NTCIR 17 Conference: Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies, December 12-15, 2023, Tokyo, Japan
DOI: https://doi.org/10.20736/0002001280

KIS's Stance Classification Model at the NTCIR-17 QA Lab-PoliInfo-4

Akira Nakada
Shizuoka University
Japan
anakada@kanolab.net

Yoshinobu Kano
Shizuoka University
Japan
kano@kanolab.net

ABSTRACT

We participated in the Stance Classification 2 (SC2) subtask of NTCIR-17 QA Lab-PoliInfo-4 as Team KIS. In this paper, we describe our stance (agreement or disagreement) classification model for utterances of Japanese politicians with domain-adaptive training in the political domain. We additionally trained the Japanese pretrained LUKE model with a Masked Language Model (MLM) on the Diet minutes dataset. We also preprocessed the model using the head+tail method to truncate utterances longer than the maximum input length. We found that these methods were effective, achieved the highest score of 97.41% in accuracy in the formal run of the subtask.

KEYWORDS

Stance Classification, Domain Adaptation, Masked Language Model

TEAM NAME

KIS

SUBTASKS

Stance Classification-2

1 INTRODUCTION

NTCIR-17 QA Lab-PoliInfo-4 [4] is a project aimed at developing real-world complex question and answer (QA) techniques using Japanese political information such as local assembly minutes and newsletters. Our team participated in the Stance Classification 2 (SC2) subtask. The goal of the Stance Classification task is to estimate the speaker's stance from the utterances of politicians.

The previous Stance Classification (SC) subtask in NTCIR-15 QA Lab-PoliInfo-2 [3] used utterances made by members of the Tokyo Metropolitan Assembly. This previous task required participants' systems to classify the binary stance of the parties of the Tokyo Metropolitan Assembly (agreement or disagreement) for each agenda, given the topics (agenda), the member' s list and the political denomination list. The evaluation of this previous task is based on and the percentage of correct answers of the inferred stance classifications. However, in actual meeting minutes, utterances expressing agreement or disagreement are often made at the beginning of the meeting, and the problem was that a high degree of accuracy could be obtained simply by finding these statements of agreement or disagreement. The WER99 team [5], which received the highest evaluation score in this previous SC subtask by 99.75% in the automatic evaluation, mainly based on a rule-based algorithm that, for example, finds explicit statements of the stances; relying on such an explicit statement could not resolve context-aware stance classification.

In the SC2 subtask of NTCIR-17 QA Lab-PoliInfo-4, task organizer set up a modified task that aims to predict the stance based only on utterances without explicit statements of agreement or disagreement, due to the issues raised in the previous SC subtask, Specifically, the task is to predict the stance after masking the agreement ("賛成") or disagreement ("反対") part of the utterance. The task organizers provide the minutes of various local government meetings other than those of the Tokyo Metropolitan Assembly, while the dataset of the previous task consisted of the utterance made by the members of the Tokyo Metropolitan Assembly.

We proposed a method to perform additional pretraining on a dataset of a target political domain or task on an existing pretrained model for the general domain. We confirmed the performance improvement of a pretrained LUKE model[7] in Japanese by performing additional training with a Masked Language Model (MLM) on a dataset of Japanese Diet proceedings.

We also confirmed further performance improvement by dividing the target political domain into the entire proceedings and the discussion part, and by additional pretraining datasets with progressively narrower dataset widths.

In order to avoid exceeding the maximum input length of the pretrained models, we truncated the sentences [6] with 128 tokens at the beginning and 384 tokens at the end of the text, while verifying the truncation with several different patterns that vary the number of tokens at the beginning and end of the text.

Our stance classification model achieved the highest score of 97.41% in accuracy of the SC2 subtask in NTCIR-17 QA Lab-PoliInfo-4.

2 RELATED WORK

2.1 Improved learning efficiency during fine tuning

Sun et al. [6] proposed the head+tail method, which uses the beginning and the end of a sentence when truncating the sentence, based on the empirical rule that important information often appears at the beginning and the end of a sentence. They verified their proposed method on an actual document classification task and showed that it achieved better accuracy than simply truncating from the beginning or the end of a sentence. They proposed a truncation method using 128 tokens for the head and 384 tokens for the tail.

In our study, we applied such a truncation of sentences, while validating multiple different patterns varying by the number of tokens at the beginning and the end.

