A Cascade Model for Argument Mining in Japanese Political Discussions: the QA Lab-PoliInfo-3 Case Study

Ramon Ruiz-Dolz
Valencian Research Institute for Artificial Intelligence
Universitat Politècnica de València
València, Spain
raruidol@dsic.upv.es

ABSTRACT

The rVRAIN team tackled the Budget Argument Mining (BAM) task, consisting of a combination of classification and information retrieval sub-tasks. For the argument classification (AC), the team achieved its best performing results with a five-class BERT-based cascade model complemented with some handcrafted rules. The rules were used to determine if the expression was monetary or not. Then, each monetary expression was classified as a premise or as a conclusion in the first level of the cascade model. Finally, each premise was classified into the three premise classes, and each conclusion into the two conclusion classes. For the information retrieval (i.e., relation ID detection or RID), our best results were achieved by a combination of a BERT-based binary classifier, and the cosine similarity of pairs consisting of the monetary expression and budget dense embeddings.

KEYWORDS

Argument Mining, Debate Analysis, Natural Language Processing

TEAM NAME

rVRAIN

SUBTASKS

Budget Argument Mining (BAM)

1 INTRODUCTION

The automatic analysis of natural language arguments has made possible to improve computer systems for human assistance in the domains of medicine [14], academic research [4], web discourse analysis [8], and autonomous debate [24] among others. The argument mining task is present in many different domains and instances [12]. However, due to its heterogeneity and complexity, it is considered as an important challenge in the Natural Language Processing (NLP) research community. The underlying linguistic structures in natural language argumentation present a great challenge for both, the human annotation of new corpora [23]; and the training/evaluation of new models for domain-independent argument mining [2, 22] or for different instances of the problem belonging to the same domain (e.g., legal) [20, 27]. Thus, advances in argument mining research will benefit from as many as different viewpoints (e.g., domains and/or task instances) approaching this task.

In this work, we describe the participation of our team *rVRAIN* to the Budget Argument Mining (BAM) task organised for the *QALab PoliInfo 3*[1] and the *NTCIR-16*[2]. The BAM is a combination of classification and information retrieval sub-tasks in the domain of political debate analysis. First, the argument classification sub-task is aimed at determining if a given monetary expression belongs to an argument, and which is its argumentative purpose (i.e., either claim or premise). Second, the relation ID detection sub-task is aimed at finding relations between monetary expressions uttered in an argumentative discourse and political budget items.

Our approach presents a BERT-based cascade model for argument mining in japanese political discussions. The model proposed in this paper for solving the BAM task tackles independently the argument classification and the relation ID detection tasks. For the former, we propose the use of handcrafted rules to determine if an expression is monetary or not. Then, a BERT-based cascade model is trained to classify each argumentative monetary expression into premise or claim, and their subsequent sub-classes (i.e., three premise and two claim sub-classes). For the latter, a BERT-based binary classifier is trained to identify possible relations between political budget items and monetary expressions. Each (possible) identified relation is then scored using the cosine similarity, and only the top relations are brought into consideration.

The rest of the paper is structured as follows. Section 2 reviews the related research and contextualises the contribution of this paper to the area of argument mining. Section 3 briefly defines the BAM task and the corpus used to carry out our experiments. Section 4 presents the architecture of the model proposed for solving the BAM task. Section 5 depicts the observed results in three different stages of the competition. Finally, Section 6 discusses the obtained results and analyses future lines of research and open challenges.

2 RELATED WORK

Argument mining approaches the automatic identification, classification and structuring of argumentative natural language [19]. It has been typically decomposed into different sub-tasks in the literature [3, 12]: argumentative discourse segmentation, argument component detection, and argumentative relation identification. Each one tackles a different step belonging to the global goal of argument mining.

Argumentative discourse segmentation is the task of detecting argument spans in a given natural language input. For example, identifying where an argumentative component begins and ends throughout the full interventions of politicians in a discussion. A basic approach for this task was to consider complete sentences and classify them into *argument*/*non-argument* [19]. In [13], the authors

[1] https://poliinfo3.net/
[2] http://research.nii.ac.jp/ntcir/ntcir-16/index.html

propose an unsupervised approach for claim segmentation based on the appearance of common linguistic structures used for argumentation (e.g., "that"). However, recent research has emphasised the relevance of context for improving the segmentation of arguments and argumentative components in natural language inputs [1]. In spoken dialogue it is common to omit contextual information to ease its flow, aimed at overcoming this problem a cascade model for identifying argumentative propositions completing the missing context is proposed in [10].

