Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 2957–2966
Marseille, 20-25 June 2022
© European Language Resources Association (ELRA), licensed under CC-BY-NC-4.0

JGLUE: Japanese General Language Understanding Evaluation

[1]**Kentaro Kurihara,** [1]**Daisuke Kawahara,** [2]**Tomohide Shibata**
[1]Waseda University, [2]Yahoo Japan Corporation
{kkurihara@akane., dkw@}waseda.jp, tomshiba@yahoo-corp.jp

Abstract

To develop high-performance natural language understanding (NLU) models, it is necessary to have a benchmark to evaluate and analyze NLU ability from various perspectives. While the English NLU benchmark, GLUE (Wang et al., 2018), has been the forerunner, benchmarks are now being released for languages other than English, such as CLUE (Xu et al., 2020) for Chinese and FLUE (Le et al., 2020) for French; but there is no such benchmark for Japanese. We build a Japanese NLU benchmark, JGLUE, from scratch without translation to measure the general NLU ability in Japanese. We hope that JGLUE will facilitate NLU research in Japanese.

Keywords: GLUE, Japanese, NLU benchmark, text classification, sentence pair classification, QA

1. Introduction

To develop high-performance natural language understanding (NLU) models, it is necessary to have a benchmark (a set of datasets) to evaluate and analyze NLU ability from various perspectives. In the case of English, the GLUE (General Language Understanding Evaluation) benchmark (Wang et al., 2018) is publicly available. Once an NLU model that can achieve a certain level of high score on GLUE is developed, a more difficult benchmark, such as SuperGLUE (Wang et al., 2019), is released, creating a virtuous cycle of benchmark construction and NLU model development.

Along with the trend of active NLU studies in English, benchmarks for languages other than English have been constructed, including CLUE (Xu et al., 2020) for Chinese, FLUE (Le et al., 2020) for French, and KLUE (Park et al., 2021) for Korean.

Although there are many studies on Japanese, which is the 13th most spoken language in the world as of 2021, there is no benchmark such as GLUE. Japanese is linguistically different from English and other languages in the following aspects.

- The Japanese alphabet includes *hiragana*, *katakana*, Chinese characters, and the Latin alphabet.
- There are no spaces between words.
- The word order is relatively free.

Due to these differences, findings on English datasets are not necessarily applicable to Japanese. Given this situation, there is an urgent need to develop a benchmark for Japanese NLU. Although individual Japanese datasets, such as JSNLI (Yoshikoshi et al., 2020) and JSICK (Yanaka and Mineshima, 2021), have been constructed, their construction methods involve mainly machine translation or manual translation from English datasets. With either of these translation methods, the unnaturalness of a translated text and the cultural/social discrepancy between an original language (mostly English) and a target language (Japanese in our case) are major problems, as discussed in Clark et al. (2020) and Park et al. (2021). Although there are also Japanese datasets in specific domains, such as hotel reviews (Hayashibe, 2020) and the driving domain (Takahashi et al., 2019), these are not suitable for evaluating NLU ability in the general domain.

Task	Dataset	Train	Dev	Test
Text	MARC-ja	187,528	5,654	5,639
Classification	JCoLA	—	—	—
Sentence Pair	JSTS	12,451	1,457	1,589
Classification	JNLI	20,073	2,434	2,508
QA	JSQuAD	63,870	4,475	4,470
	JCommonsenseQA	9,012	1,119	1,118

Table 1: JGLUE overview.

In this study, we build a general NLU benchmark for Japanese, JGLUE, from scratch without translation. JGLUE is designed to cover a wide range of GLUE and SuperGLUE tasks and consists of three kinds of tasks: text classification, sentence pair classification, and QA, as shown in Table 1. Each task consists of multiple datasets. JGLUE is available at `https://randd.yahoo.co.jp/en/softwaredata#jglue`. We hope that this benchmark will facilitate NLU research in Japanese.

2. Related Work

The first benchmark for evaluating NLU models is GLUE, which consists of two kinds of tasks, i.e., sentence classification and sentence pair classification, and nine datasets in total. SuperGLUE is a more difficult benchmark than GLUE, which contains eight datasets. It keeps the most challenging dataset of GLUE, i.e., natural language inference (NLI), and adds more difficult tasks, such as QA and commonsense reasoning.

Such benchmark construction in English has stimulated the development of NLU models, including BERT (De-

Label	Train	Dev	Test	Total
positive	165,477	4,832	4,895	175,204
negative	22,051	822	744	23,617
Overall	187,528	5,654	5,639	198,821

Table 2: Statistics of MARC-ja.

vlin et al., 2019) and many extended models. This situation has caused a growing movement to build NLU benchmarks in many languages, such as CLUE, FLUE, KLUE, IndicGLUE (Kakwani et al., 2020), ARLUE (Abdul-Mageed et al., 2021), ALUE (Seelawi et al., 2021), and CLUB (Rodriguez-Penagos et al., 2021), in Chinese, French, Korean, Indian languages, Arabic, and Catalan. Multilingual benchmarks, such as XGLUE (Liang et al., 2020), XTREME (Hu et al., 2020), and XTREME-R (Ruder et al., 2021), have also been built. Although they contain datasets in various languages, only a few of them include Japanese.

