JRIRD at the NTCIR-16 QA Lab-PoliInfo-3 Budget Argument Mining

Kazuma Kadowaki
The Japan Research Institute, Limited
Japan
kadowaki.kazuma@jri.co.jp

Shunsuke Onuma
The Japan Research Institute, Limited
Japan
onuma.shunsuke@jri.co.jp

ABSTRACT

The JRIRD team participated in the budget argument mining subtask of the NTCIR-16 QA Lab-PoliInfo-3. This paper reports on our approach to solving this problem and discusses the official results. Our system consists of two BERT models that work independently toward two objectives: argument classification (AC) and related ID detection (RID). The results show that our system performs well, especially for argument classification.

KEYWORDS

argument mining, monetary expression, BERT, argument classification, information retrieval

TEAM NAME

JRIRD

SUBTASKS

Budget Argument Mining

1 INTRODUCTION

Many organizations, including governments and corporations, decide how to spend money by preparing and discussing budget proposals. NLP researchers have attempted to automatically analyze such discussions to trace later the underlying reasons for budgets [1, 2, 5].

The budget argument mining subtask of the NTCIR-16 QA Lab-PoliInfo-3 task [3] ("PoliInfo-3") aims to support such analysis in the political domain. In this subtask, given a set of budget items within budget lists (for example, budget requests) and transcripts of the meetings in the Japanese Diet and local assemblies, participants of the subtask were asked (1) to find budget items related to each monetary expression within the transcripts and (2) to predict the argumentative role of the monetary expressions within the transcripts.

We considered both objectives as classification tasks and constructed two independent BERT models for each task. As a result, the JRIRD, our team achieved a good performance in the subtask, especially in the argument class objective.

This paper describes our budget argument mining subtask approach and its results. The remainder of this paper is organized as follows: Section 2 briefly describes the task settings. We describe our approach in Section 3. Section 4 explains the details of our implementations, and the results of the formal run are presented in Section 5. Finally, Section 6 concludes this paper.

2 TASK SETTINGS

A task overview paper [3] describes the details of the budget argument mining subtask. In this section, we briefly describe task settings.

2.1 Dataset

The dataset consists of (1) a list of budget items for the Japanese Diet and three local governments and (2) transcripts of meetings held at these assemblies.

The budget items are extracted from published budget documents (e.g., budget sheets), and each item contains a name, amount, date, competent ministry/department, and description, among other information.

Each transcript contains the politicians' utterances. The task organizers also extracted a unique set of monetary expressions (i.e., argumentative components) for each utterance. When the same monetary expression is mentioned several times in a single utterance, only one is extracted without noting its position within the utterance. Task organizers then manually classified each monetary expression into one of the following seven argument classes:

(1) Premise: Past and Decisions
(2) Premise: Current and Future / Estimates
(3) Premise: Other (examples, corrections, and others)
(4) Claim: Opinions, Suggestions, and Questions
(5) Claim: Other
(6) Not a monetary expression
(7) Other.

A set of related budget items was annotated manually for each monetary expression in the utterance.

Note that the monetary expressions within the utterances are often summed or rounded or contain notational variations (e.g., Chinese numerals, non-numeral phrases such as "free of charge" or "zero," or general approximation such as "hundreds of billions of yen"). The speakers in the meeting did not mention the exact names of the budgets and descriptions.

The statistics for the dataset are listed in Table 1.

2.2 Tasks

The budget argument mining subtask comprises two objectives: argument classification (AC) and related ID detection (RID).

The AC objective aims to assign an argument class (i.e., the discussion label or argumentative role) to each monetary expression. For a list of monetary expressions within each utterance, participants were required to predict one of the seven classes for each expression. The accuracy score of the labels was used as an evaluation metric.

Table 1: Statistics of the dataset.