The detection and classification of argument components is the argument mining task aimed at understanding the argumentative purpose of the previously segmented text. Research in this topic has usually focused on the identification of argumentative evidence and on the premise/claim classification [12]. We will focus on the latter since it has a direct relation with the BAM task and the model proposed in this work. The argument component detection is a very descriptive representation of the previously mentioned existing heterogeneity in argument mining research. Initially introduced in [19], the task was instanced as a binary classification problem. The authors make use of classical machine learning algorithms to predict premise and claim classes for the argumentative expressions. Subsequent research focused on a linguistic enrichment of the task proposed a new instance where up to four classes (i.e., *major claim*, *claim*, *premise*, *none*) were considered [25]. Some of the latest research in this topic has explored the use of end-to-end neural network-based architectures [7, 17], graph convolutional networks [18], and attention-based architectures [26] to improve previous experimental results.

Finally, the argumentative relation identification task focuses on detecting argumentative structures between the argumentation components (e.g., premises or claims). This task has been classically considered one of the most complex tasks in argument mining, and approached as a sentence pair classification problem with two classes (i.e., attack and support) [5, 7, 9]. Recent research has investigated the behaviour of state-of-the-art NLP techniques when approaching a cross-domain multi-class instance of this task [22]. However, since the BAM task and our proposed model does not approach this sub-task of argument mining, we will not go any further into this aspect.

3 BUDGET ARGUMENT MINING

This work approaches the Budget Argument Mining (BAM) instance of the argument mining task. The BAM is aimed at improving the automatic argumentative analysis of political discussion transcripts through the use of NLP techniques. It includes the argumentative discourse segmentation and the argument component detection sub-tasks. For that purpose, monetary expressions are detected in the transcripts, and it must be determined if an expression belongs to an argument or not, and which is its argumentative role in the discussion. Furthermore, the required analysis is enriched with the relation of each monetary expression with a political budget item. This way, the resulting analysis will provide a set of argumentative components and their type detected in the transcripts of a discussion, and a set of relations between the arguments and budget items.

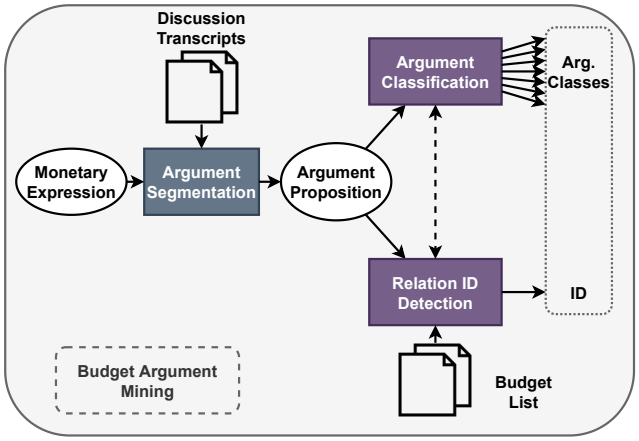

Figure 1: Budget Argument Mining task diagram.

Therefore, the BAM consists of two different sub-tasks: the argument classification (AC) and the relation ID detection (RID) (see Figure 1). A complete description of the whole task can be found in its overview [11]. However, the basic ideas of BAM are presented in the following sections in order to make this paper self-contained.

3.1 Argument Classification (AC)

The AC sub-task is aimed at covering the two first parts of argument mining: segmentation and classification. Thus, for a given monetary expression appearing in an utterance, we need to analyse if it belongs to an argumentative proposition, and which is its role in argumentation. First, the argumentative propositions containing the monetary expressions need to be segmented from the natural language transcripts. Second, these segments must be classified into seven different argumentative classes: (i) Premise: Past and Decisions; (ii) Premise: Current and Future; (iii) Premise: Other; (iv) Claim: Opinions, suggestions and questions; (v) Claim: Other; (vi) Not monetary expression; and (vii) Other.

3.2 Relation ID Detection (RID)

The RID sub-task is aimed at determining if the monetary expressions uttered in the discussion are related to a specific item in the budget list. For this purpose, each argument containing any monetary expression must be segmented. Then, a relation between the segmented text and the budget items must be established.