3. JGLUE Benchmark

JGLUE consists of the tasks of text classification, sentence pair classification, and QA, as shown in Table 1. In the following sections, we explain how to construct the datasets for each task. As one of the text classification datasets, JCoLA (the Japanese version of CoLA (Warstadt et al., 2019), the Corpus of Linguistic Acceptability) will be provided by another research organization. Since it is still under construction, this paper does not explain it.

We use Yahoo! Crowdsourcing[1] for all crowdsourcing tasks in constructing each dataset.

3.1. MARC-ja

As one of the text classification datasets, we build a dataset based on the Multilingual Amazon Reviews Corpus (MARC) (Keung et al., 2020).

MARC is a multilingual corpus of product reviews with 5-level star ratings (1-5) on the Amazon shopping site. This corpus covers six languages, including English and Japanese. For JGLUE, we use the Japanese part of MARC and to make it easy for both humans and computers to judge a class label, we cast the text classification task as a binary classification task, where 1- and 2-star ratings are converted to “negative”, and 4 and 5 are converted to “positive”. We do not use reviews with a 3-star rating.

One of the problems with MARC is that it sometimes contains data where the rating diverges from the review text. This happens, for example, when a review with positive content is given a rating of 1 or 2. These data degrade the quality of our dataset.

To improve the quality of the dev/test instances used for evaluation, we crowdsource a positive/negative judgment task for approximately 12,000 reviews. We adopt only reviews with the same votes from 7 or more out of 10 workers and assign a label of the maximum votes to these reviews. We divide the resulting reviews into dev/test data.

We obtained 5,654 and 5,639 instances for the dev and test data, respectively, through the above procedure. For the training data, we extracted 187,528 instances directly from MARC without performing the cleaning procedure because of the large number of training instances. The statistics of MARC-ja are listed in Table 2. For the evaluation metric for MARC-ja, we use accuracy because it is a binary classification task of texts.

3.2. JSTS and JNLI

For the sentence pair classification datasets, we construct a semantic textual similarity (STS) dataset, JSTS, and a natural language inference (NLI) dataset, JNLI.

3.2.1. Overview

STS is a task of estimating the semantic similarity of a sentence pair. Gold similarity is usually assigned as an average of the integer values 0 (completely different meaning) to 5 (equivalent meaning) assigned by multiple workers through crowdsourcing.

NLI is a task of recognizing the inference relation that a premise sentence has to a hypothesis sentence. Inference relations are generally defined by three labels: “entailment”, “contradiction”, and “neutral”. Gold inference relations are often assigned by majority voting after collecting answers from multiple workers through crowdsourcing.

For the STS and NLI tasks, STS-B (Cer et al., 2017) and MultiNLI (Williams et al., 2018) are included in GLUE, respectively. As Japanese datasets, JSNLI (Yoshikoshi et al., 2020) is a machine translated dataset of the NLI dataset SNLI (Stanford NLI), and JSICK (Yanaka and Mineshima, 2021) is a human translated dataset of the STS/NLI dataset SICK (Marelli et al., 2014). As mentioned in Section 1, these have problems originating from automatic/manual translations. To solve this problem, we construct STS/NLI datasets in Japanese from scratch.

We basically extract sentence pairs in JSTS and JNLI from the Japanese version of the MS COCO Caption Dataset (Chen et al., 2015), the YJ Captions Dataset (Miyazaki and Shimizu, 2016).[2] Most of the sentence pairs in JSTS and JNLI overlap, allowing us to analyze the relationship between similarities and inference relations for the same sentence pairs like SICK and JSICK.

The similarity value in JSTS is assigned a real number from 0 to 5 as in STS-B. The inference relation in JNLI is assigned from the above three labels as in SNLI and MultiNLI. The definitions of the inference relations are also based on SNLI.

[1] `https://crowdsourcing.yahoo.co.jp/`

[2] YJ Captions was constructed by crowdsourcing a task of writing five Japanese captions for each image in MS COCO (Lin et al., 2015).

