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

Keiyu Nagafuchi
Graduate School of Information Science and Technology, Hokkaido University
Japan
nagafuchi@ist.hokudai.ac.jp

Rin Sasaki
Otaru University of Commerce
Japan
g2019162@edu.otaru-uc.ac.jp

Seiya Oki
Otaru University of Commerce
Japan
g2019081@edu.otaru-uc.ac.jp

Yasutomo Kimura
Otaru University of Commerce
Japan
kimura@res.otaru-uc.ac.jp

Kenji Araki
Faculty of Information Science and Technology, Hokkaido University
Japan
araki@ist.hokudai.ac.jp

ABSTRACT

The OUC team participated in the Budget Argument Mining subtask of the NTCIR-16 Question Answering Lab for Political Information 3 (QA Lab-PoliInfo-3). In this paper, we report on our methods for this task and discuss the results. We performed argument classification using a fine-tuned BERT classifier. This method showed the second highest score (0.5716) among the participants on the test data. We also performed linking of relatedIDs using TF-IDF vectorization of documents and calculation of their cosine similarity. This method showed the highest score (0.6596) among the participants on the test data.

KEYWORDS

BERT, TF-IDF

TEAM NAME

OUC

SUBTASKS

Budget Argument Mining

1 INTRODUCTION

The Budget Argument Mining subtask of the NTCIR-16 Question Answering Lab for Political Information 3 (QA Lab-PoliInfo-3) [1] aims to connect published budget documents with discussions included in the minutes of budget meetings. Specifically, when a budget item (which includes a budget amount, the name of competent ministry or department, and an explanation) is given, this task aims to find politicians' statements related to the budget (statements including expressions of the amount of money) in the minutes and assigns three discussion labels, including Claim, Premise Other.

In this paper, we report on the methods developed by the OUC team to perform this task and discuss the results. This task consists of two subtasks, including argument classification (AC) and linking relatedIDs (RID). Although participating systems must answer both AC and RID to obtain a final score, AC and RID could be considered as independent tasks. Therefore, separating the AC and RID methods once they are considered is reasonable. In Section 2, we discuss the methods used in argument classification. In Section 3, we explain the method of linking RID. In Section 4, we show the results of our method in a formal run. In Section 5, based on the results, we discuss our method and this task.

2 METHODS: ARGUMENT CLASSIFICATION

This section describes our method to perform argument classification.

Argument classification tasks are defined as that of classifying the components of arguments into seven argument classes, which are as follows.

(1) Premise : Past and Decisions
(2) Premise : Current and Future / Estimates
(3) Premise : Other
(4) Claim : Opinions, suggestions, and questions:
(5) Claim : Other
(6) It is not a monetary expression.
(7) Other

This suggests that argument classification is a simple seven-valued classification task for utterances containing monetary expressions into seven classes [1].

We created a rule-based classifier, a Bag of Words-based classifier, and a BERT classifier to perform argument classification. The following subsections show the creation procedure of each classifier.

2.1 Rule-based classifier

The rule-based classifier is based on whether a particular expression is included in an utterance containing monetary expressions, and classifies it into the corresponding argument class.

The classification rules are shown below. If an utterance matches more than one condition, the rule on top is given priority.

(1) If the utterance contains the word "行われた" [1]: "Premise: Past and Decisions"
(2) If the utterance contains the words "見込み" or "考えて" [2]: "Premise : Current and Future / Estimates"

[1] done
[2] expected or think

Table 1: BoW-based classifier

Method name	Tokenizer	Vectorizer	Classifier
BoW_LSVC	BoW	MeCab IPADIC	LinearSVC
BoW_noun_LSVC	BoW(noun)	MeCab IPADIC	LinearSVC
TFIDF_LSVC	TF-IDF	MeCab IPADIC	LinearSVC
TFIDF_Sudachi_LSVC	TF-IDF	Sudachi Mode B	LinearSVC
BoW_SVC	BoW	MeCab IPADIC	SVC
BoW_RF	BoW	MeCab IPADIC	RandomForest
BoW_SGD	BoW	MeCab IPADIC	SGD
BoW_Ensenble	BoW	MeCab IPADIC	Ensenble

(3) If the utterance contains the words "提案する" or "質問する" [3]: "Claim : Opinions, suggestions, and questions"
(4) If the utterance contains the word "訂正" [4]: "Premise : Other"
(5) Otherwise: "Other"

2.2 BoW-based classifier

The Bag of Words (BoW)-based classifier is based on the idea of converting utterances containing monetary expressions into vectors using BoW and classifying them into argument classes using an algorithm.