3.3 Data

The data released for the BAM task is structured into three different documents: the budget data (*PoliInfo3_BAM-budget.json*), the training data (*PoliInfo3_BAM-minutes-training.json*), and the test data (*PoliInfo3_BAM-minutes-test.json*). Each document contains information from the Japanese national diet, and from three different local circumscriptions (i.e., Otaru, Ibaraki, and Fukuoka).

The budget document is a list consisting of 768 different budget items. Each budget item has eleven descriptive features: an *identifier*, a *title*, a *url*, an *item*, the *budget amount*, a list of *categories*, the *types of account*, the *department*, *last year's budget*, a *description*, and a *budget difference*.

Table 1: Class distribution of the BAM training data.

	Premise			Claim			
	Past	Future	Other	Opinions	Other	**Non monetary**	**Other**
N	260	622	212	98	23	27	6

The training document contains 29 proceedings belonging to the local circumscriptions. These proceedings consist of a total amount of 1573 utterances. Furthermore, 2 speech records from the national diet consisting of a total of 363 speeches are also included in this file. This translates to a total of 1248 monetary expressions, which are our training samples. The class distribution of these samples is depicted in Table 1. The test document follows the same structure. A total of 760 utterances from local circumscriptions and 123 speeches from the national diet are included in this file. From all these transcriptions, 520 monetary expressions remain unlabelled in this document, which is the one used in the model evaluation of the BAM shared task.

4 MODEL ARCHITECTURE

We propose a BERT-based [6] cascade model to undertake the complete BAM process (see Figure 1). All the BERT-based classifiers integrated in our cascade model were fine-tuned from the *Inui Laboratory*[3] pre-trained BERT-large Japanese Language Model. In our approach, each monetary expression will be treated as the input and a class label and a related ID as the output. The proposed architecture aims to smooth the complexity of the classification task considering the size of the training corpus and the number of classes. Furthermore, it approaches independently the AC and the RID sub-tasks. Figure 2 synthesises the proposed architecture. The code implementation of the model architecture proposed in this paper is publicly available in GitHub[4].

Before tackling the AC and RID tasks, each monetary expression was analysed together with the discussion transcripts (i.e., local government utterances and national diet speeches) to produce segmented argument propositions. The segmentation was done by considering the set of full sentences belonging to the same utterance/speech where the monetary expression was contained. A set of handcrafted rules was applied during this pre-processing to determine if the proposition was either non monetary or other than a premise or a claim (e.g., detecting the existence of monetary Japanese kanji characters such as "円"). This way, the total number of remaining classes was reduced from seven to five.

Then, the AC is tackled by three different BERT-based classification models. A high-level BERT-based binary classifier was trained to detect if an argument proposition was either a premise or a claim. Once having assigned a high-level class to the sample, two low-level BERT-based classifiers were trained for 3-class premise classification and binary claim classification. This way, the high-level model focuses on the premise/claim discriminatory features, while the low-level focuses on more specific intra-class features. Furthermore, the class complexity of the problem is also decomposed from 5-class to 3-class and binary classifications.

[3]https://github.com/cl-tohoku/bert-japanese/tree/v2.0
[4]https://github.com/raruidol/Budget-AM

Finally, the RID is tackled by a BERT-based binary classifier, and a cosine similarity calculation for pairs of Sentence-BERT [21] embeddings. In this second part of the task, the segmented argument propositions containing monetary expressions are paired together with the *item* and *description* features of the budget items. A binary classifier is used to determine if a given pair (i.e., argument proposition, budget item) could be related or not. Then, all the pairs classified as related are scored using the cosine similarity of the dense embeddings of argument propositions (AP) and budget (B) items (i.e., *item+description* features) generated using a Sentence-BERT model. The highest scored relation is used in our approach to produce the model's output.

Therefore, each monetary expression was completely analysed by our cascade model, classified into one of the seven argumentative classes, and related to one of the budget items in the list.

5 RESULTS

The evaluation of the architecture proposed in this paper has been carried out at three different levels. First, we performed a local evaluation of the models aimed at having preliminary notions of how would our proposal behave with the test data. Second, we received feedback of our model's performance in an initial "*Dry-run*" phase of the BAM task. Finally, the "*Formal-run*" evaluation of the models corresponds to the last and definitive round of the shared task.

5.1 Experimental Setup

All the experiments and results reported in this paper have been implemented and run under the following setup. For the pre-processing of the corpus, we have used *pandas* [16] for handling data structuring, and *fugashi* [15] for the analysis of Japanese natural language text. For model training and transfer learning we have used the *PyTorch* and *Transformers* [28] libraries. This powerful deep learning tools have made possible to take advantage of existing large language models in Japanese, and adapt them to our specific task. For the semantic cosine similarity calculus in RID sub-task we used the *Sentence Transformers* [21] library. Finally, the local evaluation metrics (i.e., accuracy and macro f1) have been implemented using the *sklearn* library.

