Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 6827–6834
Marseille, 11–16 May 2020

Japanese Realistic Textual Entailment Corpus

Yuta Hayashibe
Megagon Labs, Tokyo, Japan, Recruit Co., Ltd.
7-3-5 Ginza Chuo-ku, Tokyo, 104-8227, Japan
hayashibe@megagon.ai

Abstract

We perform the textual entailment (TE) corpus construction for the Japanese Language with the following three characteristics: First, the corpus consists of realistic sentences; that is, all sentences are spontaneous or almost equivalent. It does not need manual writing which causes hidden biases. Second, the corpus contains adversarial examples. We collect challenging examples that can not be solved by a recent pre-trained language model. Third, the corpus contains explanations for a part of non-entailment labels. We perform the reasoning annotation where annotators are asked to check which tokens in hypotheses are the reason why the relations are labeled. It makes easy to validate the annotation and analyze system errors. The resulting corpus consists of 48,000 realistic Japanese examples. It is the largest among publicly available Japanese TE corpora. Additionally, it is the first Japanese TE corpus that includes reasons for the annotation as we know. We are planning to distribute this corpus to the NLP community at the time of publication.

Keywords: Textual Entailment, Language Model, BERT, Reasoning Annotation

1. Introduction

In the era where massive texts are produced every day, machines are indispensable to assist with daily life. However, to be truly helpful, machines must understand the meaning of texts (natural language understanding: NLU). An essential task for NLU is Recognizing Textual Entailment (RTE), which is also known as Natural Language Inference (NLI). RTE predicts the relation between two statements, where one statement is called the "hypothesis" and the other is the "premise." The goal is to predict whether the premise entails the hypothesis or not.[1] Improving its performance is useful for natural language applications like Tatar et al. (2008) for summarization, and Harabagiu and Hickl (2006) for question answering. Many corpora have been created to train TE classifiers and evaluate RTE.

In this paper, we perform the textual entailment (TE) corpus construction with the following three characteristics: First, the corpus consists of realistic[2] sentences (Section 4). There are several methods to make hypothesis-premise pairs.[3] In most previous works, human annotators are asked to compose new sentences for the given sentences to make examples in some previous works. However, such artificial sentences lead hidden biases, because annotators unconsciously use particular words. Tsuchiya (2018) showed many TE labels in a corpus can be predicted without seeing premises. Hence we propose simple methods using semantic similarity and surface string similarity to collect natural occurring sentences to make examples.

Second, the corpus contains adversarial examples (Section 5). The performances of recent pre-trained language models such as BERT (Devlin et al., 2019) far surpass those of previous models. They achieve almost the same performance as humans. However, there are still examples that cannot be solved by them. To make TE classifiers more robust, such examples are needed for training. In this paper, we create adversarial examples by two methods: a collection of marginal examples by a classifier and a generation of almost realistic examples with a language model.

Third, the corpus contains explanations for a part of TE labels (Section 6). While there are only labels for examples in TE corpora in general, we perform the reasoning annotation where annotators are asked to check which tokens in hypotheses are the reason why the relations are non-entailment. It makes easy to validate the annotation and analyze system errors. Additionally, it enhances corpus quality by making annotators more serious. This is the first Japanese TE corpus that includes reasons for the annotation as we know.

The resulting corpus consists of 48,000 realistic Japanese examples. At the time of publication, we are planning to distribute this corpus to the NLP community. It will be the largest among publicly available Japanese TE corpora.

2. Related Work

2.1. Human Involvement in TE Corpora

The corpora in the PASCAL RTE challenge (Dagan et al., 2006; Bar-Haim et al., 2006; Giampiccolo et al., 2007; Giampiccolo et al., 2008; Bentivogli et al., 2009) are the most common English TE corpora. In this challenge, the corpora for training and testing are constructed using seven methods. Most require human composition. For example, in the method "Machine Translation (MT)", annotators manually translate sentences as well as use a machine translation system. They modify the sentence pairs to create pairs of hypotheses and premises.