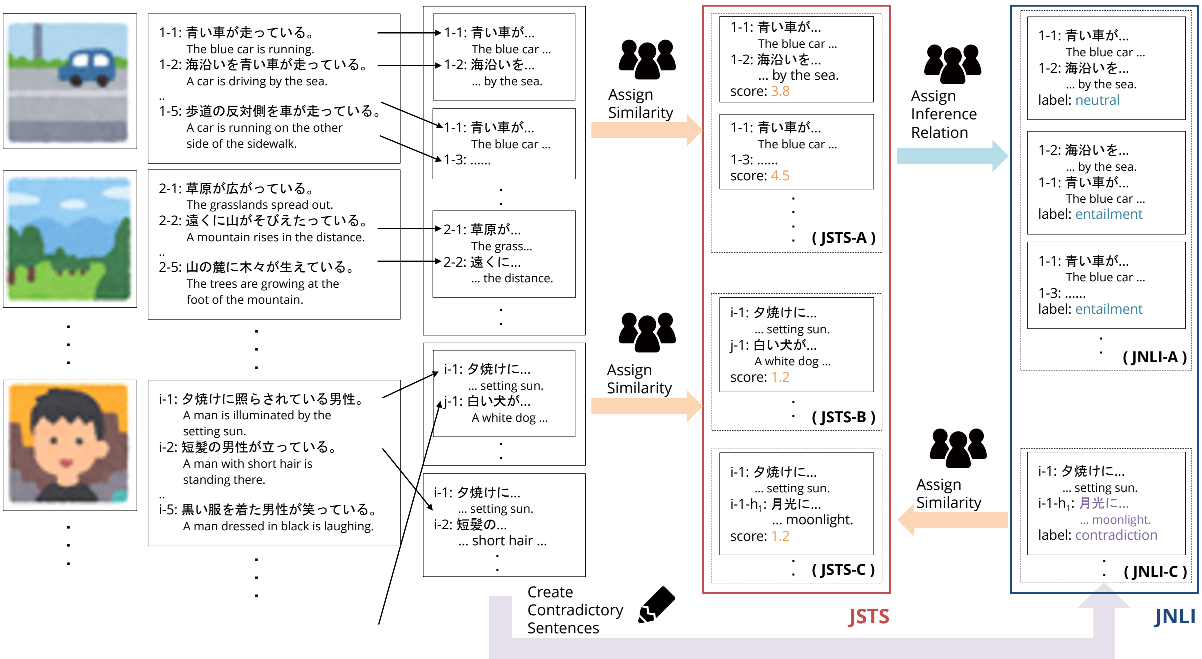


Figure 1: Our construction flow of JSTS and JNLI. Images here are taken from *Irasutoya* (https://www.irasutoya.com/) and ONWA Illust (https://onwa-illust.com/).

Sentence 1 / Premise	Sentence 2 / Hypothesis	Similarity	Relation	Origin
街中の道路を大きなバスが走っています。 A big bus is running on the road in the city.	道路を大きなバスが走っています。 There is a big bus running on the road.	4.4	entailment	A
テーブルに料理がならべられています。 The food is laid out on the table.	テーブルに食べかけの料理があります。 There are some dishes on the table that are about to be eaten.	3.0	neutral	A
野球選手がバットをスイングしています。 A baseball player swings a bat.	野球選手がキャッチボールをしています。 A baseball player plays catch.	2.0	contradiction	C
フリスビーをくわえた犬がいます。 There is a dog with a Frisbee in its mouth.	建物の前にバスが一台停車しています。 There is a bus parked in front of the building.	0.0	—	B

Table 3: Examples of JSTS and JNLI. For origin, A, B, and C indicate that the sentence pairs are contained in (JSTS-A, JNLI-A), (JSTS-B), and (JSTS-C, JNLI-C), respectively.

3.2.2. Method of Construction

Our construction flow for JSTS and JNLI is shown in Figure 1. Basically, two captions for the same image of YJ Captions are used as sentence pairs. For these sentence pairs, similarities and NLI relations of entailment and neutral are obtained by crowdsourcing. However, it is difficult to collect sentence pairs with low similarity and contradiction relations from captions for the same image. To solve this problem, we collect sentence pairs with low similarity from captions for different images. We collect contradiction relations by asking workers to write contradictory sentences for a given caption.

The detailed construction procedure for JSTS and JNLI is described below.

1. We crowdsource an STS task using two captions for the same image from YJ Captions. We ask five workers to answer the similarity between two captions and take the mean value as the gold similarity. We delete sentence pairs with a large variance in the answers because such pairs have poor answer quality. We performed this task on 16,000 sentence pairs and deleted sentence pairs with a similarity variance of 1.0 or higher, resulting in the collection of 10,236 sentence pairs with gold similarity. We refer to this collected data as JSTS-A.

2. To collect sentence pairs with low similarity, we crowdsource the same STS task as Step 1 using sentence pairs of captions for different images. We conducted this task on 4,000 sentence pairs and collected 2,970 sentence pairs with gold similarity. We refer to this collected data as JSTS-B.

3. For JSTS-A, we crowdsource an NLI task. Since inference relations are directional, we obtain in-

ference relations in both directions for sentence pairs. As mentioned earlier, it is difficult to collect instances of contradiction from JSTS-A, which was collected from the captions of the same images, and thus we collect instances of entailment and neutral in this step. We collect inference relation answers from 10 workers. If six or more people give the same answer, we adopt it as the gold label if it is entailment or neutral. To obtain inference relations in both directions for JSTS-A, we performed this task on 20,472 sentence pairs, twice as many as JSTS-A. As a result, we collected inference relations for 17,501 sentence pairs. We refer to this collected data as JNLI-A. We do not use JSTS-B for the NLI task because it is difficult to define and determine the inference relations between captions of different images.[3]

4. To collect NLI instances of contradiction, we crowdsource a task of writing four contradictory sentences for each caption in YJ Captions. From the written sentences, we remove sentence pairs with an edit distance of 0.75 or higher to remove low-quality sentences, such as short sentences and sentences with low relevance to the original sentence.

   Furthermore, we perform a one-way NLI task with 10 workers to verify whether the created sentence pairs are contradictory. Only the sentence pairs answered as contradiction by at least six workers are adopted. Finally, since the contradiction relation has no direction, we automatically assign contradiction in the opposite direction of the adopted sentence pairs. Using 1,800 captions, we acquired 7,200 sentence pairs, from which we collected 3,779 sentence pairs to which we assigned the one-way contradiction relation. By automatically assigning the contradiction relation in the opposite direction, we doubled the number of instances to 7,558. We refer to this collected data as JNLI-C.

5. For the 3,779 sentence pairs collected in Step 4, we crowdsource an STS task, assigning similarity and filtering in the same way as in Steps 1 and 2. In this way, we collected 2,303 sentence pairs with gold similarity from 3,779 pairs. We refer to this collected data as JSTS-C.

