NAISTSOC at the NTCIR-16 Real-MedNLP Task

Tomohiro Nishiyama
Nara Institute of Science and Technology
Japan
nishiyama.tomohiro.ns5@is.naist.jp

Mihiro Nishidani
Nara Institute of Science and Technology
Japan
nishidani.mihironl5@is.naist.jp

Aki Ando
Nara Institute of Science and Technology
Japan
ando.aki.ab9@is.naist.jp

Shuntaro Yada
Nara Institute of Science and Technology
Japan
s-yada@is.naist.jp

Shoko Wakamiya
Nara Institute of Science and Technology
Japan
wakamiya@is.naist.jp

Eiji Aramaki
Nara Institute of Science and Technology
Japan
aramaki@is.naist.jp

ABSTRACT

This paper describes how we tackled the Medical Natural Language Processing for Real-MedNLP task as participants of NTCIR16. We utilized BERT model for solving this task. We found that BERT model we trained is the best results of subtask 1 with joint-F1-score.

KEYWORDS

Medical Natural Language Processing, Named Entity Recognition, Case Reports, Radiography Reports, Adverse Drug Event, Case Identification

TEAM NAME

NAISTSOC

SUBTASKS

Subtask1-CR-JA
Subtask1-RR-JA
Subtask2-CR-JA
Subtask2-RR-JA
Subtask3-CR-JA (ADE)
Subtask3-RR-JA (CI)

1 INTRODUCTION

In recent years, medical records have increasingly been converted from paper to electronic format, increasing the importance of information processing technology in the medical field. However, privacy-free medical text data is still scarce in non-English speaking countries such as Japan and China.

In this Real-MedNLP task, organizers are restructuring the scheme toward the ultimate goal (so-called medical AI task) of promoting and supporting the development of practical tools and systems for the medical industry to assist medical decisions and treatment by physicians and co-medicals.

This task provides two core resources; (1) Case-Report corpus (shortly **MedTxt-CR**) and (2) Radiology-Report corpus (shortly **MedTxt-RR**). The challenges of this task are two folds as follows.

- **Few-resource NER (Subtasks 1 & 2)** : Participants extract important information from the real medical texts. This challenge is classified into two ways: just 100 training (subtask 1) and guideline learning (subtask 2). The task is classified by the amount of training data (small data or no data)
- **Applications (Subtask 3)** : This challenge was designed from the practical viewpoints. For case reports, the organizers designed an information extraction task for adverse drug event (ADE). The ADE task has been challenged through workshops (n2c2 2009 etc.). For radiography reports, organizers designed case identification task, which is to detect the reports originated from the same patient.

Further details of this task can be found in the NTCIR Real-MedNLP website[1] and the NTCIR-16 Real-MedNLP task overview paper[5].

We challenged all subtasks for Japanese corpora (Subtask1-CR-JA, Subtask1-RR-JA, Subtask2-CR-JA, Subtask2-RR-JA, Subtask3-CR-JA (ADE), and Subtask3-RR-JA (CI)) by constructing BERT-based models. The results for some parts of subtask 1 and subtask 3 were superior to those of the other teams.

Our policy for this challenge is to investigate the feasibility of the most standard approaches. Thus, the fundamental design of each system is based on the standard approach, which does not utilize any extra resources.

In terms of pre-trained models, several domain specific pre-trained models are already developed, such as BioBERT[4] and clinical BERT[2] in English, UTH-BERT [3] in Japanese. Although these domain-specific models are expected to yield better results in these tasks, in this paper, we use BERT, a more basic, non-domain-specific pre-training model, to evaluate the model as a baseline.

2 METHODS

2.1 Subtask 1

BERT is a Transformers-based large-scale language model that emerged in 2019[1]. It has been very successful in achieving high accuracy on tasks in a variety of domains by fine-tuning the pre-trained model. We approached this task by employing a simple method that utilizes BERT.