The Sentences Involving Compositional Knowledge (SICK) corpus (Marelli et al., 2014) is created to extract captions for the same picture or video and apply a three-step process to generate sentence pairs. The process includes sentence normalization, sentence expansion and creating pairs with normalized sentences and expanded sentences. It contains 9,927 English examples. For each pair, the relationship between two sentences is scored in

[1] We use binary TE labels. However, some corpora use three types such as entailment, neutral, and contradiction.

[2] In this paper we define "realistic" as spontaneous or almost equivalent; that is, realistic sentences are natural occurring sentences or natural sentences generated by a language model.

[3] We call the pair example.

five levels. Annotators do not need to compose sentences to construct the corpus. However, both normalization and expansion require handcrafted rules.

The Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015) is created by using the Flickr corpus as the premises and asking annotators to compose three sentences: one to entail it, one to contradict it, and one that is unrelated to it. It includes 57,000 English examples. In the creation of the Multi-Genre Natural Language Inference (MNLI) corpus (Williams et al., 2018), the same method is used for multiple genre texts. MNLI includes about 433,000 English examples.

The XNLI corpus (Conneau et al., 2018) is an evaluation set in 15 languages. First, they prepared 7,500 English examples in the same way as MNLI corpus construction. Then translators translated them. This corpus was intended for validation and evaluation, thus, it cannot be used for training.

2.2. Problems with Human Involving TE Corpora

Tsuchiya (2018) revealed SNLI corpus has hidden biases. He demonstrated the biases by showing that many labels in the corpus can be predicted without seeing premise sentences. This is because annotators unconsciously use particular words to create hypotheses. For instance[4], "nobody" is often used in contradiction examples. Annotators often use it to negate given premises. As another example, "championship" is often used to create neutral examples. Annotators often use it to create unrelated entities against sport game entities used in a premise.

He also found that such bias may cause a neural network model proposed for RTE to work as an entirely different model than its constructor expects.

2.3. Japanese TE Corpora

There are not many Japanese TE corpora. The RITE corpus[5] (Shima et al., 2011) and RITE 2 corpus[6] (Watanabe et al., 2013) are used in shared task workshops. The RITE corpus contains 3,503 Japanese examples. The RITE 2 corpus contains 4,746 Japanese examples. They are generated by a template-based sentence generator with natural occurring sentences extracted from a newswire QA corpus, or by manual modification of the extracted sentences from a newswire corpus, entrance exams, and Wikipedia. They are partially available to the public.

Textual Entailment Evaluation data[7] is a publicly available Japanese TE corpus, which contains about 2,700 examples. All examples are not created with natural occurring sentences and manually created for this corpus construction. The number of differences between the hypothesis and the premise in almost all examples is one. Therefore the authors insist it is easier to solve than RITE and RITE2.

A corpus containing 83,800 Japanese examples collected with handcrafted patterns and auto-expanded patterns with a web corpus is constructed by Kloetzer et al. (2013). Kloetzer et al. (2015) enlarged it by exploiting the transitivity of entailment and a self-training scheme. The enlarged corpus consists of 217.8 million Japanese entailment examples from web pages with 80% precision. Neither the original nor the expanded corpus is publicly available.

3. Preliminary Setup for Corpus Construction

Here we detail the specifications of our text source and its filtering. We utilize BERT (Devlin et al., 2019) for the filtering (described in this section), the calculation of semantic similarity (Section 4), the generation of tokens (Section 5), and fine-tuning for RTE (Section 7).

3.1. Source of Texts

In our proposed method, we directly use sentences in the corpus to create entailment pairs. When a wide range of topics is mentioned in a corpus, creating relevant sentence pairs can be laborious. Then, we use hotel reviews to construct a TE corpus. It contains a collection of statements about a particular topic, hotel reputation. Therefore it is easy to find entailment pairs from the corpus. Additionally, there are various expressions for the same matter written by many people. We consider it enables us to avoid biases made by specific persons. Consequently, hotel reviews are suitable for the first attempt to construct a TE corpus. In particular, we extracted over 20 million sentences about Japanese hotel reviews posted on Jalan,[8] which is a travel information web site.