We constructed JSTS from JSTS-A, B, and C and JNLI from JNLI-A and C. Finally, we filtered out 12 sentence pairs from JSTS and 44 pairs from JNLI based on automatic matching and manual checking. Table 3 shows examples of the JSTS and JNLI datasets. The statistics of JSTS and JNLI are listed in Tables 4 and 5, respectively.

Similarity range	Train	Dev	Test	Total
0 - 1	2,837	353	405	3,595
1 - 2	1,752	184	160	2,096
2 - 3	2,784	308	355	3,447
3 - 4	3,719	466	488	4,673
4 - 5	1,359	146	181	1,686
Overall	12,451	1,457	1,589	15,497

Table 4: Statistics of JSTS.

Label	Train	Dev	Test	Total
entailment	2,876	353	367	3,596
neutral	11,193	1,347	1,365	13,905
contradiction	6,004	734	776	7,514
Overall	20,073	2,434	2,508	25,015

Table 5: Statistics of JNLI.

Mean of variance	Standard deviation of variance
0.420	0.286

Table 6: Mean and standard deviation of variance of similarity values in JSTS.

To examine the quality of JSTS, we calculated the variance of the similarities of each sentence pair answered by 10 crowdworkers and took the mean and standard deviation for all the pairs. The resulting values were sufficiently small as listed in Table 6. These results guarantee the quality of our annotation.

To assess the inter-annotator agreement of JNLI, we calculated Fleiss' Kappa values for 10 crowdworkers' answers of all the sentence pairs. Its value was 0.399, demonstrating fair to moderate agreement. Although this result showed that each answer was not very reliable, aggregated labels obtained by majority voting could be reliable as shown in the human scores (reported in Section 4.2).

3.2.3. Evaluation Metric

The evaluation metric for JSTS is the Pearson and Spearman correlation coefficients, following STS-B, and that for JNLI is accuracy, following SNLI and MultiNLI.

3.3. JSQuAD

As QA datasets, we build a Japanese version of SQuAD (Rajpurkar et al., 2016), one of the datasets of reading comprehension, and a Japanese version of CommonsenseQA, which is explained in the next section.

Reading comprehension is the task of reading a document and answering questions about it. Many reading comprehension evaluation sets have been built in English, followed by those in other languages or multilingual ones.

In Japanese, reading comprehension datasets for quizzes (Suzuki et al., 2018) and those in the driving

[3] It is natural to define these relations as neutral. In SNLI, however, they are defined as contradiction, whereas such instances are not included in the dataset.

[タイトル] 東海道新幹線
1987年（昭和62年）4月1日の国鉄分割民営化により、JR東海が運営を継承した。西日本旅客鉄道（JR西日本）が継承した山陽新幹線とは相互乗り入れが行われており、東海道新幹線区間のみで運転される列車にもJR西日本所有の車両が使用されることがある。2020年（令和2年）3月現在、東京駅 - 新大阪駅間の所要時間は最速2時間21分、最高速度285km/hで運行されている。
【Title】Tokaido Shinkansen
With the privatization of Japan National Railways on April 1, 1987, JR Tokai took over the operation of the line. The Sanyo Shinkansen, which was taken over by West Japan Railway Company (JR West), is interconnected with the JR Tokai Shinkansen, and trains owned by JR West are sometimes used for trains operating only on the Tokaido Shinkansen section. As of March 2020, the fastest train between Tokyo Station and Shin-Osaka Station takes 2 hours and 21 minutes at a maximum speed of 285 km/h.

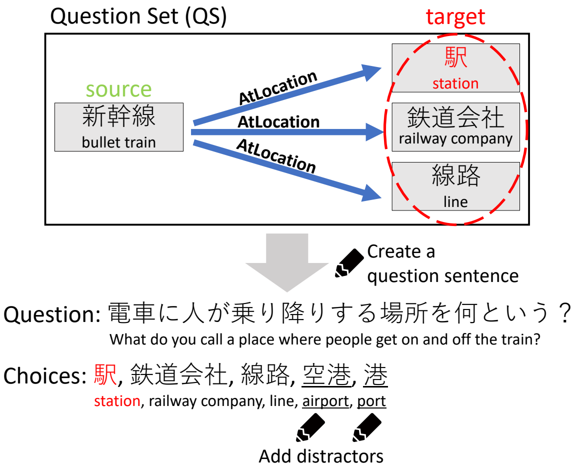
Q: 2020年、東京～新大阪間の最速の所要時間は
In 2020, what is the fastest travel time between Tokyo and Shin-Osaka?
A: 2時間21分
2 hours and 21 minutes
Q: 東海道新幹線と相互乗り入れがされている路線はどこか？
Which lines are interconnected with the Tokaido Shinkansen?
A: 山陽新幹線
Sanyo Shinkansen

Figure 2: Example of JSQuAD.

Question: 会社の最高責任者を何というか？
What do you call the chief executive officer of a company?
Choices: 教師, 部長, 社長, 部下, バイト
teacher, manager, president, subordinate, part-time worker

Question: スープを飲む時に使う道具は何？
What utensil do you use to eat soup?
Choices: スプーン, メニュー, 皿, フォーク, はし
spoon, menu, plate, fork, chopsticks

Figure 3: Examples of JCommonsenseQA. The red choices represent gold ones.