We used the BERT pre-training model 'cl-tohoku/bert-base-japanese-whole-word-masking,'[2] The tokenizer is Mecab[3] . Fine tuning was performed with a batch size of 32 and epochs of 20 (Subtask 1). The number of epochs was decided by preliminary experiments that we

[1] https://sociocom.naist.jp/real-mednlp/
[2] https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking
[3] https://taku910.github.io/mecab/

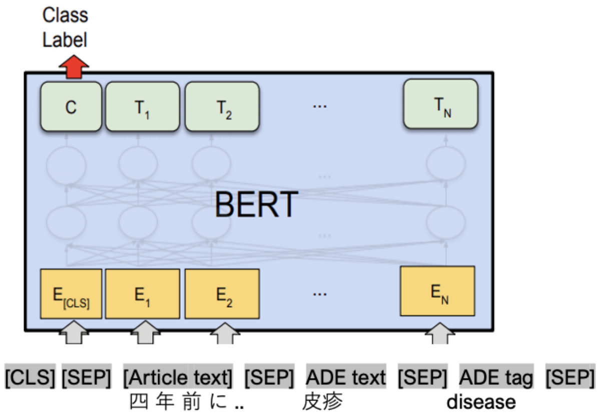

Figure 1: Subtask 3: ADE challenge for MedTxt-CR

confirmed the validation loss reached a plateau. For training, the label of each token is trained so that loss from the correct label is small. For prediction, given a token series X and a label series y, y_i corresponds to the label of token X_i taking label y_i among k labels. The probability is given by $P(y_i|X_i) = S_i(X_i, y_i)/\sum S(X_ik, y_ik)$. To obtain the series with the highest probability to be the series (X, y), the optimal series $y*$ was obtained from $y* = \arg\max \log P(y|X)$.

Since we considered to use all the information in the training set, we predicted all entities even if the task does not require to predict the specific sorts of tags (<f>, <cc>, etc.). We believe that this approach contributes to improving precision.

2.2 Subtask 2

We fine-tuned the same pre-trained model of subtask 1 by using examples of the annotation guideline as a training data. However, dataset is smaller than that of subtask 1 so that 50 epochs which are larger than subtask 1 were required before the losses fell sufficiently.

Although we fine-tuned two models in Subtask 1 for MedTxt-CR and MedTxt-RR respectively, we used the same model to predict entities in the two test sets in Subtask 2.

2.3 Subtask 3

2.3.1 Subtask 3: ADE challenge for MedTxt-CR. We regard the ADE task as a classification task for each drug. Note that we do not handle the table structure as is. Drug by drug, the system estimates the ADE level, consisting of 4 categories: 0-Unrelated, 1-Unlikely, 2-Probably, and 3-Definitely.

We fine-tuned the same pre-trained model of subtask 1 using the ADE data and the case report text before and after the ADE. A list of each ADE entity (case number, case report text, ADE entity, ADE entity label, and ADE levels) was created as training data. For the body of the case report, 50 characters before and after the ADE entity were used. The training data were 117 case reports and the validation data were 14 cases were used for validation data. The hyperparameters of the model were max epochs 10, learning rate 1e-5, max length 128, batch size 32, and optimizer Adam.

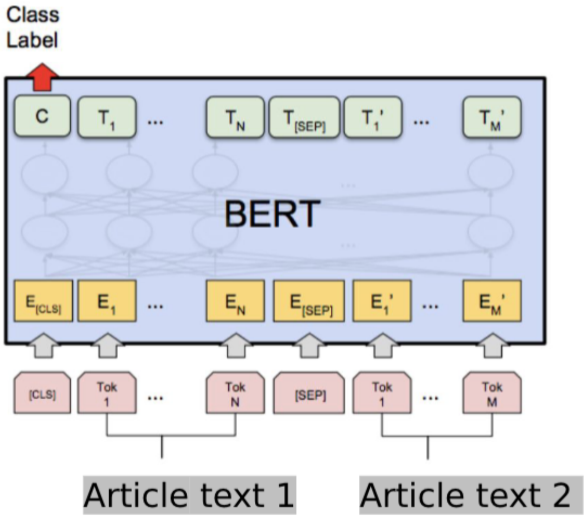

Figure 2: Subtask 3: CI challenge for MedTxt-RR

2.3.2 Subtask 3: CI challenge for MedTxt-RR. We regard the CI task as a binary classification task for two cases, which classify the two cases originate from the same patient or not. To decompose the CI tasks into micro binary tasks, we create all possible pairs of two reports.