(a) Budget items

Government	Budget items	Avg. length of budget name	Avg. length of description
The Japanese Diet	36	23.67	144.81
Fukuoka City	324	13.04	181.69
Ibaraki Prefecture	179	14.91	45.63
Otaru City	229	13.85	64.55

(b) Transcripts of meetings

		Transcripts			Monetary expressions		RID
Dataset	Government	Utterances	Sentences	Characters / sentence	Count	Non-empty RID	Count
Train	The Japanese Diet	363	3,771	35.37	165	11	13
	Fukuoka City	660	8,742	61.04	578	275	369
	Ibaraki Prefecture	570	8,271	62.05	276	17	25
	Otaru City	343	3,059	64.44	229	47	55
Test	The Japanese Diet	123	1,673	31.81	65	1	1
	Fukuoka City	78	2,682	73.62	74	4	8
	Ibaraki Prefecture	191	3,022	67.28	68	3	3
	Otaru City	491	4,372	63.54	313	39	46

The RID objective aims to connect each budget item to the discussions included in the transcripts. For a list of monetary expressions within each utterance and a list of budget items, participants must choose a budget item related to each monetary expression. While multiple budget items may be related to a single monetary expression[1], the precision at 1 (P@1) score was used as the evaluation metric. However, monetary expressions with no related budget items in the gold dataset are excluded when calculating this score.

In addition to the above two evaluation metrics, the task was evaluated using the final score, which counts monetary expressions for which the AC and RID objectives were predicted correctly. See the task overview paper [3] for the definition of this score.

3 OUR SYSTEM

We developed a BERT-based model for each of the two objectives: AC and RID. This section first describes our preprocessor for the dataset and each of the two models.

3.1 Preprocess

To limit the length of model inputs, first, we split each utterance into sentences using GiNZA NLP Library (ja_ginza model)[2]. We then scanned each sentence for each monetary expression in the dataset and obtained the positions at which a certain monetary expression appeared within the sentence.

Note that the exact monetary expression may be mentioned several times in a single utterance or a single sentence. Similarly, a single sentence may contain multiple monetary expressions.

[1]Only 27% (94 of 350) of monetary expressions had more than one related budget item in the training dataset. See the task overview paper for more detailed statistics.
[2]https://megagonlabs.github.io/ginza/

3.2 AC Objective

We regarded the AC objective as a 7-class classification task and trained our BERT model to output an argument class.

For each occurrence of monetary expressions, we feed our model three consecutive sentences, one containing the occurrence and those before and after it, to let it employ the context of the monetary expression when it outputs an argument class for the occurrence. We also inserted special tokens before and after the monetary expression to help our model distinguish it from other occurrences within the three consecutive sentences. Because each monetary expression can appear multiple times in an utterance and our model predicts an argument class for each occurrence of the monetary expression, we chose one class for each monetary expression by majority vote and used this as the prediction of the system.

3.3 RID Objective

The goal of the RID objective is to select a budget item related to monetary expressions. We reformulated this objective as a binary classification task for a pair of single candidate budget items and a sentence that contained a monetary expression.

We regarded any pair of budget items and a sentence containing any related monetary expression as a valid pair, and our RID model judged the likelihood of the pair's validity. We fed our model with two segments for each pair: (1) the concatenation of a budget name and its description from the budget list, and (2) the sentence from the utterance. We extracted candidate budget items from the budget list published by the same government in the same year to create candidate pairs.

Our system predicts a single budget item related to an entire sentence in an utterance. First, we chose a budget item whose pair had the highest likelihood for each sentence. We then chose one of the pairs with the highest likelihood. Our system outputs this pair's budget item corresponding to all monetary expressions in

Table 2: Results of the formal run.

ID	Team	Score	AC	AC_3	RID
299	JRIRD	**0.51064**	**0.58269**	0.85577	0.61702
302	JRIRD	<u>0.48936</u>	0.56538	0.84423	0.61702
300	OUC	0.44681	<u>0.57115</u>	0.83654	**0.65957**
303	JRIRD	0.40426	0.54423	0.83846	0.61702
224	fuys	0.23404	0.56923	**0.88077**	0.34043
239	rVRAIN	0.17021	0.47885	0.85000	0.21277
312	takelab	0.04255	0.39423	0.83269	0.06383
276	SMLAB	0.00000	0.38269	0.73269	0.00000
164	TO	0.00000	0.13462	0.40577	0.00000

the utterance, even if the monetary expression was not an element of the selected pair.

Note that our model always outputs one budget item, even if there may be nothing for the utterance or several in practice[3]. We assumed that humans would always check the system's outputs in real-world use cases and designed our system to support manual fact-checking. That is, we believe that it is better to output extra items than to have missing necessary items. Additionally, because our RID model predicts the likelihood of an entire sentence, we could not predict an individual budget item for each monetary expression directly, which is reserved for our future work.

