# John-Arthur at SemEval-2023 Task 4: Fine-Tuning Large Language Models for Arguments Classification

Georgios Balikas Salesforce.com geompalik@hotmail.com

#### Abstract

This paper presents the system submissions of the John-Arthur team to the SemEval Task 4 "ValueEval: Identification of Human Values behind Arguments". The best system of the team was ranked 3rd and the overall rank of the team was 2nd (the first team had the two best systems). John-Arthur team models the ValueEval problem as a multi-class, multi-label text classification problem. The solutions leverage recently proposed large language models that are fine-tuned on the provided datasets. To boost the achieved performance we employ different best practises whose impact on the model performance we evaluate here. The code is publicly available at github and the model on Huggingface hub.

#### 1 Introduction

SemEval Task 4 "ValueEval: Identification of Human Values behind Arguments" has as a goal to classify a textual argument across one or more human value categories (Kiesel et al., 2022, 2023). Understanding and identifying human values in arguments is a difficult task as they are implicit and their definitions are often vague. At the same time they are studied in various domains like social studies and formal argumentation. As a result, having a system aiding to the task could have a measurable impact on such studies.

To build an NLP system that classifies input text data among human values we build on recent work on large language models (LLMs). These models are capable of "understanding" English text and our hope is that they can be used to achieve satisfactory performance on the task. The ValueEval input data consist of three pieces of information:

- The argument's stance: which is "in favour" or "against"
- The argument's premise, and
- The argument's conclusion.

On the other hand, the prediction step is to recognize the human value for each input among 20 values. This is a multi-class, multi-label classification problem as most inputs belong to several human values.

In our experiments we discovered that LLMs are a suitable solution to the task. Properly encoding the input information for the LLMs and fine-tuning the models with the data provided by the organizers quickly improves the performance. Also, we found that bigger models achieve better results if tuned properly. Adding data to the training set also has a big impact on the obtained performance. We tried various ideas in the prototyping phase: from different ways to encode the model input to various architecture choices on how to use the model output. Compared to more traditional machine modeling lifecycle where a lot of effort is put on feature engineering, we often realized that while some feature engineering benefits the performance using smaller models, its impact on larger model is diminishing.

The best system of the John-Arthur team is currently ranked at position 3 of the private leaderboard. In terms of teams, John-Arthur is ranked 2nd, behind the Adam Smith team, who submitted the 2 best performing systems. We make the code that trains a model publicly available<sup>1</sup> and the model of the best submission also available.<sup>2</sup>

## 2 Background

Task 4 of SemEval 2023 consists of a single track. Task submissions were handled using the TIRA platform (Fröbe et al., 2023). The dataset of the challenge is described in (Mirzakhmedova et al., 2023). An example input to be classified is as follows:

<sup>&</sup>lt;sup>1</sup>https://github.com/balikasg/ SemEval2023-Task4-John-Arthur <sup>2</sup>https://burgingfore.or/halikas

<sup>&</sup>lt;sup>2</sup>https://huggingface.co/balikasg/ SemEval2023Task4

- Stance: in favour of
- Premise: We should ban human cloning as it will only cause huge issues when you have a bunch of the same humans running around all acting the same.
- Conclusion: We should ban human cloning.
- Target: "Security: societal" (among 20 classes)

This is framed as a multi-label, multi-class text classification problem. The evaluation measure the organizers chose is the macro-averaged  $F_1$  score.

## 3 System Overview

We model the problem as a text classification problem and we intend to use LLMs on the premise that they can uncover and model the implicit semantics of the input data to be able to successfully predict the human values.

## 3.1 Input Encoding

The first decision we are faced with is on how to encode the input data. We employ the notion of a cross-encoder system where in its inputs different information pieces are joined using separators. For our use-case we encode the input as:

> input = Stance + separator +Premise + separator + Conclusion

where "+" refers to the text concatenation operation. We experimented with different ways of this encoding. In particular, we concluded that using the model's separator token consistently performed better compared to other (new) separator tokens. We also found that for the low-cardinality values of Stance (in favour of vs against) it is beneficial to use separate tokenizer symbols to model them. As a result, we used '[Favour]' and '[Against]' to model them. This gave a small lift across the models we tried. As a conclusion, the input encoding of the best performing model is for the example of Section 2 is "[Favour][SEP]We should ban .. the same[SEP]We should ban human cloning[SEP]" where the actual text is lower-cased.