From 71 articles with 8 cases assigned, 648 pairs of sentences between the same case and the same number of different cases were created, and labels of 0 or 1 were assigned according to whether the cases were the same or different, and data such as (text1, text2, 0) was constructed. There were 1,296 pairs of these sentences, of which 80% were used as training data and the rest as validation data. For the test data, 250,047 pairs between all articles were created. The same pre-trained model of subtask 1 was finetuned for this task determining whether the paired texts were the same. The hyperparameters of the model were max epochs=15, learning rate=1e-5, max length=128, batch size=32, and Adam was used as the optimizer. The pairs in the test data, (text1, text2) and (text2, text1), were treated as different data, and both pairs were considered the same case only if the model determined that they were identical. Therefore, either pair was judged to be a different case if it was determined that they were not identical. Finally, identical case pairs containing common sentences were grouped together.

3 RESULTS AND DISCUSSIONS

3.1 Subtask 1

Table 1 shows the results of overall performance, Table 2 shows the entity-based performance and Table 3 shows the joint entity-based performance.

3.1.1 MedTxt-CR. The results for Subtask 1 on MedTxt-CR-JA showed a character-based accuracy of 88.18, precision of 61.96, and recall of 68.91. Although our method was simple, F1-score is 65.25, which is the best result among the submission results.

Table 1: Overall performance of subtasks 1 and 2.

		Character-base	Entity-base			Character-base (Joint)	Entity-base (Joint)		
Dataset	Subtask	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
CR-JA	1	88.18	61.96	68.91	65.25	86.05	56.23	62.53	59.21
RR-JA	1	95.13	87.11	87.23	87.17	93.39	83.02	83.14	83.08
CR-JA	2	69.34	20.80	31.81	25.12	65.45	16.14	24.60	19.49
RR-JA	2	89.24	60.50	69.87	64.85	80.45	45.70	45.70	46.68

Table 2: Entity-based performance (F1-score) of subtasks 1 and 2.

	Subtask 1		Subtask 2	
Entity	CR-JA	RR-JA	CR-JA	RR-JA
a	61.97	98.4	18.21	63.16
d	68.99	87.76	24.19	63.48
m-key	68.67	-	1.41	-
m-val	57.63	-	35.64	-
t-key	47.91	-	18.81	-
t-test	46.44	0	0	0
t-val	52.99	-	26.79	-
timex3	80.09	75.86	40.39	47.62

Table 3: Joint entity-based performance (F1-score) of subtasks 1 and 2.

	Subtask 1		Subtask 2	
Entity_Attribute	CR-JA	RR-JA	CR-JA	RR-JA
a	61.97	89.40	18.21	56.89
d	29.63	91.14	0	0
d_positive	62.85	80.99	20.41	56.78
d_suspicious	64.52	78.95	0.97	50.53
d_negative	59.65	82.55	2.32	0
d_general	49.68	0	0	0
m-key_scheduled	0	-	0	-
m-key_executed	58.72	-	2.44	-
m-key_negated	0	-	0	-
m-key_other	47.17	-	0	-
m-val	53.57	-	32.99	-
t-key	47.91	-	18.81	-
t-test_scheduled	0	-	0	-
t-test_executed	45.89	0	0	54.90
t-test_negated	-	0	-	0
t-test_other	0	0	0	0
t-val	52.99	-	26.79	-
timex3_date	85.64	81.48	32.97	68.29
timex3_time	33.33	-	0	-
timex3_duration	55.90	0	0	0
timex3_set	65.12	-	30.14	-
timex3_age	86.61	-	26.36	-
timex3_med	64.56	0	19.63	0
timex3_misc	2.08	-	0	-

In the results for each entity without considering attribute, `timex3` had the highest F1-score, 80.09. `timex3` has a total of 1353 tags in the training set, which results for its high value due to the large amount of data. Although the number of `d`-tags in the training data was the largest (n=2,348), the range of possible entity was slightly lower (68.99). This is expected because it was difficult to obtain an exact match of `d`-tags due to the long span of matchable entities.

In the joint entity-base, the accuracy of the character-base was 86.05, precision was 56.23, and recall was 62.53. Compared to the results without considering attributes, none of the values decreased significantly, suggesting that the task is only slightly more difficult than predicting entities alone.