In addition, our RID model reads the utterance and budget textual information from the budget list, but not the numerical information (i.e., amount) for each budget item. We focused on inputting the context because utilizing a numerical representation was not straightforward. Recall that the monetary expressions within the utterances are often summed or rounded or contain notational variations, as described in Section 2.1.

3.4 Model Variants

We submitted the system's output described above with an ID of 299. For the AC objective, we prepared two more variants of the systems described above and submitted the outputs of these systems, with IDs 302 and 303. Note that the three systems share the same output for the RID objective.

We modified how the ID 302 model distinguished monetary expressions. Instead of adding special tokens before and after each monetary expression, as described above, we masked the monetary expression (i.e., replaced it with a special token).

For the ID 303 model, we modified the context length. Instead of inputting three consecutive sentences for ID 302, we inputted as many sentences as possible, ensuring that the one with the monetary expressions appears in the middle and that both the preceding and following contexts do not exceed 512 characters.

4 IMPLEMENTATION DETAILS

This section explains the implementation of the proposed model in detail.

[3]Only 28% (350 of 1,248) of monetary expressions had at least one related budget item in the training dataset.

4.1 Experimental Environment

We used HuggingFace's Transformers [8] to fine-tune the BERT models[4].

To train our models, we used the NICT BERT Japanese Pretrained Model[5] (32k vocabulary version) as a starting point. Following the model's instructions, we used MeCab-Jumandic [4] to tokenize Japanese sentences and the model's vocabulary to tokenize them into subwords using byte-pair-encoding [7] before inputting them into our BERT models.

Our experiments were conducted on a single NVIDIA TITAN RTX GPU. On average, training our BERT models took 3 min per epoch for the AC objective and 15 min per epoch for the RID objective.

4.2 Hyper-Parameter Selection

We prepared our validation dataset to fine-tune our BERT model because the task organizers only provided the training and test datasets. We split the first few utterances from the training dataset for each government ensuring that the number of monetary expressions in the split did not exceed 10%, and used this split for validation.

We then performed a hyper-parameter search and chose a model whose outputs achieved the best score on our validation datasets based on the evaluation scripts provided by the task organizers[6].

For the AC objective, we attempted every combination of epochs of {1, 2, 3, 5, 10, 15, 20, 25, 30} and a learning rate of {2e-5, 3e-5, 5e-5}. To achieve the RID objective, we attempted every combination of epochs of {1, 2}[7] and a learning rate of {5e-6, 1e-5, 2e-5, 3e-5, 5e-5}. The other hyperparameters remained fixed throughout our experiments: a batch size of 32 and maximum sequence length of 256.

5 RESULTS OF THE FORMAL RUN

Our official formal run results are presented in Table 2. Bolded and underlined scores indicate the best and second-best results among all the participants[8], respectively. In addition to the AC and RID metrics, we also added the AC_3 metric, which regards the AC objective as a 3-class classification task (i.e., premise, claim, and others).

The results show that our approach (ID 299) achieved the best performance in terms of the AC objective compared to the other approaches. This also indicates that our models achieved the second-best performance for the RID objective[9]. We observed that masking the numeral expressions (ID 302) worsened performance. We

[4]More precisely, we used the `BertForSequenceClassification` class of implementation, which consists of a linear layer on top of the pooled output (i.e., the final hidden vector of the `[CLS]` token).

[5]https://alaginrc.nict.go.jp/nict-bert/index.html

[6]In cases where multiple models had the same score, we chose a model using the F1 score for all the BERT outputs. The outputs here include those not outputted by our system after the majority vote for the AC objective and pairs ranked second or lower for the RID objective.

[7]In our preliminary experiment, we confirmed that increasing the number of epochs did not improve the performance of the RID objective. This is probably because the RID objective has massive inputs, as we considered all possible pairs as inputs.

[8]We only included the best results from each of the other participants in Table 2. See the task overview paper [3] for the results for other participants.

[9]Table 2 does not underline our RID results because the OUC team had several submissions that shared the same RID results as the best model.

Table 3: Confusion matrix of our model for the AC objective (ID 299).