5.2 Local Evaluation

During the local evaluation, we have tested different model architectures. The most basic approach consisted of a 7-class BERT-based classification model (*7BERT*). We also experimented with a 5-class BERT-based classification model together with a set of handcrafted rules for the underrepresented classes (i.e., non monetary and other) (*5BERT*). Finally, we evaluated the BERT-based cascade model proposed in this work for tackling the BAM task (*rVRAIN*). Each of these models was also evaluated considering a balanced version of the corpus, where premise and claim training sample distributions were more balanced than the original corpus (*BD*). For the evaluation, we considered the accuracy and the Macro-F1 scores. This decision was made based on the strong unbalance between classes observed in the training corpus. Furthermore, we evaluated our models using a 10-fold cross validation. Table 2 summarises the obtained results during the local evaluation of our models.

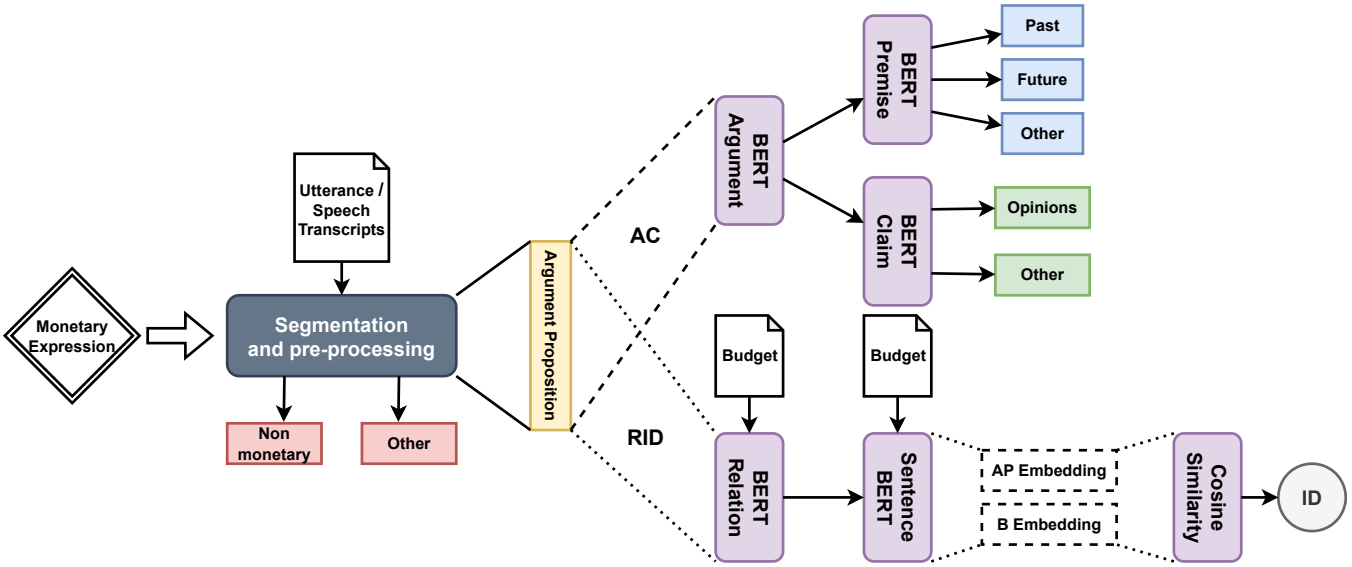

Figure 2: *rVRAIN* model architecture proposal.

Table 2: Local evaluation of the different models for AC.

Model	Accuracy	Macro-F1
7BERT	0.71	0.19
7BERT(BD)	0.56	0.16
5BERT	**0.76**	0.25
5BERT(BD)	0.55	0.19
rVRAIN	0.47	**0.27**
rVRAIN(BD)	0.42	0.22

Table 3: Dry-run (early) evaluation of the different models for BAM.

Team	Score AC+RID
fuys	0.51
rVRAIN (5BERT)	0.45
rVRAIN	0.40
OUC	0.33
rVRAIN (7BERT)	0.25
RB	0.09

We can observe how the best accuracy score was obtained by the *5BERT* model. However, *rVRAIN* achieved the best performance considering the Macro-F1 score. This means that our cascade model generalised better on this task, by doing a better classification of the samples belonging to underrepresented classes.

