Japanese Named Entity Recognition from Automatic Speech Recognition Using Pre-trained Models

Seiichiro Kondo[1] **Naoya Ueda**[1] **Teruaki Oka**[1]
Masakazu Sugiyama[2] **Asahi Hentona**[2] **Mamoru Komachi**[1]
[1]Tokyo Metropolitan University [2] AI Shift
{kondo-seiichiro, ueda-naoya}@ed.tmu.ac.jp, teruaki-oka@tmu.ac.jp
{sugiyama_masakazu, hentona_asahi}@cyberagent.co.jp,
komachi@tmu.ac.jp

Abstract

Japanese named entities extracted from automatic speech recognition frequently contain speech recognition errors and unknown named entities due to abbreviations and aliases. One possible solution to this problem of the named entity extraction task is to use a pre-trained model trained on a large quantity of text to acquire various contextual information. In this study, we performed named entity recognition on the logs of a task-oriented dialogue system for road traffic information in Fukui, Japan, using pre-trained BERT-based models and T5. In our experiments using our prepared data, the F1 scores of BERT and T5 are higher than that of string match by 20.2 point and 21.1 points, respectively. The results confirmed that these pre-trained models exhibited significantly higher accuracies on unseen entities than methods based on dictionary matching.

1 Introduction

In interactive voice response services, it is necessary to accurately perform entity extraction and mention detection from the speech recognition results of user's speech to carry out dialogue via voice. However, this input contains a variety of noise, including speech recognition errors, compared to written text. Moreover, users' utterances may include personal representations for a certain entities. In our full research, we plan to address these issues in two steps: named entity recognition (NER) and entity linking (EL). The first step, NER, recognizes a part of the text that is likely to represent a named entity. In the second step, we then attempt to link it to a prepared set of entities even if it contains errors. With this architecture in mind, in this study, we tackle the first step, which is NER from speech recognition results.

This study focused on named entity recognition (NER) in the context of a task-oriented dialogue system that provides information in response to the user's requests pertaining to road traffic. In our system, NER is accomplished by linking the automatic speech recognition (ASR) text with a dictionary created for each task (see Fig. 1). One way to achieve more accurate recognition is to extract the named entities before the linking.

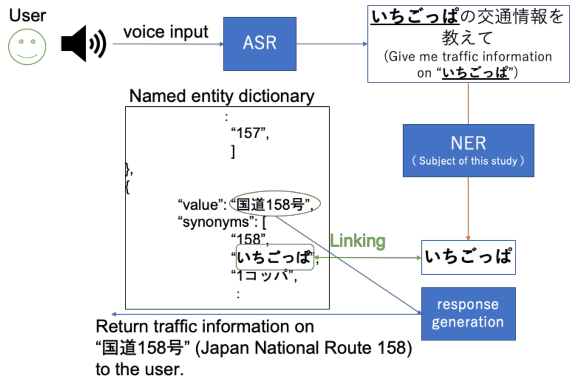

Figure 1: Flowchart of our envisioned spoken dialogue system. This study focuses on the NER component. The goal is to extract named entities even from noisy text caused by speech recognition errors or abbreviations.

Although we enforce the inputs to include location names by system-driven conversation, speech recognition errors and unknown named entities from abbreviations and aliases may occur. For example, "いちごっぱ" (*ichi-go-ppa*, 158) is a colloquial expression for "国道158号" (*kokudou-hyaku-goju-hachi-gou*, Japan National Route 158). In addition, there are many cases in Japanese where the sounds are close, but the surface forms and meanings are entirely different. Therefore, in this setting, NER using conventional methods is difficult, especially for rule-based methods.