Label		Prediction							
		Premise Past	Premise Future	Premise Other	Claim Opinion	Claim Other	Not a money	Other	Total
Gold	Premise: Past & Decisions	**43**	35	23	0	0	0	0	101
	Premise: Current & Future / Estimates	16	**161**	13	5	0	0	1	196
	Premise: Other	14	41	**84**	6	0	0	0	145
	Claim: Opinions, Suggestions & Questions	1	28	6	**7**	0	0	0	42
	Claim: Other	0	2	2	0	**0**	0	0	4
	Not a monetary expression	7	12	3	0	0	**8**	0	30
	Other	1	0	1	0	0	0	**0**	2
	Total	82	279	132	18	0	8	1	520

Table 4: Per-government results of our model (ID 299).

Gov. Code	Government	Score	AC	AC_3	RID
-	The Japanese Diet	1.00000	0.47692	0.61538	1.00000
401307	Fukuoka City	0.25000	0.63514	0.91892	0.25000
080004	Ibaraki Prefecture	0.66667	0.79412	0.95588	0.66667
012033	Otaru City	0.51282	0.54633	0.86901	0.64103
Total		0.51064	0.58269	0.85577	0.61702

also confirmed this tendency in the NTCIR-16 FinNum-3 task [1, 6].

Furthermore, extending the context (ID 303) negatively affected the AC objective's performance, suggesting that this objective does not require a broad context. For future work, we will build a model that takes a shorter context (e.g., a sentence or even shorter phrases) as input, rather than three sentences, as used for our ID 299 and 302 models.

Table 3 presents the confusion matrix of the proposed model for the AC objective. The bolded numbers indicate the correct predictions. Our model seems to prefer outputting Premise classes, which are the gold label for most of the dataset. This seems to be why our model achieved higher accuracy in the objective, despite not predicting minor labels correctly.

The performance of our model (ID 299) for each government is shown in Table 4. The results suggest that the performance of our model varies depending on the government. In future work, we may need to build a model that considers the characteristics of each government.

6 CONCLUSIONS

We participated in the budget argument mining subtask of the PoliInfo-3 task, where we constructed a model consisting of two BERT models: one for argument classification (AC) and another for related ID detection (RID) objectives. Consequently, our team achieved good performance in the subtask, especially in the argument class objective.

In future work, we will attempt to improve the model performance, especially for minor classes for the AC objective, by considering the effect of context length and the characteristics of each government. We will also attempt to utilize numerical representations for each budget item (e.g., amount) to improve the performance of the RID objective. Furthermore, we will investigate the effect of using a joint learning approach for the two objectives instead of two independent models.

As another research direction, we will attempt to investigate the performance of our approach on different datasets, including those from other domains, such as finance and economics.

REFERENCES

[1] Chung-Chi Chen, Hen-Hsen Huang, Yu-Lieh Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2022. Overview of the NTCIR-16 FinNum-3 Task: Investor's and Manager's Fine-grained Claim Detection. In *Proceedings of the 16th NTCIR Conference on Evaluation of Information Access Technologies*. Tokyo, Japan.

[2] Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2020. Overview of the NTCIR-15 FinNum-2 Task: Numeral Attachment in Financial Tweets. In *Proceedings of the 15th NTCIR Conference on Evaluation of Information Access Technologies*. Tokyo, Japan.

[3] Yasutomo Kimura, Hideyuki Shibuki, Hokuto Ototake, Yuzu Uchida, Keiichi Takamaru, Madoka Ishioroshi, Masaharu Yoshioka, Tomoyoshi Akiba, Yasuhiro Ogawa, Minoru Sasaki, Ken-ichi Yokote, Kazuma Kadowaki, Tatsunori Mori, Kenji Araki, Teruko Mitamura, and Satoshi Sekine. 2022. Overview of the NTCIR-16 QA Lab-PoliInfo-3 Task. In *Proceedings of the 16th NTCIR Conference on Evaluation of Information Access Technologies*. Tokyo, Japan.

[4] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying Conditional Random Fields to Japanese Morphological Analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Barcelona, Spain, 230–237.

[5] John Lawrence and Chris Reed. 2019. Argument Mining: A Survey. *Computational Linguistics* 45, 4, 765–818.

[6] Shunsuke Onuma and Kazuma Kadowaki. 2022. JRIRD at the NTCIR-16 FinNum-3 Task: Investigating the Effect of Numerical Representations in Manager's Claim Detection. In *Proceedings of the 16th NTCIR Conference on Evaluation of Information Access Technologies*. Tokyo, Japan.

[7] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 1715–1725.

[8] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 38–45.