3.1.2 MedTxt-RR. The MedTxt-RR results for Subtask1 showed a character-based accuracy of 95.13, precision of 87.23, and recall of 87.17. The MedTxt-RR results for Subtask2 showed a letter-based accuracy of 95.13, precision of 87.23, and recall of 87.17.

There are two possible reasons for the higher values than in the MedTxt-CR results. This may be because there were fewer types of tags to predict, and the radiation report had more similar patterns.

The joint based results showed that only timex3_date, the date information, could be predicted, while timex3_duration, the duration information, and timex3_med, the treatment-related information, could not be predicted. This is because the number of tags in the training set was less than two for each.

3.2 Subtask 2

Table 1 shows the results of overall performance, Table 2 shows the entity-based performance and Table 3 shows the joint entity-based performance.

3.2.1 MedTxt-CR. In Subtask 2, character-based accuracy was 69.34, precision was 20.80, and recall was 31.71. As in Subtask 1, timex3 had the highest result for each entity. The results of m-key was 1.41, which was very small. The m-val score was also relatively good at 35.64, although the number of training data was very small (7). This indicates that the learning process of m-val is easier than others because the words of m-val are normally expressed as a numerical value.

3.2.2 MedTxt-RR. In Subtask 2, the character-based accuracy, precision, and recall were 80.45, 45.70, and 45.70, respectively. Accuracy is about 10 points lower, and precision, recall and F1-score are about 40 points lower than the results of Subtask 1, but are higher than those of MedTxt-CR. It is surprising that such a high level of learning can be achieved from guideline examples alone.

Considering that the same model was used and the difference was so large, we can imagine that the nature of the documents is very different between MedTxt-CR and MedTxt-RR. Notably, the prediction results of the models differ greatly depending on the

Table 4: Performance of subtask 3 (ADE).

ADE (val=0)	Presicion	95.21
	Recall	76.04
	F1-score	84.55
ADE (val=1)	Precision	0
	Recall	0
	F1-score	0
ADE (val=2)	Precision	0
	Recall	0
	F1-score	0
ADE (val=3)	Precision	6.98
	Recall	52.94
	F1-score	12.33
Report-level	Precision	12.73
	Recall	77.78
	F1-score	21.88

types of medical documents, even when the models were trained using annotated documents following to the same guidelines.

3.3 Subtask 3: ADE challenge for MedTxt-CR

Compared to the other teams, this was the task with the lowest value. Especially, less correct answers could be obtained for any case other than ADE (val =0).

3.4 Subtask 3: CI challenge for MedTxt-RR

The result is 0.5415. Although the team had the best value among the submitting teams, compared to the results of the other teams in English, there is room for improvement.

4 CONCLUSION

This paper discusses Team NAIST's system for NTCIR 16 RealMedNLP task, composed of three tasks, NER, ADE (a task to extract adverse drug event), and CI (a task to indetify the radiology repots from the same patient). We employed the method using BERT, which is a basic way to solve taskes about natural language. The results of a part of Subtask 1 and Subtask 3 was supirior to the other teams, suggesting that a vanilla BERT is already strong enough, and specified task oriented tunes are sometimes not efficient.

ACKNOWLEDGEMENTS

This work was supported by JST, AIP Trilateral AI Research, Grant Number JPMJCR20G9, and JST AIP-PRISM Grant Number JPMJCR18Y1, Japan.

REFERENCES

[1] 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)*. Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[2] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. 2019. ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission. https://doi.org/10.48550/ARXIV.1904.05342

[3] Yoshimasa Kawazoe, Daisaku Shibata, Emiko Shinohara, Eiji Aramaki, and Kazuhiko Ohe. 2021. A clinical specific BERT developed using a huge Japanese clinical text corpus. *PLoS One* 16, 11 (Nov. 2021), e0259763.

[4] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (Feb. 2020), 1234–1240.

[5] Shuntaro Yada, Yuta Nakamura, Shoko Wakamiya, and Eiji Aramaki. 2022. Real-MedNLP: Overview of REAL document-based MEDical Natural Language Processing Task. In *Proceedings of the 16th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-16, National Center of Sciences, Tokyo.* National Institute of Informatics (NII).