To improve text processing functionality, this study focused on NER on ASR texts. [1] We focused on context-based NER in the ASR texts because named entities may be unknown surfaces as

[1]Note that, although Omachi et al. (2021) postulated that an end-to-end (E2E) approach for processing speech recognition results might be preferable, we used existing ASR to enable the flexible exchange of modules and resources, making it necessary to process ASR texts.

a result of speech recognition errors or abbreviations. Based on the assumption that contextual information can be used effectively by pre-trained models trained on a large number of sentences, we used BERT-based large-scale pre-trained models for NER (Devlin et al., 2019; Clark et al., 2020). We also investigated the performance of T5 (Raffel et al., 2020), a pre-trained encoder-decoder model.

2 Related Work

NER from ASR. Wang et al. (2021) performed NER from speech recognition using the matching approach. They used the embeddings of the top N prediction candidates of ASR. In this study, we experimented with only the top predicted ASR candidate, and performed string matching with rule-based NER for simplicity.

NER from speech recognition with neural models for English has been studied previously. Raghuvanshi et al. (2019) extracted personal names from text containing speech recognition errors using additional information not contained in the text and reported that the recall was improved. Yadav et al. (2020) studied the E2E approach and were able to extract named entities robustly and efficiently. We used neural models to perform NER from Japanese ASR texts under the assumption that the ASR architecture cannot be changed. In other words, in this study, experiments will be conducted in a setting where only the text processing section is involved and no other information, such as voice information, is used.

Japanese NER. Rule-based matching methods (Sekine et al., 1998) and machine learning-based methods (Utsuro and Sassano, 2000; Sassano and Utsuro, 2000) have been proposed for Japanese NER. However, these studies focused on manually written texts, whereas ASR texts often contain problems specific to speech recognition, such as speech recognition errors. In this study, we attempted to extract named entities from such ASR texts.

NER in Japanese speech recognition has been performed using support vector machines (SVMs). Sudoh et al. (2006b,a) reported that when training SVMs on ASR text, precision can be improved by incorporating a confidence feature that indicates whether a word is correctly recognized. In contrast, we aimed to extract named entities from text containing speech recognition errors, focusing on recall to lead to subsequent linking tasks and using

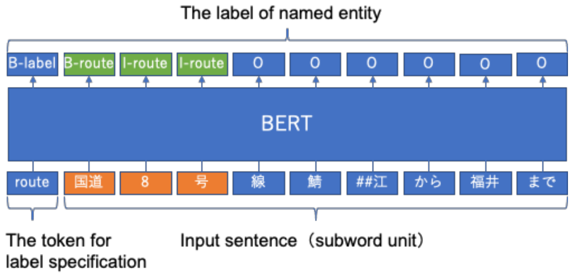

Figure 2: NER using BERT-based models

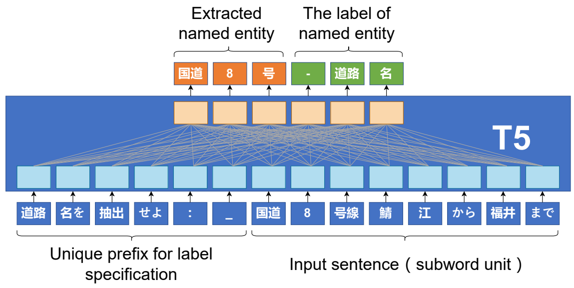

Figure 3: NER using T5

a pre-trained model for this purpose.

3 Our Method

In this study, we used road traffic data for NER evaluation of ASR text containing speech recognition errors and obtained named entities related to roads and addresses. We assumed that the output labels (roads and addresses) could be specified as a precondition.

3.1 NER using BERT based models

Devlin et al. (2019) demonstrated the state-of-the-art performance of a fine-tuned BERT model on the CoNLL-2003 NER task (Tjong Kim Sang and De Meulder, 2003). Following their approach, we considered NER as a sequence labeling task. The text was tokenized, split into subwords, and labeled based on the BIO model, in which "B" was assigned to the beginning, "I" was assigned to the interior and the end of the named entities, and "O" was assigned to any other tokens. The schematic view is presented in Figure 2. Labels for road information were considered as "`{B, I}-route`", and labels for address information as "`{B, I}-address`". To specify a label, we prefixed the statement with a "route" or "address" token and gave "B-label".