2.2 Domain adaptation through additional pretraining

Gururangan et al. [2] showed the effect of additional pretraining using a dataset of the target domain or task, based on a pretrained model that has already been pretrained on a general corpus. They proposed two methods: *Domain-Adaptive Pretraining (DAPT)*, which pretrains on the dataset of the domain to which the target task belongs, and *Task-Adaptive Pretraining (TAPT)*, which pretrains once on the data of the target task before solving the task.

In our study, we propose to apply an additional pretraining method based on the Masked Language Model (MLM) to a dataset of parliamentary minutes and the utterances of the local council meeting minutes distributed by the task organizers, adapting it to the political domain and task.

3 DATASET

3.1 Stance Classification 2 Subtask Dataset

We describe the task official datasets released for the SC2 subtask in this subsection.

The utterances of politicians were limited to those which can be clearly associated with a specific topic. In addition, the Japanese words *agreement ("賛成")* and *disagreement ("反対")* in the utterances were replaced by special tokens *[STANCE]* to ensure that there are no explicit expressions of the speaker's stance in the utterances. Using these explicit statements of agreement or disagreement, a heuristic rule assigned a gold binary stance label for each utterance. Preliminary experiments have shown that this method rarely assigns wrong labels. Note that, for the training data, an unmasked version before replaced by special tokens *[STANCE]* is also distributed.

Each field of the dataset used for training and test is shown in Table 1. For the training data, 8,534 utterances from 27 municipalities in Saitama Prefecture were used. For the test data, 2,160 utterances from 28 municipalities in Saitama Prefecture (27 municipalities and one more municipality) and 80 utterances from one municipality in Fukuoka Prefecture were used.

3.2 Diet Minutes

3.2.1 Diet Minutes Dataset. we use the minutes of the plenary sessions and committee meetings of the Diet minutes[1], for our additional pretraining on the political domain. We use the utterance texts of the House of Representatives from 2000 to 2022, and the utterance texts of the House of Councillors from 2003 to 2022, which are selected for training. If the input text exceeds 512 tokens, the text is divided into a block(s) by dividing at the period ("。"), so that a block is within 512 tokens. Our Diet minutes dataset is 694,907 blocks in total.

3.2.2 Discussion Part Dataset. We create a discussion part of the Diet minutes by extracting utterances that contain a Japanese word discussion ("討論"). The discussion part dataset is 13,204 blocks in total.

[1]Compiled from 「国立国会図書館小史」(国立国会図書館) https://www.ndl.go.jp/jp/aboutus/outline/history/short_history.html

4 PROPOSED METHODS

4.1 Dealing with long sentences

The head+tail method is applied when the input text exceeds the model's maximum input sequence length of 512 tokens. Head+tail method uses only the specified number of tokens from the beginning and the end when truncating sentences as a preprocessing step.

4.2 Additional pretraining

We applied two additional pretraining methods to a pretrained model that has been pretrained on a general corpus (Wikipedia, etc.). Figure 1 shows an overview of our training methods.

4.2.1 Domain-Adaptive Pretraining(DAPT). Domain-Adaptive Pretraining (DAPT) [2] is a method to perform additional pretraining on the data of the domain to which the target task belongs. In our study, we use the minutes of plenary sessions and committee meetings of the Japanese Diet minutes as the data of the political domain to which the target task belongs. We also propose to use the discussion part of the Diet minutes for additional pretraining as the data of the domain closer to the target task.

The additional pretraining is done with the Masked Language Model (MLM), which is the conventional BERT model [1] pretraining task. The mask probability is the same as in the conventional MLM, where 15% of the total tokens are randomly selected. Of the randomly selected tokens, 80% are replaced with masked tokens, 10% are replaced with another randomly selected token, and the remaining 10% are given to the model as they were in their original tokens.

4.2.2 Task-Adaptive Pretraining (TAPT). Task-Adaptive Pretraining (TAPT) [2] is a method in which additional pretraining is performed once on the target task data before solving the task. In our study, the target task text used for TAPT was the utterance texts of the distributed unmasked traing data.

As with DAPT, MLM is used for additional pretraining.

4.3 Fine-Tuning

To adjust the hyperparameters during fine-tuning, we performed a k-fold cross-validation, where the labeled politician's utterance texts are divided into k pieces, one of which is the validation data and the remaining k-1 pieces are the training data, and the performance calculation is repeated k times.