3.2. Pre-training of BERT

BERT[9] is a model based on the Transformer (Vaswani et al., 2017) to encode tokens in texts. The state-of-the-art results in many tasks such as GLUE (Wang et al., 2019) and SQuAD (Rajpurkar et al., 2016) have recently been achieved by models using BERT. It can obtain language representations from raw large texts in pre-training and be fine-tuned for specific tasks.

While some pre-trained models trained with web texts or Wikipedia are public, our preliminary results indicate that fine-tuning of a pre-trained model trained with the same corpus has a better performance. Therefore, we performed pre-training from scratch.

For tokenization, we used SentencePiece[10] (Kudo, 2018) which is an unsupervised text tokenizer. It does not require an annotated corpus or a dictionary. It automatically learns units of sentences for the predetermined vocabulary size. Herein we set the size to 32,000 and trained SentencePiece with our corpus.

We set the batch size to 512, the number of attention heads to 12, the number of layers to 12, and the number of hidden

[4] These examples are from his presentation slide: https://www.slideshare.net/slideshow/embed_code/96587673

[5] http://research.nii.ac.jp/ntcir/permission/ntcir-9/perm-ja-RITE.html

[6] http://research.nii.ac.jp/ntcir/permission/ntcir-10/perm-ja-RITE.html

[7] http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?NLPresources

[8] https://www.jalan.net

[9] https://github.com/google-research/bert

[10] https://github.com/google/sentencepiece

Hotel reputation	The interior was also cute. Bread was also rich in variety. The bathtub was spacious and convenient!
Not hotel reputation	Since I got a job, I took my mother on a trip! It was a trip with my dog. I want to stay here the next time I visit Kyoto.

Table 1: Examples of sentences in reviews

Name	Purpose	Positive	Negative	Total
SF_{BASE}	train	1,928	1,642	3,570
	test	221	179	400
SF_{+ME}	train	1,516	284	1,800
	test	166	34	200

Table 2: Number of examples for sentence filtering

Test	SF_{BASE}			SF_{+ME}			SF_{BOTH}		
Model	P	R	F1	P	R	F1	P	R	F1
$M_{SF_{BASE}}$	94.1	93.7	**93.9**	92.1	63.3	75.0	93.4	80.6	86.5
$M_{SF_{+ME}}$	93.5	85.1	89.1	92.5	96.4	94.4	93.0	89.9	91.5
$M_{SF_{BOTH}}$	93.2	93.2	93.2	91.6	98.8	**95.1**	92.5	95.6	**94.0**

Table 3: Performance of sentence filtering

layers to 12. These are the same parameters used in BERT-Base officially distributed by the authors. We trained the BERT model for 1,500,000 steps with TPU. The performance of the obtained model is 0.568 for the accuracy of the masked token prediction and 0.950 for the accuracy of the next sentence prediction.

3.3. Sentence Filtering

Some sentences in reviews are not about hotel reputation but are personal matters (Table 1). To exclude not hotel reputation sentences, we created a classifier to determine whether a given sentence is a reputation of a hotel or not.

First, we selected 3,975 sentences that consist of three tokens[11] for annotation. Then, we asked three annotators to determine whether each sentence is a hotel's reputation or not. After that, we set the temporary label with majority voting. The final labels are determined by one annotator while ensuring the overall consistency of the corpus. We call this SF_{BASE} and use randomly selected 400 examples for test and the 3,570 for training. We call the fine-tuned model with the training data $M_{SF_{BASE}}$.

Then, we classify 30,000 sentences that consist of over four tokens with it and selected 2,000 sentences so that the distribution of classification scores was not biased. We asked five workers to annotate them and set the final labels by one annotator. We call this SF_{+ME} and use randomly selected 200 examples for test and the 1,800 for training.

Table 2 shows the number of examples. We fine-tuned our pre-trained BERT model with 5,970 sentences. Table 3 shows the performance. $M_{SF_{BOTH}}$ is trained with all 5,370 training examples in SF_{BASE} and SF_{+ME}. It performed best on Corpus SF_{BOTH}, which consists of both SF_{BASE} and B SF_{+ME}. We use it for sentence filtering.

[11] Punctuations could be one token.