5.3 Dry-Run Evaluation

The Dry-Run evaluation phase of the BAM shared task used the test file to evaluate our submissions, and was divided into two different stages. During the early stage (see Table 3), the evaluation script assigned a unique score to the team submissions. This score combined the performance of the models in AC and RID sub-tasks. In the early stage, we evaluated the performance of the same models evaluated during the local evaluation, except for the balanced data versions. Our models achieved the 2nd and 3rd best scores for the BAM task. *RB* stands for the random baseline provided by the organisers of the task.

However, the evaluation script was updated the last month of the Dry-Run evaluation. The late stage (see Table 4) of the Dry-Run evaluation provided individual scores for the AC and RID tasks, together with a global evaluation of the performance of the model in the BAM task. Aimed at easing the readability of the results, we will only include the best performing approaches of each team in the tables. Our best performing model in the late stage was the one using the *5BERT* model for AC, but we could not evaluate the cascade architecture during this phase. Furthermore, the new evaluation script only considered those samples with both, the argument class and the relation ID correctly predicted, to increase the global score of the BAM task. This explains why our approach was the 3rd ranked with the best general score, even though it was the 2nd in the AC and the 1st in the RID sub-tasks.

5.4 Formal-Run Evaluation

During the Formal-Run evaluation, the same test file than with the Dry-Run was used. We achieved our best results using the proposed cascade model architecture for AC, together with the proposed semantic similarity calculation method for RID. As presented in Table 5, the *rVRAIN* achieved the 4th best performing position from

Table 4: Dry-run (late) evaluation of the different models for BAM.

Team	Score AC+RID	AC	RID
fuys	**0.13**	**0.57**	0.17
OUC	**0.13**	0.37	**0.21**
rVRAIN (5BERT)	0.06	0.48	**0.21**
takelab	0.00	0.33	0.00
RB	0.00	0.13	0.00

Table 5: Formal-run evaluation of the different models for BAM. (*)The team contains task organisers.

Team	Score AC+RID	AC	RID
*JRIRD**	**0.51**	**0.58**	0.61
*OUC**	0.45	0.57	**0.66**
*fuys**	0.23	0.57	0.34
rVRAIN	0.17	0.48	0.21
rVRAIN (5BERT)	0.06	0.48	0.21
takelab	0.04	0.39	0.06
SMLAB	0.00	0.38	0.00
RB	0.00	0.13	0.00

a total of 6 participating teams. However, our approach was the best performing one from the teams that did not include task organisers.

6 DISCUSSION

We have described the participation of *rVRAIN*'s team at the *Budget Argument Mining* task organised in the *QALab PoliInfo 3* and the *NTCIR-16*. The organisers proposed a new instance of the argument mining task, a classic in the NLP area of research. In this new instance, the main goal was to correctly classify arguments containing monetary expressions and relate them to items in a list of political budgets. In this paper, we have proposed a new approach to this task relying in the latest advances in NLP (i.e., Transformer-based architectures). The proposed cascade model architecture achieved the fourth position in the performance ranking, and it was the best among teams without task organisers.

Several observations can be drawn from this paper's proposal and experimentation. First, we have seen how when dealing with highly unbalanced corpora, a system can benefit from defining a set of handcrafted rules and relaxing the class complexity of the task. Instead of approaching the complete problem with the use of a unique classifier. Second, we have also observed that no improvement could be achieved by forcing the balance of the corpus. When using the balanced version, the score dropped significantly. This is most probably because the real distribution that the model has to predict is not balanced, but the corpus size limitation can also have a major role in this issue.

Finally, we foresee the implementation of some communication between the models for AC and RID during their training as a future work improvement of the model's performance on the BAM task. Furthermore, we also consider that using different test sets for each phase of the shared task (i.e., Dry-Run and Formal-Run) would be beneficial for the generalisation of the findings in this topic.

ACKNOWLEDGMENTS

This work is supported by the Spanish Government project PID2020-113416RB-I00.

REFERENCES

[1] Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. 2017. Unit segmentation of argumentative texts. In *Proceedings of the 4th Workshop on Argument Mining*. 118–128.

[2] Khalid Al Khatib, Henning Wachsmuth, Matthias Hagen, Jonas Köhler, and Benno Stein. 2016. Cross-domain mining of argumentative text through distant supervision. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: human language technologies*. 1395–1404.