	text
match	鯖江から敦賀市へ向かう高速道路 (Highway from Sabae to Tsuruga City)
fallback	えーとサザエさん、サザエ市春江町 (Well, Sazae-san, Sazae City, Harue-cho)

Table 1: Example of match and fallback data (the underlined parts are named entities):サザエ (*sazae*, turban shell) is a recognition error of 鯖江 (*sabae*), which is the name of a city in Fukui.

		train	dev	test
match	utterance	1,757	220	220
	address	1,220	144	147
	route	802	104	110
fallback	utterance	949	118	122
	address	197	30	26
	route	92	8	17

Table 2: The number of data instances used in the experiment (the number of utterances and named entities with each label).

3.2 NER using T5

We performed NER with a Seq2Seq pre-trained model because Constantin et al. (2019) reported that Seq2Seq models achieve excellent sequence labeling of noisy texts. Also,Phan et al. (2021) performed NER using domain-adapted T5 on medical literature. Their experimental results show that NER can be performed with high F1 scores even with a seq2seq model, namely T5. Following their approach, we considered NER as a question-and-answer task. A text with a special label at the beginning was the input sequence and named entities corresponding to this special label were output. The system was set up such that each extracted named entity was added to the end with a label followed by a dash. The schematic view is presented in Figure 3. Labels for road information were considered as "道路名" (road name), and the labels for address information as "住所名" (address name). To specify the label, the special tokens "道路名を抽出せよ:" (extract road names) or "住所名を抽出せよ:" (extract address names) were added at the beginning of the sentence.

4 Experiments

4.1 Data

In this study, we conducted NER using a system-driven dialogue log containing road traffic information in Fukui, Japan [2]. The dialogue logs were obtained from the turns where the user seemed to have uttered the names of roads or addresses based on the conversation before and after. A dictionary of the named entities to be extracted was provided. In this dictionary, aliases, abbreviations, and speech recognition errors were registered (the dictionary is shown in Fig. 1). The target texts for which NER succeeded and failed were *match* and *fallback*, respectively; an example is presented in Table 1.

The match data were labeled by dictionary matching, with incorrect labels manually removed. For the data in fallback, we manually annotated named entities related to the road and the address. This annotation was performed considering speech recognition errors and any named entities existing in Fukui even though not in the dictionary. Notably, because fallback data were annotated based on whether the named entities exist in Fukui, a difference existed in the criteria of labeled words between match and fallback data. We randomly split match and fallback data so that the training, development, and test data were 8:1:1. The number of data instances is shown in Table 2. Note that these data were not arbitrarily sampled but were obtained over a certain period.

Achieving accurate NER with match and small fallback training data is practical, since the former requires only a reasonably sized dictionary but the latter needs human annotation. For data collection, we considered match data as inexpensive to obtain because they could be extracted by dictionary matching, and fallback data as expensive because they could not (Subsection 4.4).

4.2 Setting

We compared four NER systems, viz., a string-matching model based on a dictionary, two pre-trained BERT-based models, and T5. The script of transformers, published by Huggingface [3], was used for fine-tuning all models.

For the BERT-based models, we used the BERT

[2] We will perform NER on other dialogue log data as well in the future.

[3] https://github.com/huggingface/transformers

method	data	P	R	F1	c_P	c_R	c_F1	P	R	F1	c_P	c_R	c_F1
String	M	96.3	**100**	**98.1**	96.3	**100**	98.1	—	—	—	—	—	—
Match	F	50.0	23.3	31.7	50.0	23.3	31.7	—	—	—	—	—	—
		trained using all data						trained by match data					
BERT	M	97.3	97.3	97.3	**99.2**	99.2	**99.2**	97.3	97.3	97.3	98.8	98.8	98.8
	F	67.9	83.7	75.0	67.9	83.7	75.0	**58.8**	46.5	**51.9**	**58.8**	46.5	**51.9**
ELECTRA	M	96.9	98.1	97.5	98.1	99.2	98.6	**97.7**	**98.1**	**97.9**	**99.2**	**99.6**	**99.4**
	F	66.0	72.1	68.9	66.0	72.1	68.9	54.5	41.9	47.4	57.6	44.2	50.0
T5	M	**98.0**	97.7	97.9	**99.2**	98.8	99.0	97.3	97.7	97.5	98.5	98.8	98.6
	F	**74.0**	**86.0**	**79.6**	**74.0**	**86.0**	**79.6**	41.3	**60.5**	49.1	42.3	**62.8**	50.9