Question: 電車に人が乗り降りする場所を何という？
What do you call a place where people get on and off the train?
Choices: 駅, 鉄道会社, 線路, 空港, 港
station, railway company, line, airport, port

Figure 4: Construction flow of JCommonsenseQA.

domain (Takahashi et al., 2019) have been built, but none are in the general domain. We use Wikipedia to build a dataset for the general domain. The construction process is basically based on SQuAD 1.1 (Rajpurkar et al., 2016).

First, to extract high-quality articles from Wikipedia, we use Nayuki[4], which estimates the quality of articles on the basis of hyperlinks in Wikipedia. We randomly chose 822 articles from the top-ranked 10,000 articles. For example, the articles include “熊本県 (Kumamoto Prefecture)” and “フランス料理 (French cuisine)”.

Next, we divide an article into paragraphs, present each paragraph to crowdworkers, and ask them to write questions and answers that can be answered if one understands the paragraph. Figure 2 shows an example of JSQuAD. We ask workers to write two additional answers for the dev and test sets to make the system evaluation robust.

3.3.1. Evaluation Metric

The evaluation metrics are Exact Match (EM) and F1, as in SQuAD. In English, F1 is calculated on a word level. However, if it is calculated on a word level in Japanese, the value differs depending on the word segmenter used. Therefore, we calculate it on a character level.

3.4. JCommonsenseQA

3.4.1. Overview

JCommonsenseQA is a Japanese version of CommonsenseQA (Talmor et al., 2019), which consists of five-choice QA to evaluate commonsense reasoning ability. Figure 3 shows examples of JCommonsenseQA. In the same way as CommonsenseQA, JCommonsenseQA is built using crowdsourcing with seeds extracted from the knowledge base ConceptNet (Speer et al., 2017). ConceptNet is a multilingual knowledge base that consists of triplets of two concepts and their relation. The triplets are directional and represented as (source concept, relation, target concept), for example (bullet train, AtLocation, station).

3.4.2. Method of Construction

The construction flow for JCommonsenseQA is shown in Figure 4. First, we collect question sets (QSs) from ConceptNet, each of which consists of a source concept and three target concepts that have the same relation to the source concept. Next, for each QS, we crowd-

[4] https://www.nayuki.io/

source a task of writing a question with only one target concept as the answer and a task of adding two distractors. We describe the detailed construction procedure for JCommonsenseQA below, showing how it differs from CommonsenseQA.

1. We collect Japanese QSs from ConceptNet. CommonsenseQA uses only forward relations (source concept, relation, target concept) excluding general ones such as "RelatedTo" and "IsA". JCommonsenseQA similarly uses a set of 22 relations[5], excluding general ones, but the direction of the relations is bidirectional to make the questions more diverse. In other words, we also use relations in the opposite direction (source concept, relation^{-1}, target concept).[6] With this setup, we extracted 43,566 QSs with Japanese source/target concepts and randomly selected 7,500 from them.

2. Some low-quality questions in CommonsenseQA contain distractors that can be considered to be an answer. To improve the quality of distractors, we add the following two processes that are not adopted in CommonsenseQA. First, if three target concepts of a QS include a spelling variation or a synonym of one another, this QS is removed. To identify spelling variations, we use the word ID of the morphological dictionary JumanDic[7]. Second, we crowdsource a task of judging whether target concepts contain a synonym. As a result, we adopted 5,920 QSs from 7,500.

3. For each QS, we crowdsource a task of writing a question sentence in which only one from the three target concepts is an answer. In the example shown in Figure 4, "駅 (station)" is an answer, and the others are distractors. To remove low-quality question sentences, we remove the following question sentences.

   - Question sentences that contain a choice word (this is because such a question is easily solved).
   - Question sentences that contain the expression "XX characters".[8] (XX is a number)
   - Improperly formatted question sentences that do not end with "?".

   As a result, 5,920 × 3 = 17,760 question sentences were created, from which we adopted 15,310 by removing inappropriate question sentences.

4. In CommonsenseQA, when adding distractors, one is selected from ConceptNet, and the other is created by crowdsourcing. In JCommonsenseQA, to have a wider variety of distractors, two distractors are created by crowdsourcing instead of selecting from ConceptNet.

   To improve the quality of the questions[9], we remove questions whose added distractors fall into one of the following categories:

   (a) Distractors are included in a question sentence.

   (b) Distractors overlap with one of existing choices.

   As a result, distractors were added to the 15,310 questions, of which we adopted 13,906.

5. We asked three crowdworkers to answer each question and adopt only those answered correctly by at least two workers. As a result, we adopted 11,263 out of the 13,906 questions.

Finally, we filtered out 14 questions based on automatic pattern matching and manual checking.

3.4.3. Evaluation Metric

The evaluation metric for JCommonsenseQA is accuracy following CommonsenseQA.

4. Evaluation using JGLUE

By using the constructed benchmark, we evaluated several publicly available pretrained models.

4.1. Experimental Settings

The pretrained models used in the experiments are shown in Table 7. These models were fine-tuned in accordance with each task/dataset as follows[10]:

- Text classification and sentence pair classification tasks: classification/regression problems with vector representations of the `[CLS]` tokens.

