian language. In *Lecture Notes in Computer Science*, pages 57–68. Springer International Publishing.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. 2022. JGLUE: Japanese general language understanding evaluation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2957–2966, Marseille, France. European Language Resources Association.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Sewon Min, Jordan Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, Colin Raffel, Adam Roberts, Tom Kwiatkowski, Patrick Lewis, Yuxiang Wu, Heinrich Küttler, Linqing Liu, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Sohee Yang, Minjoon Seo, Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Edouard Grave, Ikuya Yamada, Sonse Shimaoka, Masatoshi Suzuki, Shumpei Miyawaki, Shun Sato, Ryo Takahashi, Jun Suzuki, Martin Fajcik, Martin Docekal, Karel Ondrej, Pavel Smrz, Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen-tau Yih. 2021. Neurips 2020 efficientqa competition: Systems, analyses and lessons learned. In *Proceedings of the NeurIPS 2020 Competition and Demonstration Track*, volume 133 of *Proceedings of Machine Learning Research*, pages 86–111. PMLR.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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

Vésteinn Snæbjarnarson and Hafsteinn Einarsson. 2022. Natural questions in Icelandic. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4488–4496, Marseille, France. European Language Resources Association.

ByungHoon So, Kyuhong Byun, Kyungwon Kang, and Seongjin Cho. 2022. JaQuAD: Japanese Question Answering Dataset for Machine Reading Comprehension.

Masatoshi Suzuki, Jun Suzuki, Koji Matsuda, Kyosuke Nishida, and Naoya Inoue. 2020. Jaqket: Construction of a japanese qa dataset on the subject of quizzes. *Proceedings of Annual Meeting of the Association for Natural Language Processing*, 26:237–240.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

A Examples of Good Questions

Examples of good questions obtained in Section 3.2 are shown in Table 13.

Type	Example
Fact	ナスカの地上絵がある所はどこ?
	Where are the Nazca Lines?
Reason	ビール瓶の色が茶色なのはなぜでしょう?
	Why are beer bottles brown?
How to	ナスに油を吸わせない方法
	How to keep eggplant from absorbing oil?

Table 13: Examples of good questions.

B Open-Domain NQ

From JNQ, we additionally define the task of open-domain NQ tasks and establish baselines. We show the statistics of the task in Table 14.

Experimental Setup Following the EfficientQA competition (Min et al., 2021), which uses the NQ dataset for open-domain question answering, we use JNQ to conduct the same task. *Given a question, a system tries to output a short answer without reference.* We target questions labeled as being present for short answers and remove questions whose answers have more than three words because we considered these questions to be difficult to answer precisely. We use exact match (EM) for an evaluation metric.

We use retriever-reader models as baselines. We use TF-IDF and a DPR retriever (Karpukhin et al., 2020) for the retriever and a DPR reader for the reader. We first use the retriever to retrieve 100 relevant paragraphs to the question from a database of Wikipedia and then employ the reader to find the answer from the retrieved paragraphs. We use DPR checkpoints from the second AIO competition[7].

Results We show the results of open-domain NQ in Table 15. The TF-IDF retriever performs slightly better than DPR on the test set. We speculate that because the average length of the questions is relatively short, salient phrases and rare entities in the questions make DPR difficult to retrieve accurately (Chen et al., 2022). Additionally, we found that some questions are unsuitable for open-domain QA. For instance, there is no standard answer to questions such as “なぜ貧しい国はなくならないのか” (Why don’t poor countries disappear?) and “男の子の髪の毛の切り方” (How to cut a boy’s hair?). We plan to exclude these questions in future work.

Task	Train	Dev	Test
Open-Domain NQ	2,317	298	284

Table 14: Statistics of the task of open-domain NQ. The number refers to the number of instances.

	Dev	Test
	EM	
TF-IDF + DPR reader	30.2	**30.3**
DPR	**31.2**	29.9

Table 15: Performance on open-domain NQ.

[7] https://sites.google.com/view/project-aio/competition2