Table 3: Experimental results for string matching, BERT, ELECTRA, and T5. M and F denote match and fallback data, respectively. "c_" means results of re-scoring named entities as true positives when they are predicted to be longer than the reference.

published by Tohoku University [4] and ELECTRA published by Megagon Labs [5]. We fine-tuned both models through token classification. For BERT, we set the batch size to 8, the number of epochs to 3, and the maximum sequence length to 258. For ELECTRA, we set the learning rate to 0.00005, the batch size to 8, the number of epochs to 20, and the maximum sequence length to 128. The T5 model was fine-tuned from the model published by Megagon Labs [6]. We set the learning rate of T5 to 0.0005, the batch size to 8, the number of epochs to 20, and the maximum sequence length to 128. Note that the pre-training datasets for T5 and ELECTRA were approximately the same size, while that for BERT was much smaller.

4.3 Evaluation

We evaluated the performance by calculating precision, recall, and F1 scores, considering a perfect match as a true positive. Because of the difference between the labeling criteria of the match and fallback data (Subsection 4.1), we evaluated each test set separately. Evaluation of match data serves as a measure of whether named entities flagged by dictionary matching can be extracted, whereas evaluation of fallback data measures whether it is possible to extract named entities that are not included in the dictionary.

In this study, we assumed that entity linking was performed in the downstream task. If the extracted named entities are shorter than the original entities, linking may become problematic. In contrast, when the extracted named entities are longer than the original entities, the problem in linking is considered minor. Therefore, under the lenient evaluation setting, we considered the cases in which the named entities were covered as true positives, but we still considered the cases in which the partial matches were not covered as false positives. For example, if the named entity "**8号線**" (Route 8) is in the reference, the extraction of "**国道8号線**" (Japan National Route 8) is acceptable, but the extraction of only "**8号**" (Route 8) is not acceptable.

4.4 Results

Experimental results for the match and fallback test sets are presented in Table 3 for when both datasets were used as training data and for when only the match data was used.

String Matching For the match data, the recall was 100 because the data was created using dictionary matching. Precision was not 100 because we manually removed mislabeled data (Subsection 4.1) when creating the match data. Conversely, for the fallback data all the evaluation scores were less than 50.0, and so the unseen named entities were not sufficiently extracted.

BERT For the match test data, the F1 and c_F1 scores of BERT were comparable or superior to that of string matching, which was a desirable result for NER for subsequent tasks. For the fallback test data, the score was 20.2 points higher when training using only the match data and 43.3 points higher

[4] https://huggingface.co/cl-tohoku/bert-base-japanese-v2

[5] https://huggingface.co/megagonlabs/transformers-ud-japanese-electra-base-discriminator

[6] https://huggingface.co/megagonlabs/t5-base-japanese-web

model	text	translation
BERT (address)	吉田郡 **永平寺町**	(Yoshida-gun Eiheiji-cho)
T5 (address)	田尻町から**福井市**までの**福井市**内まで	(From Tajiri-cho to Fukui City to Fukui City)
BERT/T5 (route)	**イチゴったー**	(Ichigotta)

Table 4: Example of NER failure in match data. Bold and underlined texts denote the reference and hypothesis.