The final prediction is determined by an ensemble of the predictions of each of the k fold models by a majority vote.

5 EXPERIMENTS AND RESULTS

5.1 Comparison of pretrained models

We compared different pretrained models as the source of our additional pretraining. We used four types of pretrained models: *BERT-base*[2] and *BERT-large*[3] published by Tohoku University, *LUKE-base*[4] and *LUKE-large*[5] provided by Studio-Ousia.

[2]https://huggingface.co/cl-tohoku/bert-base-japanese
[3]https://huggingface.co/cl-tohoku/bert-large-japanese
[4]https://huggingface.co/studio-ousia/luke-japanese-base
[5]https://huggingface.co/studio-ousia/luke-japanese-large

Table 1: Data fields of the stance classification 2 task

Field Name	Explanation
id	Question ID (Japanese local government ID and serial number)
prefecture	Name of the prefecture
assembly	Name of the local government
meeting	Name and serial number of the regular meeting
date	Date of the meeting
speaker	Speaker name of the utterance
utterance	An utterance by an politician whose explicit tokens are replaced with [STANCE]
target	topic of the utterance
stance	'Agreement' or 'Disagreement'

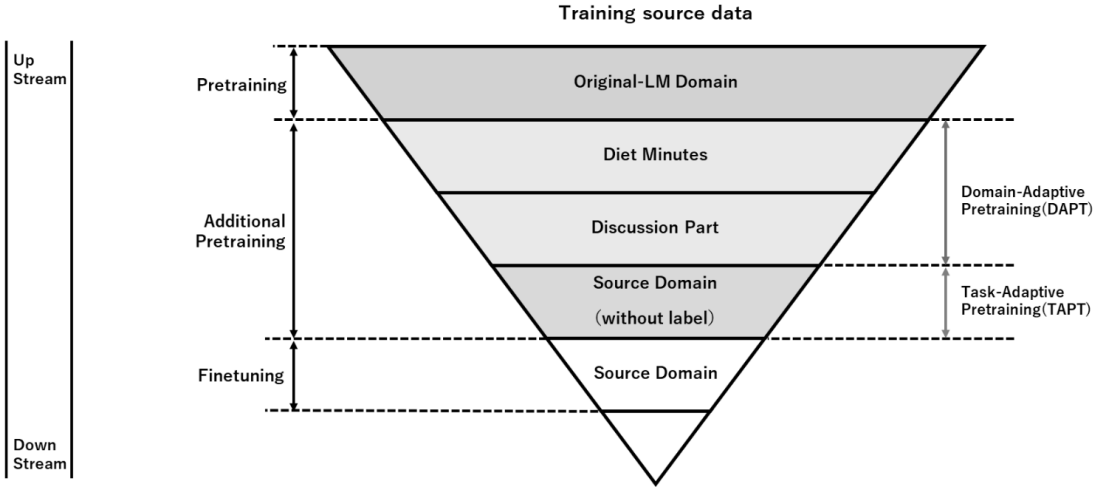

Figure 1: Outline of our training method

BERT model is widely used as a baseline for many natural language processing tasks, and was also used in this study. Recently, LUKE model has been gaining attention for its use of abundant background knowledge by integrating it with knowledge graphs, and has been spreading as a highly accurate model. Therefore, we decided to use it in this study as well.

Table 2 shows the accuracies of the cross-validation using the training data and the accuracy of the test data in the formal run (test data), for each pretrained model with fine-tune only (without additional pretraining).

When comparing the different pretrained models, LUKE-large performed the best among the four pretrained models, with an average accuracy of 0.9453 over k-folds during cross-validation; and a stable performance of 0.9563 when submitted on the test data, evaluated by the test data. Therefore, we selected LUKE-large as the pretrained model for the following experiments of the additional pretraining.

5.2 Effectiveness in dealing with long sentences

We verified five patterns of sentence truncations using 512 (0 from the tail), 384 (128 from the tail), 256 (256 from the tail), 128 (384 from the tail), and 0 (512 from the tail) tokens from the beginning to match with the maximum input sequence length of LUKE-large, which is 512 tokens.