Name	Purpose	Entailment	Non-Entailment	Total
SemShort	train	3,539	961	4,500
	test	394	106	500
SemLong	train	10,588	7,412	18,000
	test	1,142	858	2,000
Surf	train	4,699	4,301	9,000
	test	526	474	1,000
BASE	(total)	20,888	14,112	35,000

Table 4: Number of examples in BASE

4. Construction of Realistic TE Corpus

To avoid bias caused by manual writing, we simply asked annotators to select labels about entailment for sentence pairs in the corpus. That is, all examples are composed of naturally occurring sentences.

Random selection of pairs to make examples for the annotation is not efficient, because almost all of them are negative examples even though we use a domain-specific corpus. Then, we make pairs of similar sentences in two scales: semantic similarity and surface string similarity.

When a sentence describing a complex situation is used in a hypothesis, it is difficult to find the entailing premise sentences. Therefore, we decided to use only short token sentences as hypothesis sentences. We found that sentences whose numbers of tokens in SentencePiece are three can be meaningful hypotheses. For example, the following two sentences consist of three tokens.

- 格安で泊まれ (stay at a cheap price) / ました (did) / 。(I stayed at a cheap price.)
- パン (bread) / 好きにはたまらない (Irresistible for the lovers.) / 。(Irresistible for bread lovers.)

This is owing to SentencePiece learning to treat frequently occurring expressions in the domain as one token. Note that, these sentences are split into more than three tokens by typical generic tokenizers based on generic dictionary. For example, MeCab[12] with the IPA dictionary splits the former into six tokens and the latter into seven tokens as below.

- 格安/で/泊まれ/まし/た/。
- パン/好き/に/は/たまら/ない/。

We collected 35,000 examples as BASE in the following methods and each example was labeled by five crowdworkers. The final labels were determined by one annotator while respecting majority vote and ensuring the overall consistency of the annotation. BASE consists of three subcorpora: SemShort, SemLong, and Surf. Table 4 shows the number of examples.

4.1. Examples Based on Semantic Similarity

First, we used pairs of semantically similar sentences. This is based on the assumption that similar sentences are likely to be in TE relations.

[12] https://taku910.github.io/mecab/

BERT encodes tokens with a pre-training model.[13] We adopted the method proposed by Arora et al. (2017) to obtain sentence embeddings from token embeddings. Their method is simple but powerful. First, weighted averages of token embeddings are computed for each sentence. The weights are the smoothed inverse occurrence probabilities. Then, a matrix is formed with the sentences in the corpus, and to give its first singular vector. Finally, a vector, which removes the projections of the average vector on the first singular vector, is used as the sentence embedding.

We used the inner product of two sentences as sentence similarity. We exploited faiss[14] (Johnson et al., 2017) which is a library for an efficient similarity-based search.

We collected two types of premises. One type is examples with short premises. As hypotheses, we used 2,149 sentences that consist of three tokens and are manually classified as a hotel reputation. Higher ranked sentences are almost the same expressions as the hypotheses. Hence we obtained five semantically similar sentences for each hypothesis, as premises that are ranked 96th to 100th and then created pairs. Then we randomly extracted 5,000 pairs from 10,745 ($= 2149 \times 5$) pairs. We call this SemShort.

The other type is examples with long premises. As premises, we used 20,000 sentences that consist of more than four tokens and are automatically classified as a hotel reputation. For each premise, we obtained five semantically similar sentences that are automatically classified as hotel reputation and consist of three tokens. Then we randomly extracted 20,000 pairs from 100,000 ($= 20000 \times 5$) pairs. We call this SemLong.

4.2. Examples Based on Surface String Similarity

The annotation for semantically similar pairs reveals many positive cases. To collect more negative examples, we used sentences whose characters are similar pairs. We expected that more negative cases would be collected because the important parts of TE often differ in the pairs. For instance, though two sentences have many common characters, one does not entail the other.

- 駅 (station) から (from) も (also) 近く (close) 立地は (location) 最高です (best)。(The location is the best because it is close to the station.)
- 繁華街 (downtown) から (from) も (also) 近く (close) 立地は (location) 最高です (best)。(The location is the best because it is close to downtown.)