- JSQuAD: the classification problem of whether each token in a paragraph is a start/end position of an answer span.[11]

[5] The relations are Antonym, AtLocation, CapableOf, Causes, CausesDesire, DefinedAs, DerivedFrom, Desires, DistinctFrom, EtymologicallyDerivedFrom, HasA, HasFirstSubevent, HasLastSubevent, HasPrerequisite, HasProperty, InstanceOf, MadeOf, MotivatedByGoal, NotDesires, PartOf, SymbolOf, and UsedFor.

[6] For example, from triplets such as (station, AtLocation^{-1}, bullet train), we obtain the target concepts "bullet train", "timetable", and "ticket gate" for the source concept "station".

[7] https://github.com/ku-nlp/JumanDIC

[8] This is set up to exclude questions like "What is a word that means *overpriced* in two Chinese characters?".

[9] A question here refers to a set of a question sentence and choices.

[10] Fine-tuning was performed using the transformers library provided by Hugging Face. https://github.com/huggingface/transformers

[11] XLM-RoBERTa$_{\text{BASE}}$ and XLM-RoBERTa$_{\text{LARGE}}$ use the unigram language model as a tokenizer and they are excluded from the targets because the token delimitation and the start/end of the answer span often do not match, resulting in poor performance.

Model Name	Basic Unit	Pretraining Texts
Tohoku BERT$_{\text{BASE}}$ (cl-tohoku/bert-base-japanese-v2)	subword (MeCab + BPE (Sennrich et al., 2016))	Japanese Wikipedia
Tohoku BERT$_{\text{BASE}}$ (char) (cl-tohoku/bert-base-japanese-char-v2)	character	Japanese Wikipedia
NICT BERT$_{\text{BASE}}$	subword (MeCab + BPE)	Japanese Wikipedia
Waseda RoBERTa$_{\text{BASE}}$ (nlp-waseda/roberta-base-japanese)	subword (Juman++ + Unigram LM)	Japanese Wikipedia + CC
XLM-RoBERTa$_{\text{BASE}}$ (xlm-roberta-base)	subword (Unigram LM)	multilingual CC

Table 7: Pretrained models used in our experiments. Names in the parentheses represent the model names in the Hugging Face Hub. Large-sized models are also used corresponding to Tohoku BERT$_{\text{BASE}}$ and XLM-RoBERTa$_{\text{BASE}}$. MeCab (Kudo et al., 2004) and Juman++ (Morita et al., 2015) are Japanese word segmenters. "CC" in pretraining texts represents Common Crawl.

Name	Value
learning rate	{5e-5, 3e-5, 2e-5}
epoch	{3, 4}
warmup ratio	0.1
max seq length	512 (MARC-ja), 128 (JSTS, JNLI), 384 (JSQuAD), 64 (JCommonsenseQA)

Table 8: Hyperparameters used in our experiments. (numbers in curly brackets represent the range of possible values).

Question: 顔を洗う場所は？
Where do you wash your face?
Choices: 洗面所, 店, 台所, 化粧台, ストック
washroom, store, kitchen, vanity, stock

Figure 5: Example of JCommonsenseQA where the output of XLM-RoBERTa$_{\text{LARGE}}$ (underlined choice) was correct while the output of Tohoku BERT$_{\text{BASE}}$ (wavy underlined choice) was incorrect.

- JCommonsenseQA: a multiple choice problem where each choice is represented by the vector representation of the `[CLS]` token of the concatenation of a question and a possible choice.

The best hyperparameters were searched using the dev set, and the performance was evaluated on the test set using the best hyperparameters. The used hyperparameters are listed in Table 8.

4.2. Results

Table 9 shows the performance of each model along with human scores. The human scores were obtained using crowdsourcing in the same way as the dataset construction. The comparison of the models is summarized as follows:

- Overall, XLM-RoBERTa$_{\text{LARGE}}$ performed the best. This may be due to the LARGE model size and the use of Common Crawl as pretraining texts, which is larger than Wikipedia.
- As for the basic unit, the subword-based model (Tohoku BERT$_{\text{BASE}}$) performed consistently better than the character-based model (Tohoku BERT$_{\text{BASE}}$ (char)).
- Since JCommonsenseQA requires commonsense knowledge that is hard to be described in Wikipedia, the models pretrained on Common Crawl performed better. Figure 5 shows an example where the output of XLM-RoBERTa$_{\text{LARGE}}$ (which uses Common Crawl as pretraining texts) was correct while the output of Tohoku BERT$_{\text{BASE}}$ (which does not use Common Crawl) was incorrect.
- In all the datasets other than JCommonsenseQA, the performance of the best model equaled or exceeded the human score.

4.3. Discussion

Is the amount of training data enough? The amount of training data was changed by a factor of 0.75 and 0.5 to see how the performance changed. The model with the best performance for each dataset was used. The learning curve is shown in Figure 6. The performance is almost saturated for all the datasets, indicating that the amount of the constructed data is sufficient.