[3] Jianzhu Bao, Chuang Fan, Jipeng Wu, Yixue Dang, Jiachen Du, and Ruifeng Xu. 2021. A Neural Transition-based Model for Argumentation Mining. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 6354–6364.

[4] Jianzhu Bao, Bin Liang, Jingyi Sun, Yice Zhang, Min Yang, and Ruifeng Xu. 2021. Argument Pair Extraction with Mutual Guidance and Inter-sentence Relation Graph. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 3923–3934.

[5] Oana Cocarascu and Francesca Toni. 2017. Identifying attack and support argumentative relations using deep learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 1374–1379.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[7] Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural end-to-end learning for computational argumentation mining. *arXiv preprint arXiv:1704.06104* (2017).

[8] Ivan Habernal and Iryna Gurevych. 2017. Argumentation mining in user-generated web discourse. *Computational Linguistics* 43, 1 (2017), 125–179.

[9] Yufang Hou and Charles Jochim. 2017. Argument relation classification using a joint inference model. In *Proceedings of the 4th Workshop on Argument Mining*. 60–66.

[10] Yohan Jo, Jacky Visser, Chris Reed, and Eduard Hovy. 2019. A cascade model for proposition extraction in argumentation. In *Proceedings of the 6th Workshop on Argument Mining*. Association for Computational Linguistics, 11–24.

[11] 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. 2022. Overview of the NTCIR-16 QA Lab-PoliInfo-3 Task. *Proceedings of The 16th NTCIR Conference* (6 2022).

[12] John Lawrence and Chris Reed. 2020. Argument mining: A survey. *Computational Linguistics* 45, 4 (2020), 765–818.

[13] Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. 2014. Context dependent claim detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 1489–1500.

[14] Tobias Mayer, Elena Cabrio, Marco Lippi, Paolo Torroni, and Serena Villata. 2018. Argument Mining on Clinical Trials.. In *COMMA*. 137–148.

[15] Paul McCann. 2020. fugashi, a Tool for Tokenizing Japanese in Python. *arXiv preprint arXiv:2010.06858* (2020).

[16] Wes McKinney et al. 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, Vol. 445. Austin, TX, 51–56.

[17] Gaku Morio and Katsuhide Fujita. 2018. End-to-end argument mining for discussion threads based on parallel constrained pointer architecture. *arXiv preprint arXiv:1809.00563* (2018).

[18] Gaku Morio and Katsuhide Fujita. 2019. Syntactic graph convolution in multi-task learning for identifying and classifying the argument component. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*. IEEE, 271–278.

[19] Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th international conference on artificial intelligence and law*. 98–107.

[20] Prakash Poudyal, Jaromír Šavelka, Aagje Ieven, Marie Francine Moens, Teresa Gonçalves, and Paulo Quaresma. 2020. ECHR: legal corpus for argument mining. In *Proceedings of the 7th Workshop on Argument Mining*. 67–75.

[21] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[22] Ramon Ruiz-Dolz, Jose Alemany, Stella Heras, and Ana Garcia-Fornes. 2021. Transformer-based models for automatic identification of argument relations: A cross-domain evaluation. *IEEE Intelligent Systems* (2021).

[23] Ramon Ruiz-Dolz, Montserrat Nofre, Mariona Taulé, Stella Heras, and Ana García-Fornes. 2021. VivesDebate: A New Annotated Multilingual Corpus of Argumentation in a Debate Tournament. *Applied Sciences* 11, 15 (2021), 7160.

[24] Noam Slonim, Yonatan Bilu, Carlos Alzate, Roy Bar-Haim, Ben Bogin, Francesca Bonin, Leshem Choshen, Edo Cohen-Karlik, Lena Dankin, Lilach Edelstein, et al. 2021. An autonomous debating system. *Nature* 591, 7850 (2021), 379–384.

[25] Christian Stab and Iryna Gurevych. 2014. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 46–56.

[26] Christian Stab, Tristan Miller, and Iryna Gurevych. 2018. Cross-topic argument mining from heterogeneous sources using attention-based neural networks. *arXiv preprint arXiv:1802.05758* (2018).

[27] S Villata et al. 2020. Using Argument Mining for Legal Text Summarization. In *Legal Knowledge and Information Systems: JURIX 2020: The Thirty-third Annual Conference, Brno, Czech Republic, December 9-11, 2020*, Vol. 334. IOS Press, 184.

[28] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771* (2019).