We used SimString[15] (Okazaki and Tsujii, 2010) which quickly finds the sentences in a corpus that are similar to a given sentence. As hypotheses, we use the same 2,149 sentences used in SemShort. For each hypothesis, we obtained all surface string similar sentences as premises that are automatically classified as a hotel reputation and consist of three tokens. Then, we randomly extracted 10,000 pairs. We call this Surf.

Hypothesis	駅近く (near the station) 便利 (convenient) 。 (It is convenient because the location is near the station.)
Premise	駅前で (in front of the station) アクセス (access) 良く (good) 便利 (convenient) 。 (It is convenient and has good access because it is located in front of the station.)
Replacement	[MASK] アクセス (access) 良く (good) 便利 (convenient) 。
#1	交通 (traffic) アクセス (access) 良く (good) 便利 (convenient) 。 (It is convenient and has good traffic access.)
#2	駅からも (from the station) アクセス (access) 良く (good) 便利 (convenient) 。 (It is convenient and has good access from the station.)
Insertion	[MASK] 駅前で (in front of the station) アクセス (access) 良く (good) 便利 (convenient) 。
#3	立地は (location is) 駅前で (in front of the station) アクセス (access) 良く (good) 便利 (convenient) 。 (It is convenient and has good access because the location is in front of the station.)
#4	博多 (Hakata) 駅前で (in front of the station) アクセス (access) 良く (good) 便利 (convenient) 。 (It is convenient and has good access because the location is in front of the station.)

Table 5: Examples of generated sentences with MLM

5. Construction of Adversarial TE Corpus

We additionally constructed two sub-corpora ME and MLM with the following methods. By adding them to the training source, we expect the system obtains more robustness for classification.

5.1. Marginal Examples by a Classifier

ME collects examples with a lower confidence than the model $\mathrm{M_{BASE}}$ which is trained with examples in BASE. As hypotheses, we randomly selected 2,000 sentences, which consist of three tokens and are automatically classified as a hotel reputation. For each hypothesis, we randomly extracted 500 sentences as premises. Then we classified TE with the model and obtained 10,000 less confident examples from 1,000,000 ($= 2000 \times 500$) examples. We call these examples ME.

5.2. Generated Examples with the Masked Language Model

The other method collects adversarial examples outside the text corpus. To collect adversarial examples, several methods are proposed. Samanta and Mehta (2017) replaced a word with its synonym using a dictionary for classification

[13] Actually, it is possible to use embeddings of a special token [CLS] as sentence embeddings. However, the performance of the similarity calculation is bad without fine-tuning.

[14] https://github.com/facebookresearch/faiss

[15] http://www.chokkan.org/software/simstring/index.html.en

#1	Hypothesis	駅からも近く (near the station) 利便性が (convenient) 高い (very)。 (It is near the station and very convenient.)
	Premise	立地条件はいいです。(Good location.)
#2	Hypothesis	朝食の (breakfast) メニューが (menu) 豊富 (abundant) 。 (The breakfast menu is abundant!!)
	Premise	朝ごはんは良かったですよ!! (Breakfast was good!)

Table 6: Examples of reasoning annotations. Underlined tokens are the reasons for non-entailment.

tasks. Belinkov and Bisk (2018) replaced characters to generate noise for machine translations. Zhang et al. (2019) scrambled words using back-translations, filters, and human judgments for paraphrase identification. We used the masked language model (MLM) of BERT to create realistic adversarial examples. By using this, it enables not only synonym substitution but also various types of sentence generation. Additionally, it does not need manual editing to generate realistic sentences.

An MLM task is often referred to as a cloze task. The model is trained to predict masked tokens[16] represented by special tokens [MASK] while considering the whole sentence. It can generate new probable sentences by replacing tokens in a sentence with [MASK] or inserting [MASK] between the original tokens. We explored new examples by modifying the original examples. This is the first attempt to generate adversarial examples with MLM.

We used 2,903 examples in BASE to generate adversarial examples labeled as entailment by all annotators. We replaced each token in each example to [MASK] and predict probable 20 tokens. We also insert a token [MASK] to each gap between words in the example and predicted 20 probable tokens. Table 5 shows examples. Examples #1 and #2 are generated sentences by replacing "駅前で" (in front of a station) in the premise into predicted tokens. Examples #3 and #4 are ones by inserting predicted tokens before it.