model	text	translation
BERT/T5 (address)	**横倉**ってどこや	(Where is Yokokura)
BERT (route)	**青年の道**	(Youth Road)
T5 (address)	低い	(low)
BERT/T5 (route)	**アイワかどう**	(Aiwakado)
BERT (address)	あの高みの方のエルパ行きのバスは取った後	(After taking the bus to Elpa at that height)
T5 (address)	あの高みの方のエルパ行きのバスは取った後	(After taking the bus to Elpa at that height)

Table 5: Example of NER in fallback data. Bold and underlined texts denote the reference and hypothesis.

when training using all data compared with string matching. In particular, the improvement in recall was remarkable, which indicated that BERT could extract unique named entities that could not be extracted by string matching. Adding the fallback data to the match data for training considerably increased the score. This increase is attributed to words not included in the match data (dictionary) being considered during training.

ELECTRA The score of ELECTRA was lower than that of BERT except for match test data when the model was trained using match data. This result shows that the amount of data used for pre-training has a small impact on the results of NER.

T5 The trend observed for T5 is the same as that for BERT. For the match test data, the performance was comparable to BERT. For the fallback test data, the precision was lower than that of BERT when training with only the match data. However, adding the fallback data to the match data for training improved the precision, and the F1 score was +4.6 points compared with BERT, which indicates that the extraction is consistent with the intention.

5 Discussion

Comparison to human performance To evaluate the upper limit of the fallback test data, we calculated the human recognition score by asking another person, not the annotator. Note that in human recognition, labeling is performed while checking whether the named entity exists in Fukui, considering the speech recognition errors. Precision, recall, and F1 were 80.0, 97.6, and 87.9, respectively. These results and Table 3 suggest that there is still room for improvement based on the performance of the pre-trained models.

method	error	false positive		false negative		
		NT	PM	ND	AE	others
human	11	8	2	0	0	1
BERT	21	14	3	2	2	0
T5	19	13	1	1	4	0

Table 6: Error analysis of each NER method in fallback data. We categorized the type of NER errors and the type of named entities that could not be extracted. NT: not tagged as named entity in test data. PM: partial match to the extraction span. ND: named entity not in the dictionary. AE: ASR error. Note that the number of samples in the NT and PM include those incorrectly labeled beginning with "I" by BERT.

Error analysis in fallback data Table 6 shows the classification results of NER error patterns for human and pre-trained models and the number of samples per type of named entities that could not be extracted in the fallback data. The number of extraction errors of BERT is higher than the others but is similar to that of T5.

In BERT outputs, some incorrect labels start with "I" for spans that might be part of named entities. These examples can be attributed to BERT's failed attempts to consider the context. Note that false positive extraction errors can be tolerated because entity linking is assumed to be performed subsequently in this setting.

Focusing on false negative errors, it seems that there is room for improvement in the extraction of named entities that are not included in the dictionary and named entities with speech recognition errors since a human could distinguish them. Since many of the error cases in the results obtained in this study are short in user speech, it may be necessary to use external data.

Examples Table 4 presents an example in which T5 successfully extracts but BERT fails, an example in which BERT successfully extracts but T5 fails, and an example in which both fail for match data. For fallback data, in addition to the same types of examples as Table 4, Table 5 shows an example of successful extractions with BERT and T5.

Because the test data of "match" are based on a dictionary match, the reference does not include "吉田郡" (*Yoshida-gun*) and "田尻町" (*Tajiri-cho*), but it must be noted that these are place names that exist in Fukui, and are therefore examples that should be extracted.

"イチゴったー" (*Ichigotta*) is thought to be a misrecognition of "いちごっぱ" (*ichi-go-ppa*, 158), which is sometimes uttered for "158号線" (*Hyakugojuhachi-gosen*, Route 158). Moreover, "アイワかどう" (*Aiwakado*) is thought to be a misrecognition of "舞若道" (*Maiwakado*), which is sometimes uttered for "舞鶴若狭自動車道" (*Maizuru-wakasa-jidosyado*). It is thought that the NER may be complicated by three main problems: simplified spoken language such as aliases or abbreviations, speech recognition errors, and specific to the Japanese language of notation, *kanji*, *hiragana*, and *katakana*. Currently, named entities such as these examples are handled using dictionaries, but the creation of dictionaries is time-consuming and they are limited in coverage. We plan to devise specific solutions for each of these problems in future work.