Table 3 shows the accuracies of cross-validations using the training data and the accuracies on the formal run using the test data, for each truncation method. For the conventional method of truncating 512 tokens from the beginning, the average accuracy for each fold of cross-validations was 0.9453, and 0.9563 for submission on the test data. We found that using the beginning and the end of the sentence when truncating the sentences resulted in better performance in all cases, than simply cutting off the tails.

We used the combination of 384 tokens from the beginning and 128 tokens from the end in subsequent experiments, since the average accuracies for each fold of the cross-validations was the best score (0.9568) and its accuracy on the test data showed a stable performance among others.

5.3 Effectiveness of Additional Pretraining

5.3.1 Domain-Adaptive Pretraining（DAPT）. Using the Diet minutes described in the Dataset section, we performed the additional pretraining using MLM. The sentence truncation is performed using 384 tokens from the beginning and 128 tokens from the end.

Table 2: Comparison of pretrained models

model	k-fold cross validation (k=5) using train data			test data
	accuracy (max.)	accuracy (min.)	accuracy (avg.)	accuracy
BERT-base	0.9274	0.9127	0.9191	0.9304
BERT-large	0.9268	0.9004	0.9139	0.9254
LUKE-base	0.9390	0.9285	0.9341	0.9469
LUKE-large	**0.9549**	**0.9349**	**0.9453**	**0.9563**

Table 3: Comparison of truncation methods

head+tail ratio	k-fold cross validation (k=5) using train data			test data
	accuracy (max.)	accuracy (min.)	accuracy (avg.)	accuracy
512 : 0	0.9549	0.9349	0.9453	0.9563
384 : 128	**0.9654**	0.9443	**0.9568**	0.9621
256 : 256	0.9555	**0.9496**	0.9531	**0.9652**
128 : 384	0.9596	0.9420	0.9502	**0.9652**
0 : 512	0.9653	0.9436	0.9531	0.9612

The results of the fine-tuning after the additional pretraining are shown in Table 4. **DAPT1** corresponds to the additional pretraining using the full text of the Diet minutes dataset, and **DAPT2** corresonds to the addional pretraining using the discussion part of the Diet minutes dataset.

Minutes-specific model (DAPT1). Without DAPT, the average accuracy for each fold in the cross-validation was 0.9568 using the training data, and 0.9621 for the test data. DAPT using the full text of the Diet minutes dataset improved the performance, with an average accuracy of 0.9630 for cross-validation of the training data and 0.9705 for the test data.

Discussion-specific model (DAPT2). We performed additional pretraining with MLM using the discussion part dataset, which is a subset of the Diet minutes dataset, described in the previous section.

Without DAPT, the average accuracy of each fold in the cross-validation was 0.9568 for the training data, and 0.9621 for the test data. DAPT with the discussion part dataset did not improve performance, which average accuracy was 0.9551 for the cross-validation using the training data, and 0.9610 for the test data.

However, DAPT using the discussion part dataset in addition to the minutes-specific model improved their performances; the average accuracy of the cross-validation was 0.9625 using the training data, and 0.9728 for the test data.

5.3.2 Task-Adaptive Pretraining (TAPT). We performed additional pretraining using MLM, using the utterance text portion of the training data, which is the Task-Adaptive Pretraining (TAPT). The results of fine-tuning after the additional pretraining are shown as **TAPT** in Table 5. The sentence truncation is same as DAPT, using 384 tokens from the beginning and 128 tokens from the end.

Without TAPT, the average accuracy of the cross-validation was 0.9568 using the training data, and 0.9621 for the test data. TAPT did not improve performance much, with an accuracy of 0.9629 for test data (**TAPT**). However, when TAPT was used in addition to the DAPT using the Diet minutes dataset (**DAPT1+TAPT**), the performance in the test data was improved to 0.9696, while this score is lower than that of the DAPT using Diet minutes dataset alone (**DAPT1**).

DAPT1+DAPT2+TAPT, which corresponds to TAPT in addition to the DAPT model using the discussion part that was further in addition to the DAPT using the Diet minutes data, showed an improvement in performance, with an accuracy of 0.9741 for test data. We submitted this model as our formal run, which performed the best among the participants in the formal run evaluation on the test data.

6 DISCUSSION

We confirmed that the head+tail method of sentence truncation is effective for the minutes data as shown in Table3. In this task, utterance about the stance of an answer often appeared at the beginning and end of sentences, and we believe that this is one of the factors that led to the improvement in performance.