The following procedure is used to create a BoW-based classifier.

(1) Extract pairs of utterances containing monetary expressions and argument class from training data of minutes and construct a dataset. If duplicate utterances are present, delete them, leaving only a single copy of each unique utterance.
(2) Segment the utterances in the dataset into words using a morphological analyzer, and create a BoW.
(3) Using the BoW, convert the utterances in the dataset into vectors, and use the argument class corresponding to the utterance to train the classifier.

We modified the morphological analyzers and classification algorithms to create a total of eight BoW-based classifiers. We used MeCab[5] and Sudachi [2] as morphological analyzers. We also used scikit-learn[6] to train the classifiers. Table 1 shows the details of the eight classifiers we created.

2.3 BERT classifier

The BERT classifier was created by fine-tuning the pre-trained BERT model published by Tohoku University[7] to fit the argument classification of this task.

The procedure for creating the BERT classifier is shown below.

(1) Extract a pair of utterances containing monetary expressions and an argument class from training data of minutes and construct a dataset. If there are duplicate utterances, delete them, leaving only a single copy of each unique utterance.
(2) Divide the dataset into training, validation, and testing data at a ratio of 6:2:2.

[3] suggest or ask
[4] correct
[5] https://taku910.github.io/mecab/
[6] https://scikit-learn.org/stable/
[7] https://huggingface.co/cl-tohoku/

Table 2: BERT classifier

Method name	Base model
BERT_base	bert-base-japanese-whole-word-masking
BERT_base_v2	bert-base-japanese-v2
BERT_large	bert-large-japanese
BERT_base_ml64	bert-base-japanese-whole-word-masking

(3) Perform fine-tuning using training and validation data. We used 10 epochs , and the model that minimizes the loss was adopted as the classifier.

We created a total of four classifiers for BERT [3] by changing the pre-trained model of the underlying BERT, the sequence length and the batch size during training. Table 2 shows the pre-trained BERT models that are the basis of the four classifiers created. The sequence length was the maximum number of words in the training dataset + 2 for BERT_base and BERT_base_v2, 128 for BERT_large and 64 for BERT_base_ml64. The batch size during training is set to 16 only for BERT_base_v2 and 32 for the others.

3 METHODS: RID LINKING

This section describes our method to solve the linking RID task.

RIDs are given to link a budget item (budgetID) to the relevant argumentative component [1].

We transformed one sentence of an utterance containing budget item information and monetary expression into a document vector, and RIDs were associated with those whose cosine similarity exceeded a threshold. We used Sentence-BERT [4] and TF-IDF to convert to document vector to perform RID linking. The following subsections show the description of the method using Sentence-BERT and the method using TF-IDF.

3.1 Sentence-BERT

Sentence-BERT is a specialized model designed to compute document vectors. Our Sentence-BERT is a fine-tuned version of the pre-trained BERT model published by Tohoku University [8]. To perform fine-tuning, we used the JSNLI dataset, which is publicly available from Kyoto University [5]. Only a single epoch was used. We created two models, one of which was trained on the NLI task, and the other of which was trained on the STS task. The model trained by the NLI task uses the JSNI dataset to infer whether the relation between any two given sentences involves contradiction, entailment, or neutral. For the model trained in the STS task, the JSNLI dataset was processed and trained to have a cosine similarity of 0 when the given two sentences were contradiction, 1 when they were entailment, and 0.5 when they were neutral. We refer to the two models as SBERT_NLI and SBERT_STS.