BERT and T5 can extract "横倉" (*Yokokura*). This named entity was not included in the training data even after adding fallback to the training data. This suggests that BERT and T5 can extract unknown named entities based on contextual information. Moreover, although "青年の道" (Youth Road) displayed in the fallback is not included in the training data, it is a road name that exists in Fukui. T5 was able to extract it because it predicted the road name from the word "道" (road) at the end. Only T5 could predict the road name from such a context, probably because of its different model structure and NER method. The identification of these factors is a subject for future research. Both BERT and T5 extracted "高みの（方の）" (height) as a named entity representing an address, which reveals that these models contextually tried to extract named entities from expressions that represent directions ("方").

"低い" (*hikui*, low) was extracted by T5. This may be a speech recognition error for "Fukui." Some of the user input in this experiment is shorter than a typical question answering task, and contextual information is not available in such examples. Nevertheless, the fact that T5 was able to extract this word is a remarkable result.

6 Conclusion

We performed Japanese NER on speech recognition using pre-trained BERT-based models and T5. The results of the experiment showed that data generated by dictionary matching was generally well extracted by the pre-trained models. Furthermore, by adding manually annotated data to the training data, we confirmed that it is possible to extract named entities not included in the dictionary. In future, we will consider more context-sensitive methods, including fine-tuning methods, to robustly extract named entities from noisy text containing unknown named entities, such as adding data that masks named entities to the training data.

References

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020*.

Stefan Constantin, Jan Niehues, and Alex Waibel. 2019. Incremental processing of noisy user utterances in the spoken language understanding task. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 265–274.

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.

Motoi Omachi, Yuya Fujita, Shinji Watanabe, and Matthew Wiesner. 2021. End-to-end ASR to jointly predict transcriptions and linguistic annotations. In

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1861–1871.

Long N. Phan, James T. Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. 2021. SciFive: a text-to-text transformer model for biomedical literature. *CoRR*, abs/2106.03598.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Arushi Raghuvanshi, Vijay Ramakrishnan, Varsha Embar, Lucien Carroll, and Karthik Raghunathan. 2019. Entity resolution for noisy ASR transcripts. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 61–66.

Manabu Sassano and Takehito Utsuro. 2000. Named entity chunking techniques in supervised learning for Japanese named entity recognition. In *Proceedings of the 18th Conference on Computatinal Linguistics (COLING)*, page 705–711.

Satoshi Sekine, Ralph Grishman, and Hiroyuki Shinnou. 1998. A decision tree method for finding and classifying names in Japanese texts. In *Proceedings of the 6th Workshop on Very Large Corpora*, pages 148–152.

Katsuhito Sudoh, Hajime Tsukada, and Hideki Isozaki. 2006a. Discriminative named entity recognition of speech data using speech recognition confidence. In *Proceedings of InterSpeech*, pages 337–340.

Katsuhito Sudoh, Hajime Tsukada, and Hideki Isozaki. 2006b. Incorporating speech recognition confidence into discriminative named entity recognition of speech data. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 617–624.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Takehito Utsuro and Manabu Sassano. 2000. Minimally supervised Japanese named entity recognition: Resources and evaluation. In *Proceedings of the 2nd International Conference on Language Resources & Evaluation (LREC)*, pages 1229–1236.

Haoyu Wang, John Chen, Majid Laali, Kevin Durda, Jeff King, William Campbell, and Yang Liu. 2021. Leveraging ASR N-Best in Deep Entity Retrieval. In *Proceedings of the Interspeech 2021*, pages 261–265.

Hemant Yadav, Sreyan Ghosh, Yi Yu, and Rajiv Ratn Shah. 2020. End-to-end named entity recognition from English speech. In *Proceedings of the Interspeech 2020*, pages 4268–4272.