The results shown in Table 4 and Table 5 indicate that the additional pretraining improved performance compared to the original pretrained model. This would be because the political domain data appeared less frequently in the training data of the original pretrained model, and the additional pretraining allowed us to acquire knowledge about the political domain.

We confirmed that the combination of DAPT and TAPT improved the performance more than DAPT or TAPT alone. This implies that DAPT could cover the smallness of the training data in TAPT.

7 CONCLUSION

In this study, we verified the effect of additional pretraining on a pretrained model with a dataset of the target political domain or task. Comparisons with the original pretrained models showed

Table 4: Comparison of Domain-Adaptive Pretraining (DAPT)

method	k-fold cross validation (k=5) using train data			test data
	accuracy (max.)	accuracy (min.)	accuracy (avg.)	accuracy
without DAPT	0.9654	0.9443	0.9568	0.9621
DAPT1 (Diet minutes)	0.9695	**0.9566**	**0.9630**	0.9705
DAPT2 (Disccusion part)	0.9590	0.9490	0.9551	0.9610
DAPT1 + DAPT2	**0.9672**	0.9537	0.9625	**0.9728**

Table 5: Comparison of Task-Adaptive Pretraining (TAPT)

method	k-fold cross validation (k=5) using train data			test data
	accuracy (max.)	accuracy (min.)	accuracy (avg.)	accuracy
without TAPT	0.9654	0.9443	0.9568	0.9621
TAPT	**0.9883**	**0.9830**	**0.9852**	0.9629
DAPT1 + TAPT	0.9672	0.9596	0.9633	0.9696
DAPT1 + DAPT2 + TAPT	0.9736	0.9602	0.9652	**0.9741**

that our additional pretraining improved the performance. Further performance gains were observed when the dataset was narrowed in steps and additional training was performed in accordance with these steps. Because domain specificity increases from upstream to downstream tasks, narrowing and specializing the data set was effective. The additional pretraining using small to medium-sized data has an advantage in terms of learning cost as well, compared with training from scratch.

In the future, we would like to apply our additional pretrained model using the Diet minute dataset to tasks in other political domains to evaluate its generalized performance in the political domain.

ACKNOWLEDGMENTS

We thank Prof. Tetsuji Kuboyama in Gakushuin University for his help in providing the Diet minutes dataset used in this study. This study was partially supported by Mext Kakenhi Japan (JP22H00804, JP20K20509), and SECOM Science and Technology Foundation.

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, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[2] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 8342–8360. https://doi.org/10.18653/v1/2020.acl-main.740

[3] Yasutomo Kimura, Hideyuki Shibuki, Hokuto Ototake, Yuzu Uchida, Keiichi Takamaru, Madoka Ishioroshi, Teruko Mitamura, Masaharu Yoshioka, Tomoyosi Akiba, Yasuhiro Ogawa, Minoru Sasaki, Kenichi Yokote, Tatsunori Mori, Kenji Araki, Satoshi Sekine, and Noriko Kando. 2020. Overview of the NTCIR-15 QA Lab-PoliInfo-2 Task. *Proceedings of The 15th NTCIR Conference*.

[4] Yasuhiro Ogawa, Yasutomo Kimura, Hideyuki Shibuki, Hokuto Ototake, Yuzu Uchida, Keiichi Takamaru, Kazuma Kadowaki, Tomoyoshi Akiba, Minoru Sasaki, Akio Kobayashi, Masaharu Yoshioka, Tatsunori Mori, Kenji Araki, Satoshi Sekine, and Teruko Mitamura. 2023. Overview of the NTCIR-17 QA Lab-PoliInfo-4 Task. *Proceedings of The 17th NTCIR Conference*. https://doi.org/10.20736/0002001326

[5] Yuichi Sasazawa and Naoaki Okazaki. 2020. WER99 at the NTCIR-15 QA Lab-PoliInfo-2 Classification Task. *Proceedings of The 15th NTCIR Conference*.

[6] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to Fine-Tune BERT for Text Classification?. In *Chinese Computational Linguistics*, Maosong Sun, Xuanjing Huang, Heng Ji, Zhiyuan Liu, and Yang Liu (Eds.). Springer International Publishing, Cham, 194–206.

[7] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 6442–6454. https://doi.org/10.18653/v1/2020.emnlp-main.523