Then we classified their TE by the model $\mathrm{M}_{+\mathrm{ME}}$ which is trained with examples in BASE and ME, and obtained 3,000 less confident examples from 653,606 generated examples.[17] We call these examples MLM.

6. Reasoning Annotation

The e-SNLI[18] (Camburu et al., 2018) contains natural language explanations for the entailment relations in the SNLI corpus. Annotators are asked to select words that they considered essential for the label from the premise, the hypotheses, or both and compose explanations for the premise, the hypothesis, and the label. They also demonstrated its usefulness by showing several experiments: experiments that output of prediction labels with human-interpretable full-sentence justifications, and one to evaluate transfer capabilities to out-of-domain NLI corpus.

We performed one of their annotations that we considered most important. This is the first attempt of reasoning annotation in Japanese TE corpus as we know. We showed a list of tokens of hypotheses and asked workers to select which tokens directly support non-entailment. We consider it helps validation of annotations and analysis of system errors. In addition, we also expected to find annotation errors. Because crowd workers' rewards are determined by the number of examples labeled[19], some works are not labeled carefully. With our analysis, crowd workers tended to label entailment when the hypothesis and the premise are similar.

For the tokenization, we used the SentencePiece model with a vocabulary size of 8,000 to create the token unit fine-grading. We asked three annotators to label 5,080 examples in ME and 655 examples in MLM, which are labeled as entailment. Table 6 shows some examples. Although this is more costly than binary labeling, it helps with the exclusion of false entailment examples. For instance, all five workers incorrectly labeled the entailment for example #1. In this annotation, all three workers annotated "駅からも近く" (near the station) as the reason for non-entailment due to the explicit statement in the premise.

Name	Purpose	Entailment	Non-Entailment	Total
BASE	train	18,826	12,674	31,500
	test	2,062	1,438	3,500
ME	train	4,643	4,357	9,000
	test	432	568	1,000
MLM	train	278	2,422	2,700
	test	19	281	300

Table 7: Number of examples

Test	BASE			ME			MLM		
Model	P	R	F1	P	R	F1	P	R	F1
$\mathrm{M}_{\mathrm{BASE}}$	91.7	96.7	94.1	55.5	69.4	61.7	8.5	57.9	14.8
$\mathrm{M}_{+\mathrm{ME}}$	93.0	96.6	**94.7**	72.8	92.4	81.4	8.0	57.9	14.0
$\mathrm{M}_{\mathrm{ALL}}$	92.7	96.5	94.6	75.3	91.2	**82.5**	20.0	42.1	**27.1**

Table 8: Performance of RTE

7. Analysis of Our TE Corpus

Table 7 shows the distribution of the final corpus. We used 10% of the corpus for testing. The rest is used for training. As a benchmark using our corpus, we fine-tuned our pre-trained BERT model and compared three models: $\mathrm{M}_{\mathrm{ALL}}$, $\mathrm{M}_{+\mathrm{ME}}$, and $\mathrm{M}_{\mathrm{BASE}}$. $\mathrm{M}_{\mathrm{ALL}}$ is trained with all 43,200 training examples in BASE, ME, and MLM. $\mathrm{M}_{+\mathrm{ME}}$ is trained with 40,500 training examples in BASE or ME. $\mathrm{M}_{\mathrm{BASE}}$ is trained only with 31,500 training examples in BASE. We set the batch size to 32, the maximum total input sequence

[16] Strictly speaking, some masked tokens to predict in the MLM task are replaced with random tokens to enhance the robustness of prediction.

[17] We removed duplicated examples.

[18] https://github.com/OanaMariaCamburu/e-SNLI

[19] To be precise, annotations obtain rewards when they label all of the examples in a set and they correctly label a check example the set.