Annotation artifacts in JNLI In datasets constructed by asking crowdworkers to write sentences, a problem called annotation artifacts arises, especially in NLI (Poliak et al., 2018; Tsuchiya, 2018). If hypothesis sentences are written by workers and include annotation artifacts, a system looking at only hypotheses could achieve moderate performance. We tested this hypothesis-only baseline on JNLI.
First, we extracted a subset of JNLI for this experiment. Specifically, from the sentence pairs whose relation is contradiction, we extracted the sentence pairs in which

	MARC-ja acc		JSTS Pearson/Spearman		JNLI acc		JSQuAD EM/F1		JCommonsenseQA acc	
Model	Dev	Test	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Human	0.989	0.990	0.899/0.861	0.909/0.872	0.925	0.917	0.871/0.944	0.873/0.946	0.986	0.988
Tohoku BERT$_{\text{BASE}}$	0.958	0.957	0.899/0.859	0.901/0.855	0.899	0.876	0.871/0.941	0.879/0.946	0.808	0.782
Tohoku BERT$_{\text{BASE}}$ (char)	0.956	0.957	0.882/0.841	0.889/0.842	0.892	0.861	0.864/0.937	0.864/0.937	0.718	0.728
Tohoku BERT$_{\text{LARGE}}$	0.955	0.961	0.908/0.870	0.907/0.863	0.900	0.878	0.880/0.946	0.881/0.950	0.816	0.822
NICT BERT$_{\text{BASE}}$	0.958	0.960	0.903/0.867	0.909/0.865	0.902	0.881	**0.897/0.947**	**0.904/0.952**	0.823	0.807
Waseda RoBERTa$_{\text{BASE}}$	0.962	0.962	0.901/0.865	0.901/0.857	0.895	0.876	0.864/0.927	0.868/0.926	**0.840**	**0.849**
XLM-RoBERTa$_{\text{BASE}}$	0.961	0.962	0.870/0.825	0.880/0.831	0.893	0.872	-	-	0.687	0.708
XLM-RoBERTa$_{\text{LARGE}}$	**0.964**	**0.965**	**0.915/0.882**	**0.916/0.880**	**0.919**	**0.902**	-	-	**0.840**	0.842

Table 9: Performance on JGLUE dev/test sets.

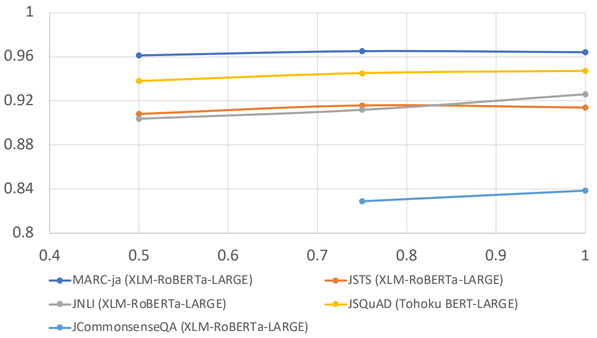


Figure 6: Learning curves varying amounts of training data. Note that the performance of XLM-RoBERTa$_{\text{LARGE}}$ in JCommonsenseQA at a fraction of 0.5 is extremely low, and thus this datapoint is excluded from this graph.

Model	Hypothesis -only	Majority baseline
Tohoku BERT$_{\text{BASE}}$	0.658	0.553
XLM-RoBERTa$_{\text{LARGE}}$	0.553	

Table 10: Accuracy on the JNLI dev sets for the hypothesis-only experiment.

a worker-generated contradictory sentence is a hypothesis. From the sentence pairs whose relation is entailment or neutral, we extracted one-way sentence pairs. We then compared the hypothesis-only baseline with the majority baseline, where all the outputs are neutral. The results are shown in Table 10. Since the hypothesis-only baseline using Tohoku BERT$_{\text{BASE}}$ model outperformed the majority baseline, it is presumed that annotation artifacts are present. We hope that studies on the mitigation of annotation artifacts are conducted based on our constructed dataset.

Lexical overlap in JSQuAD To assess the quality of JSQuAD, we investigated lexical overlap, which was pointed out for SQuAD (Clark et al., 2020). Lexical overlap is the ratio of word overlap between a paragraph and a question. It is reported that the larger the ratio is, the more easily it can be solved by a model. We calculated the ratio of lexical overlap for each paragraph and question pair of JSQuAD by segmenting them into words.[12] As a result, its average value was 0.795, indicating that JSQuAD contains the same problem as SQuAD. Because there has been no benchmark in Japanese so far, it is expected that studies on this problem in Japanese will proceed from our benchmark as a starting point.

5. Conclusion and Future Work

This paper described the construction procedure of JGLUE, a general language understanding benchmark for Japanese. We hope that JGLUE will be used to comprehensively evaluate pretrained models and construct more difficult NLU datasets, such as HotpotQA (Yang et al., 2018), a multi-hop QA dataset, and Adversarial GLUE (Wang et al., 2021).

In the future, we plan to build Japanese datasets for generation tasks such as GLGE (Liu et al., 2021) and for few-shot tasks such as FLEX (Bragg et al., 2021).

6. Acknowledgements

This work was carried out in a joint research project between Yahoo Japan Corporation and Waseda University.

7. Bibliographical References

Abdul-Mageed, M., Elmadany, A. A., and Nagoudi, E. M. B. (2021). ARBERT & MARBERT: deep bidirectional transformers for arabic. *CoRR*, abs/2101.01785.

Bragg, J., Cohan, A., Lo, K., and Beltagy, I. (2021). FLEX: Unifying evaluation for few-shot NLP. In A. Beygelzimer, et al., editors, *Advances in Neural Information Processing Systems*.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*.

Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollar, P., and Zitnick, C. L. (2015). Microsoft coco captions: Data collection and evaluation server.

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

Clark, J. H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T., Nikolaev, V., and Palomaki, J. (2020). TyDi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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, June.

Hayashibe, Y. (2020). Japanese realistic textual entailment corpus. In *LREC2020*, pages 6827–6834, Marseille, France, May. European Language Resources Association.

Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., and Johnson, M. (2020). XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization.

Kakwani, D., Kunchukuttan, A., Golla, S., N.C., G., Bhattacharyya, A., Khapra, M. M., and Kumar, P. (2020). IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online, November.

Keung, P., Lu, Y., Szarvas, G., and Smith, N. A. (2020). The multilingual Amazon reviews corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4563–4568, Online, November.

Kudo, T., Yamamoto, K., and Matsumoto, Y. (2004). Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 230–237, Barcelona, Spain, July.

Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbé, B., Besacier, L., and Schwab, D. (2020). FlauBERT: Unsupervised language model pre-training for French. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France, May.

Liang, Y., Duan, N., Gong, Y., Wu, N., Guo, F., Qi, W., Gong, M., Shou, L., Jiang, D., Cao, G., Fan, X., Zhang, R., Agrawal, R., Cui, E., Wei, S., Bharti, T., Qiao, Y., Chen, J.-H., Wu, W., Liu, S., Yang, F., Campos, D., Majumder, R., and Zhou, M. (2020). XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., and Dollár, P. (2015). Microsoft coco: Common objects in context.

Liu, D., Yan, Y., Gong, Y., Qi, W., Zhang, H., Jiao, J., Chen, W., Fu, J., Shou, L., Gong, M., Wang, P., Chen, J., Jiang, D., Lv, J., Zhang, R., Wu, W., Zhou, M., and Duan, N. (2021). GLGE: A new general language generation evaluation benchmark. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 408–420, Online, August.

Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., and Zamparelli, R. (2014). A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Miyazaki, T. and Shimizu, N. (2016). Cross-lingual image caption generation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1780–1790.

Morita, H., Kawahara, D., and Kurohashi, S. (2015). Morphological analysis for unsegmented languages using recurrent neural network language model. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2292–2297, Lisbon, Portugal, September.

Park, S., Moon, J., Kim, S., Cho, W. I., Han, J. Y., Park, J., Song, C., Kim, J., Song, Y., Oh, T., Lee, J., Oh, J., Lyu, S., Jeong, Y., Lee, I., Seo, S., Lee, D., Kim, H., Lee, M., Jang, S., Do, S., Kim, S., Lim, K., Lee, J., Park, K., Shin, J., Kim, S., Park, L., Oh, A., Ha, J.-W., and Cho, K. (2021). KLUE: Korean language understanding evaluation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., and Van Durme, B. (2018). Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana, June. Association for Computational Linguistics.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (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, November.

Rodriguez-Penagos, C., Armentano-Oller, C., Villegas, M., Melero, M., Gonzalez, A., de Gibert Bonet, O., and Pio, C. C. (2021). The catalan language CLUB. *CoRR*, abs/2112.01894.

Ruder, S., Constant, N., Botha, J., Siddhant, A., Firat, O., Fu, J., Liu, P., Hu, J., Garrette, D., Neubig, G., and Johnson, M. (2021). XTREME-R: Towards more challenging and nuanced multilingual evaluation.

Seelawi, H., Tuffaha, I., Gzawi, M., Farhan, W., Talafha, B., Badawi, R., Sober, Z., Al-Dweik, O., Freihat, A. A., and Al-Natsheh, H. (2021). ALUE: Arabic language understanding evaluation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 173–184, Kyiv, Ukraine (Virtual), April.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August.

Speer, R., Chin, J., and Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb.

Suzuki, M., Matsuda, K., Okazaki, N., and Inui, K. (2018). Construction of a question answering dataset with answerability by reading. In *NLP2018*. in Japanese.

Takahashi, N., Shibata, T., Kawahara, D., and Kurohashi, S. (2019). Machine comprehension improves domain-specific Japanese predicate-argument structure analysis. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 98–104, Hong Kong, China, November.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. (2019). CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota, June.

Tsuchiya, M. (2018). Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium, November.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2019). SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, et al., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Wang, B., Xu, C., Wang, S., Gan, Z., Cheng, Y., Gao, J., Awadallah, A. H., and Li, B. (2021). Adversarial glue: A multi-task benchmark for robustness evaluation of language models.

Warstadt, A., Singh, A., and Bowman, S. R. (2019). Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June.

Xu, L., Hu, H., Zhang, X., Li, L., Cao, C., Li, Y., Xu, Y., Sun, K., Yu, D., Yu, C., Tian, Y., Dong, Q., Liu, W., Shi, B., Cui, Y., Li, J., Zeng, J., Wang, R., Xie, W., Li, Y., Patterson, Y., Tian, Z., Zhang, Y., Zhou, H., Liu, S., Zhao, Z., Zhao, Q., Yue, C., Zhang, X., Yang, Z., Richardson, K., and Lan, Z. (2020). CLUE: A Chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4762–4772, Barcelona, Spain (Online), December. International Committee on Computational Linguistics.

Yanaka, H. and Mineshima, K. (2021). JSICK: Japanese sentences involving compositional knowledge dataset. In *JSAI2021*. in Japanese.

Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W., Salakhutdinov, R., and Manning, C. D. (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, October-November. Association for Computational Linguistics.

Yoshikoshi, T., Kawahara, D., and Kurohashi, S. (2020). Multilingualization of a natural language inference dataset using machine translation. In *The Special Interest Group Technical Reports of IPSJ*. in Japanese.