3.2 TF-IDF

In the method of calculating document vectors using TF-IDF, we tried several word segmentation granularities using Sudachi [2]. Sudachi, a Japanese morphological analyzer, has three modes: mode A, designed to perform short word segmentation, mode B for medium word segmentation, and mode C for long word segmentation. We

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

Table 3: Our official scores on the formal run

ID	Method name (AC+RID)	Score	AC	RID
300	BERT_base + TFIDF_modeA	**0.4468**	**0.5712**	**0.6596**
309	BERT_base_ml64 + TFIDF_modeA	0.4255	0.5385	**0.6596**
263	BoW_SVC + TFIDF_modeA	0.4255	0.4827	**0.6596**
308	BERT_large + TFIDF_modeA	0.4043	0.5615	**0.6596**
305	BERT_base_v2 + TFIDF_modeA	0.4043	0.5577	**0.6596**
251	BoW_LSVC + TFIDF_modeA	0.4043	0.4904	**0.6596**
252	TFIDF_Sudachi_LSVC + TFIDF_modeA	0.4043	0.4885	**0.6596**
250	BoW_LSVC + TFIDF_modeB	0.4043	0.4904	0.5745
301	BoW_Ensenble + TFIDF_modeA	0.3830	0.4750	**0.6596**
277	BoW_RF + TFIDF_modeA	0.3830	0.4231	**0.6596**
248	BoW_LSVC + TFIDF_modeC	0.3830	0.4904	0.5532
278	BoW_SGD + TFIDF_modeA	0.2979	0.4615	**0.6596**
230	BoW_LSVC + SBERT_NLI	0.1489	0.4904	0.1702
177	Rulebased + SBERT_NLI (dry run ver.)	0.1277	0.3731	0.2128
234	TFIDF_LSVC + SBERT_NLI	0.0851	0.4750	0.1702
233	BoW_noun_LSVC + SBERT_NLI	0.0851	0.4231	0.1702
219	Rulebased + SBERT_NLI	0.0851	0.3731	0.1702
211	Rulebased + SBERT_NLI	0.0851	0.3731	0.1702
212	Rulebased + SBERT_STS	0.0851	0.3731	0.1489
183	Rulebased + Doc2Vec	0.0000	0.3731	0.1277
217	Rulebased + miss	0.0000	0.3731	0.0000

used this Sudachi function to create three TF-IDF models with different word segmentation granularities. In addition, Sudachi has a word normalization function. In all TF-IDF modeling, only nouns were extracted and words were normalized. We refer to the three models as TFIDF_modeA, TFIDF_modeB, and TFIDF_modeC.

4 RESULTS

This section describes our official results on the Budget Argument Mining subtask.

Table 3 shows the scores of our methods (AC+RID) at the formal run. We tried various combinations of AC and RID methods and submitted them as a formal run. Among our methods, the BERT_base and TFIDF_modeA methods of ID300 showed the highest scores (0.44681). This was the third highest score among the task participants.

As mentioned in Section 1, Budget Argument Mining is evaluated with a score that takes into account both AC and RID. However, because AC and RID are considered independent tasks, the methods and scores for each task are described in a separate table.

Table 4 shows the AC scores. The BERT_base method exhibited the highest score (0.57115). The results showed that the methods with the highest AC scores were those that used BERT.

Table 5 shows the RID scores. The TFIDF_modeA method showed the highest score (0.6596). Unlike AC, the method using SBERT resulted in a lower score. In contrast, the method using TF-IDF showed a higher score. Also, the shorter the length of the word segmentation, the higher the score.

5 DISCUSSIONS

This section discusses our results on the Budget Argument Mining subtask. We have proposed several methods to perform this task. In this section, we discuss the difficulty of problems based on the number of correct and incorrect answers of our proposed method.

Table 4: Our official AC scores on the formal run

Method name	AC Score
BERT_base	**0.5712**
BERT_large	0.5615
BERT_base_v2	0.5577
BERT_base_ml64	0.5385
BoW_LSVC	0.4904
TFIDF_Sudachi_LSVC	0.4885
BoW_SVC	0.4827
TFIDF_LSVC	0.4750
BoW_Ensenble	0.4750
BoW_SGD	0.4615
BoW_noun_LSVC	0.4231
BoW_RF	0.4231
Rulebase	0.3731

Table 5: Our official RID scores on the formal run

Method name	RID score
TFIDF_modeA	**0.6596**
TFIDF_modeB	0.5745
TFIDF_modeC	0.5532
SBERT_NLI	0.1702
SBERT_STS	0.1489