#	Hypothesis	Premise	Gold	M_{BASE}	M_{+ME}	M_{ALL}
1	朝食美味い。 (Breakfast is delicious.)	値段も安いし、朝ご飯も美味いしかった。 (Low price, delicious breakfast.)	E	E	E	E
2	駐車場も無料です。(Parking is free.)	この立地で駐車場無料は嬉しいです。 (I'm glad the parking lot is free despite the good location.)	E	E	E	E
3	パン好きにはたまらない。(Irresistible for bread lovers.)	読書好きにはたまらないホテルです。 (This hotel is irresistible for reading lovers.)	NE	NE	NE	NE
4	ロビーも最高でした。 (Great lobby.)	無料のラウンジも珈琲最高でした。 (The free lounge and coffee were great.)	NE	NE	NE	NE
5	お風呂は大浴場のみ。 (Only large public baths are available.)	お風呂は大浴場あり。 (There is a large public bath.)	NE	E	NE	NE
6	見た目も味も最高でした。 (It looks and tastes great.)	味も最高でした。 (The taste was great.)	NE	E	NE	NE
7	格安で泊まれました。(I stayed at a cheap price.)	部屋タイプお任せプランだったので安く泊まれました！ (Because I used the plan without soecifying the room type, I could stay cheaply!)	E	NE	E	E
8	シングルルームでした。 (It was a single room.)	清潔感あるシングルルームで、空気清浄機もあり、喫煙室だったが匂いも気にならなかったです (It was a clean single room with an air purifier and it was a smoking room but I did not mind the smell.)	E	NE	E	E
9	夜景最高です。(The night view is great.)	ベイサイドで予約すると日の出がばっちりでとても満足です。 (When you book a bayside room, the sunrise is perfect and you are very satisfied.)	NE	E	E	NE
10	値段も安くまた利用したいです。 (I want to use it again because the price is cheap.)	値段が安くディズニーランドに近いのでまた利用するかもしれません。 (Due to the cheap price and proximity to Disneyland, I may use it again.)	E	NE	NE	NE
11	気軽に利用しています。(I use casually.)	家の様な感覚で使っています。 (I use it like I stay my house.)	E	NE	NE	NE
12	品数も多く満足です! (I am satisfied that the number of items is large!)	バイキングがサイコーです！ (The buffet is great!)	NE	E	E	E

Table 9: Example of predictions by the three models. Note that E means "entailment" and NE means "non-entailment".

length to 25[20], and the training epochs to 3. Table 8 shows the performance of the three models. Table 9 shows some prediction examples.

First, we discuss the value of ME. For BASE, the performance of M_{BASE} is worse than that of M_{+ME} (94.1 and 94.7 in F1). This means that the annotation for marginal examples is effective to improve model performance. Examples #5 to #8 are enhanced using the ME training source.

Next, we discuss the value of MLM. While the performance of M_{+ME} and M_{ALL} for BASE is almost the same, that for ME is not (81.4 and 82.5 in F1). This indicates that the annotation for auto-generated examples and training with them is effective. Example #9 cannot be classified correctly by M_{+ME}.

Finally, we discuss the difficulty of the adversarial examples. In the tests of all models with ME and MLM, the performance is significantly worse than that with BASE. Examples #10 to #12 give false-negatives even with M_{ALL}. It seems to be difficult to classify when multiple statements are present in a hypothesis. For example, there are two statements "I want to use it again" and "cheap" in example #10. Similarly, when a deep understanding of words is needed, it also seems to be difficult to classify. For example, the meaning of "casually" should be recognized in example #11. Hence, a more sophisticated classification model than simple fine-tuning is necessary. Example #12 gives a false-positive even with M_{ALL}. According to the reasoning annotation, the token "品数も多く" (the number of items is large) is the evidence for non-entailment. This shows that the models cannot take the meaning of the token into account and suggests that training examples with such tokens or modifications of the model architecture should be added to utilize word knowledge explicitly. These indicate that there are still examples that are difficult to classify with the existing BERT model. Note that other types of adversarial examples can be obtained by methods different from the BERT-based methods we used.

[20] In our observation, 25 is enough length because there are few long sentences.

8. Conclusion

In this paper, we performed the textual entailment (TE) corpus construction with three characteristics: the collection of examples based on similarity, the collection of adversarial examples, and the annotation of reasons for non-entailment by selecting tokens. All of them do not need manual editing. As a result, we constructed the corpus consisting of 48,000 realistic Japanese examples.