## 3.2 Model Selection

In the preliminary phase of our solution development we identified the families of Roberta (Liu

et al., 2019) and deberta (He et al., 2021) models to be promising model architectures. As a result, we iterated on these models and tried different hyper-parameters on batch-size and learning rates to verify the convergence and how the model would fit in a GPU. To fit the models in the Google Colab GPUs we used a batch size of 16 and a learning rate of 2e - 5 for the base models. Moving to smaller or bigger models, we used a rule of thumb and adjusted the batch size (dividing or multiplying by 2) and doing the inverse operation for the learning rate. Towards the submission deadline of the competition, we rented a Google Cloud virtual machine with a A100 GPU to be able to submit a run with the biggest deberta model available: "microsoft/deberta-v2-xxlarge"<sup>3</sup>. The code heavily relies on the popular Huggingface Transformers python package (Wolf et al., 2019). Table ?? shows the results of fine-tuning different model architectures for 6 epochs with a batch size of 4, a learning rate of 0.5e - 05 where the training data are the "arguments-training.csv" the task organizers provided and the F1 score is reported on the "arguments-validation.csv" dataset. From the Table we observe that "microsoft/deberta-v2-xxlarge" clearly outperforms the rest of the models and this is why we ended up using it in the final submissions.

While we experimented with several modeling choices we found that in the xxlarge model using the model's output ([CLS] token) without any postprocessing or extra hidden layer performed the best according to the validation metrics. We particularly experimented with what type of model output to use on the classification head and tried among different choices between using:

- the [CLS] token representation
- mean or max or concatenation of mean and max pooling of the the other token representations
- concatenate pooling of not only the last but the last 3 output layers of the model with or without weights

but while some of these worked on small Roberta models, they did not on the xxlarge deberta model.<sup>4</sup>

<sup>3</sup>https://huggingface.co/microsoft/ deberta-v2-xxlarge

<sup>&</sup>lt;sup>4</sup>A description of such techniques is available at https://www.kaggle.com/code/rhtsingh/ utilizing-transformer-representations-efficiently.

Test set / Approach	All	Self-direction: thought	Self-direction: action	Stimulation	Hedonism	Achievement	Power: dominance	Power: resources	Face	Security: personal	Security: societal	Tradition	Conformity: rules	Conformity: interpersonal	Humility	Benevolence: caring	Benevolence: dependability	Universalism: concern	Universalism: nature	Universalism: tolerance	Universalism: objectivity
Main																					
Best per category	.59	.61	.71	.39	.39	.66	.50	.57	.39	.80	.68	.65	.61	.69	.39	.60	.43	.78	.87	.46	.58
Best approach	.56	.57	.71	.32	.25	.66	.47	.53	.38	.76	.64	.63	.60	.65	.32	.57	.43	.73	.82	.46	.52
BERT	.42	.44	.55	.05	.20	.56	.29	.44	.13	.74	.59	.43	.47	.23	.07	.46	.14	.67	.71	.32	.33
1-Baseline	.26	.17	.40	.09	.03	.41	.13	.12	.12	.51	.40	.19	.31	.07	.09	.35	.19	.54	.17	.22	.46
xxlarge deberta xxlarge deberta (full data)	.53 .55	.51 .56	.69 .70	.28 .27	.20 .25	.63 .65	.43 .50	.55 .52	.37 .39	.71 .76	.58 .60	.62 .63	.55 .60	.50 .69	.17 .24	.53 .55	.41 .41	.72 .74	.87 .86	.43 .44	.57 .58

Table 1: Achieved  $F_1$ -score of team John-Arthur per test dataset, from macro-precision and macro-recall (All) and for each of the 20 value categories. Approaches marked with \* were not part of the official evaluation. Approaches in gray are shown for comparison: an ensemble using the best participant approach for each individual category; the best participant approach; and the organizer's BERT and 1-Baseline.

Model Name	$F_1$ score
microsoft/deberta-v3-small	0.3563
microsoft/deberta-v3-base	0.3776
roberta-base	0.4260
microsoft/deberta-v3-large	0.4585
roberta-large	0.501
microsoft/deberta-v2-xlarge	0.5178
microsoft/deberta-v2-xxlarge	0.5268

Table 2: Macro- $F_1$  score on the validation set depending on the model architecture that is fine-tuned. All models are fine-tuned for 6 epochs with a batch size of 4, a learning rate of 0.5e - 05. For the predictions, a threshold of 0.2 is used.

To cope with the multi-label aspect of the problem we used a binary cross-entropy loss that is applied on the model logits.<sup>5</sup>

#### 3.3 Model Output Post-processing

We pass the model outputs from a sigmoid function is order to get float values in [0,1] to resemble class probabilities. The task, in the predictions file requires a binary prediction for each human value, where 1 indicates that the input example belongs to this human value. To go from a float value  $p \in [0,1]$  to a binary output a threshold is required. Typically one chooses a threshold of 0.5 but this is a parameter that can be tuned on the validation set. In our submissions we used a threshold of 0.20. The macro-averaged  $F_1$  metric was implemented as a callback on the Trainer class using scikit-learn (Pedregosa et al., 2011).

While we chose to use a global threshold i.e., a single value across all classes, for the submissions, the is some room for improvement. One can come up with a strategy where each class has a different threshold that is learned on the validation data. In fact, in early experiments we validated the effectiveness of this approach as a promising post-processing trick, but in the end we did not implement it. We leave this as future work.