Table 6: Number of argument classes in gold standard data

Argument class	Number
Premise : Past and Decisions	101
Premise : Current and Future / Estimates	196
Premise : Other	145
Claim : Opinions, suggestions, and questions	42
Claim : Other	4
It is not a monetary expression	30
Other	2

Table 7: Number of incorrect argument classes for all our methods

Argument class	Number
Premise : Past and Decisions	26
Premise : Current and Future / Estimates	0
Premise : Other	19
Claim : Opinions, suggestions, and questions	25
Claim : Other	4
It is not a monetary expression	23
Other	2

5.1 Argument classification

Table 6 shows the number of correct answers in the test data, that is, the number of each argument class in the gold standard data.

Of these correct answers, “Premise : Current and Future / Estimates” was the only argument class that was answered correctly by all of our proposed methods of argument classification. The number of correct answers was 107.

Table 7 shows the number of incorrect argument classes for all of our proposed methods.

From the two tables, it may be observed that the percentages of correct answers for “Claim : Opinion, suggestion, and question”, “Claim : Other”, “It is not a monetary expression” and “Other” were

low. In particular, "Claim: Other" and "Other" were incorrect in all methods.

These suggest that "Premise" classes present relatively easy problems in argument classification. In contrast, difficult problems include "Claim", "It is not monetary expression", and "Other" classes. This probably occurred because the argument classes in the dataset were biased and had a significant impact on the training process of the classifier.

5.2 Linking RID

When we checked the statements with RID in the gold standard data, we found that most of them had the keywords related to the budget item in the previous or following statement. Therefore, it is likely that the TF-IDF method showed good results in linking RID. In addition, when we checked the statements of the questions that were answered incorrectly by the TF-IDF method, we found that many of them required us to infer the relevant budget items from the surrounding context. The poor results of the Sentence-BERT method may be attributed to the fact that the actual input text often omits explanations of budget items. Because only one sentence of the utterance was used in the current method, future research should consider a method that also takes the surrounding context into account.

6 CONCLUSIONS

The OUC team participated in the Budget Argument Mining subtask of the NTCIR-16 Question Answering Lab for Political Information 3 (QA Lab-PoliInfo-3).

We performed argument classification using a fine-tuned BERT classifier. This method showed the second-highest score (0.57115) among the participants in the testing data. We also performed RID linking using TF-IDF vectorization of documents and calculation of their cosine similarity. This method showed the highest score (0.6596) among the participants on the testing data.

We also discussed this task using the results of our method. Most of the questions for which our method predicted correct classes in argument classification were in the "Premise" class, and most of the questions for which our method predicted incorrect classes were in the other classes. Therefore, we found that it was necessary to consider the class bias of the dataset. When we checked the statements with RID in the gold standard data, we found that most of them had keywords related to the budget item written before and after the statement. In addition, when we checked the statements of the questions that were answered incorrectly by the TF-IDF method, many of them required us to infer the relevant budget items from the surrounding context. Because only one sentence of the utterance was used in the current method, a method that also considers the surrounding context must be developed in the future.

REFERENCES

[1] Yasutomo Kimura, Hideyuki Shibuki, Hokuto Ototake, Yuzu Uchida, Keiichi Takamaru, Madoka Ishioroshi, Masaharu Yoshioka, Tomoyoshi Akiba, Yasuhiro Ogawa, Minoru Sasaki, Kenichi Yokote, Kazuma Kadowaki, Tatsunori Mori, Kenji Araki, Teruko Mitamura, and Satoshi Sekine. Overview of the ntcir-16 qa lab-poliinfo-3 task. *Proceedings of The 16th NTCIR Conference*, 6 2022.

[2] Kazuma Takaoka, Sorami Hisamoto, Noriko Kawahara, Miho Sakamoto, Yoshitaka Uchida, and Yuji Matsumoto. Sudachi: a japanese tokenizer for business. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, may 2018.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[4] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[5] Takumi Yoshikoshi, Daisuke Kawahara, and Sadao Kurohashi. Multilingualization of a natural language inference dataset using machine translation (in japanese). *IPSJ SIG Technical Report (NL)*, Vol. 2020, No. 6, pp. 1–8, 2020.