In reasoning annotation, we focused only on non-entailment examples to exclude false entailment. However,

false entailment examples may exist. One future direction is to enrich the reasoning annotation.This will not only enhance the quality of the corpus but also be useful for error analysis of system predictions.
Our corpus is the largest among publicly available Japanese TE corpora. We are planning to enrich this corpus and distribute it to the NLP community at the time of publication.

Acknowledgments

We recognize Dr. Yuki Arase at Osaka University for the many discussions and insightful comments. Furthermore, we thank the anonymous reviewers for their careful reading and valuable comments.

9. Bibliographical References

Arora, S., Liang, Y., et al. (2017). A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *Proceedings of the International Conference on Learning Representations 2017*.

Belinkov, Y. and Bisk, Y. (2018). Synthetic and Natural Noise Both Break Neural Machine Translation. In *International Conference on Learning Representations*.

Devlin, J., Chang, M.-W., et al. (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*, pages 4171–4186.

Harabagiu, S. and Hickl, A. (2006). Methods for Using Textual Entailment in Open-Domain Question Answering. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 905–912.

Johnson, J., Douze, M., et al. (2017). Billion-scale similarity search with GPUs. *arXiv*.

Kudo, T. (2018). Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 66–75.

Okazaki, N. and Tsujii, J. (2010). Simple and Efficient Algorithm for Approximate Dictionary Matching. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 851–859.

Rajpurkar, P., Zhang, J., et al. (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.

Samanta, S. and Mehta, S. (2017). Towards Crafting Text Adversarial Samples. *arXiv*.

Tatar, D., Tamaianu-Morita, E., et al. (2008). Summarization by Logic Segmentation and Text Entailment. *Advances in Natural Language Processing and Application*, pages 15–26.

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*.

Vaswani, A., Shazeer, N., et al. (2017). Attention Is All You Need. In *Proceedings of Neural Information Processing Systems 2017*, pages 5998–6008.

Wang, A., Singh, A., et al. (2019). GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of International Conference on Learning Representations*, pages 1–20.

Zhang, Y., Baldridge, J., et al. (2019). PAWS: Paraphrase Adversaries from Word Scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1298–1308.

10. Language Resource References

Bar-Haim, R., Dagan, I., et al. (2006). The Second PASCAL Recognising Textual Entailment Challenge. In *The Second PASCAL Recognising Textual Entailment Challenge*.

Bentivogli, L., Dagan, I., et al. (2009). The Fifth PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the Second Text Analysis Conference*.

Bowman, S. R., Angeli, G., et al. (2015). A Large Annotated Corpus for Learning Natural Language Inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Camburu, O.-M., Rocktäschel, T., et al. (2018). e-SNLI: Natural Language Inference with Natural Language Explanations. In *Advances in Neural Information Processing Systems 31*, pages 9539–9549.

Conneau, A., Rinott, R., et al. (2018). XNLI: Evaluating Cross-lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485.

Dagan, I., Glickman, O., et al. (2006). The PASCAL Recognising Textual Entailment Challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190.

Giampiccolo, D., Magnini, B., et al. (2007). The Third PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9.

Giampiccolo, D., Dang, H. T., et al. (2008). The Fourth PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the First Text Analysis Conference*.

Kloetzer, J., De Saeger, S., et al. (2013). Two-Stage Method for Large-Scale Acquisition of Contradiction Pattern Pairs using Entailment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 693–703.

Kloetzer, J., Torisawa, K., et al. (2015). Large-Scale Acquisition of Entailment Pattern Pairs by Exploiting Transitivity. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1649–1655.

Marelli, M., Menini, S., et al. (2014). A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Con-*

ference on Language Resources and Evaluation, pages 216–223.

Shima, H., Kanayama, H., et al. (2011). Overview of NTCIR-9 RITE: Recognizing Inference in TExt. In *Proceedings of NTCIR-9*.

Watanabe, Y., Miyao, Y., et al. (2013). Overview of the Recognizing Inference in Text (RITE-2) at NTCIR-10. In *Proceedings of the 10th NTCIR Conference*, pages 385–404.

Williams, A., Nangia, N., et al. (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*, pages 1112–1122.