#### 4 Experimental Setup

The organizers released several data splits. The main dataset comprised 3 splits (train, test, validation) that consist of 5,393, 1,897 and 1,576 data points respectively. Also, 100 more labeled data points were released as "arguments-validation-zhihu.tsv" from related work. In early iterations we found that the model greatly improved with more training data. We discovered this by observing a lift across all model architectures when adding as few as 100 extra data points in the training data ("arguments-validation-zhihu.tsv") which were few compared to the 5,393 data points of the training dataset.

Motivated by this observation that our systems benefits a lot from extra training data, we decided to submit two runs with the same modeling

<sup>&</sup>lt;sup>5</sup>More information on how to do multilabel classification with Huggingface on this thread https://discuss.huggingface.co/t/ fine-tune-for-multiclass-or-multilabel-multiclass/ 4035?page=2

Team Name	Top-Score
Adam-Smith	0.561
John-Arthur	0.553
Theodor-winger	0.538
mao-zedong	0.533
confucius	0.531
BERT	0.420

Table 3: Top-5 teams of SemEval 2022 Task 4. BERT was the strong baseline provided by the Task organizers. The best performing system beat this baseline by a very large margin, for example the John-Arthur system achieves a score of 13.2 absolute points more (0.553 vs 0.42).

and training configurations. The first run uses as training data the training data and the "arguments-validation-zhihu.tsv" dataset while the second run uses as training data the original training and validation datasets (5,393 + 1,897 data points) and as validation the 100 points of "arguments-validation-zhihu.tsv". We opted for having a validation set because we wanted to adhere to best neural network modeling practises and monitor the evaluation loss and the evaluation metrics to ensure that the model trains properly. We leave using the concatenation of all available data for training as well as performing some form of data augmentation or distant supervision using unlabeled data as promising directions of future work.

### **5** Results

Table 1 summarizes the results of the 2 runs. Recall that the main difference between these runs is that the second includes more training data and a smaller evaluation dataset. There are several classes where the systems of John-Arthur achieved the best performance obtained in the task. We did not perform further error analysis to evaluate how much these results can be improved using, for example, a different threshold per class or even different models per class. We leave this for future work.

Table 3 shows the best system scores of the top-5 team. John-Arthur team is ranked 2nd which speaks to the high performance that LLMs can achieve when carefully tuned on text analysis tasks. What is more, these top-ranked system beat the string baseline (BERT) provided by the competition organizers by a very large margin (often >10 absolute points of the macro-averaged  $F_1$  measure) which we believe is a promising and valuable outcome for the study of human values.

#### 6 Conclusion

In this paper we presented the submission of the John-Arthur team in Task 4 of SemEval 2023. The best submission of the team is ranked in position 3 of the leaderboard while the team itself is ranked 2nd. The core of the solution is an deberta-v2-xxlarge fine-tuned model with some pre-processing and post-processing custom operations. The code of the best performing system of the team are open-sourced with the hope that it can benefit the community<sup>6</sup>.

## 7 Acknowledgments

We would like to thank the machine learning community and Huggingface for developing the amazing open-source "transformers" library that powered our systems. Our systems also benefit from the numerous system discussions and code resources in Kaggle, and notably Jeremy Howard's "Iterate Like a Grandmaster" notebook that inspired our solution, who we would like to thank for making available such high quality content. Last, we would like to thank Google Colab for making GPU resources available for free in a notebooking environment.

#### References

- Maik Fröbe, Matti Wiegmann, Nikolay Kolyada, Bastian Grahm, Theresa Elstner, Frank Loebe, Matthias Hagen, Benno Stein, and Martin Potthast. 2023. Continuous Integration for Reproducible Shared Tasks with TIRA.io. In Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Berlin Heidelberg New York. Springer.
- Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.
- Johannes Kiesel, Milad Alshomary, Nicolas Handke, Xiaoni Cai, Henning Wachsmuth, and Benno Stein. 2022. Identifying the Human Values behind Arguments. In 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022), pages 4459– 4471. Association for Computational Linguistics.
- Johannes Kiesel, Milad Alshomary, Nailia Mirzakhmedova, Maximilian Heinrich, Nicolas Handke, Henning Wachsmuth, and Benno Stein. 2023. Semeval-2023 task 4: Valueeval: Identification of human values behind arguments. In *Proceedings of the* 17th International Workshop on Semantic Evaluation,

<sup>&</sup>lt;sup>6</sup>https://github.com/balikasg/ SemEval2023-Task4-John-Arthur

Toronto, Canada. Association for Computational Linguistics.

- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.
- Nailia Mirzakhmedova, Johannes Kiesel, Milad Alshomary, Maximilian Heinrich, Nicolas Handke, Xiaoni Cai, Barriere Valentin, Doratossadat Dastgheib, Omid Ghahroodi, Mohammad Ali Sadraei, Ehsaneddin Asgari, Lea Kawaletz, Henning Wachsmuth, and Benno Stein. 2023. The Touché23-ValueEval Dataset for Identifying Human Values behind Arguments. *CoRR*, abs/2301.13771.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- 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-ofthe-art natural language processing. *arXiv preprint arXiv:1910.03771*.