Aristoxenus at SemEval-2023 Task 4: A Domain-Adapted Ensemble Approach to the Identification of Human Values behind Arguments

Dimitrios Zaikis
School of Informatics
Aristotle University of Thessaloniki
dimitriz@csd.auth.gr

Stefanos D. Stefanidis
Independent researcher
stdistefanidis@gmail.com

Konstantinos Anagnostopoulos
Independent researcher
anag.konst@gmail.com

Ioannis Vlahavas
School of Informatics
Aristotle University of Thessaloniki
vlahavas@csd.auth.gr

Abstract

This paper presents our system for the SemEval-2023 Task 4, which aims to identify human values behind arguments by classifying whether or not an argument draws on a specific category. Our approach leverages a second-phase pre-training method to adapt a RoBERTa Language Model (LM) and tackles the problem using a One-Versus-All strategy. Final predictions are determined by a majority voting module that combines the outputs of an ensemble of three sets of per-label models. We conducted experiments to evaluate the impact of different pre-trained LMs on the task, comparing their performance in both pre-trained and task-adapted settings. Our findings show that fine-tuning the RoBERTa LM on the task-specific dataset improves its performance, outperforming the best-performing baseline BERT approach. Overall, our approach achieved a macro-F1 score of 0.47 on the official test set, demonstrating its potential in identifying human values behind arguments.

1 Introduction

The ValueEval task aims to develop a system for automatically detecting the values expressed in natural language arguments within English texts (Kiesel et al., 2023). Identifying human values is critical for gaining insights into people's behavior, evaluating content, personalizing experiences, and resolving conflicts. Analyzing the values expressed in language, including beliefs, attitudes, and motivations, can help us understand the quality and relevance of content and its potential impact (Kiesel et al., 2022). Moreover, identifying individual values can be useful in conflict resolution by enabling us to comprehend the underlying beliefs and motivations of opposing viewpoints. This can facilitate finding common ground and working towards a resolution that is acceptable to all. Therefore, identifying human values has the potential to play a significant role in various fields, including psychology, sociology, marketing, and others dealing with human behavior and communication.

In this paper, we propose a Transformer-based Language Model (LM) system for the ValueEval task, which utilizes second-phase pre-training in an One-Versus-All (OVA) setting to identify the human values expressed in arguments. Our approach combines both data and algorithm adaptation concepts, whereby we second-phase pre-train an LM to better adapt to the domain and transform the data to better represent the task. To align with the nature of the task and the dataset (Mirzakhmedova et al., 2023), we implement a form of prompt engineering. This involves transforming the premise, stance, and conclusion inputs into a single sentence while replacing the stance with a predefined template. Moreover, we task-adapt the RoBERTa LM by aligning it with the masked language-modeling objective to predict the probability of each stance given an argument and conclusion. To improve the model's performance, we train multiple models for each label, based on different hyperparameters and versions of the dataset that are sampled differently. Finally, we use majority voting to form the final predictions.

In addition to the system description presented in this paper, we make the following observations based on our approach and experiments: Firstly, we observed that second-phase pre-training in the form of task-adaptation allows the underlying LM to better represent the task in the embedding space. This leads to improved performance in identifying the human values expressed in arguments. Secondly, we found that utilizing an OVA approach, also known as One-Versus-Rest, dramatically improves performance compared to using a single multi-label classifier. Finally, we observed that the

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1037–1043
July 13-14, 2023 ©2023 Association for Computational Linguistics

effectiveness of data sampling techniques varied per label, with a subset of per-label models performing better without it. This highlights the importance of experimenting with different techniques to find the optimal approach for each label.

2 Background

While human values have long been an important consideration in formal argumentation, this task represents the first attempt to computationally identify the values behind arguments. To that end, Kiesel et al. (2022) presented the first dataset containing the conclusion, the premise's stance towards the conclusion, and the premise itself, as show in the example in Table 1.

Argument ID	A01010
Conclusion	We should prohibit school prayer
Stance	against
Premise	it should be allowed if the student wants to pray as long as it is not interfering with his classes

Table 1: Example that includes the conclusion, stance and premise of an argument.

The task involves determining whether a given textual argument relates to a specific category from a set of 20 value categories of human values derived from the social science literature. The baseline approaches for this multi-label classification problem include a "1-Baseline" where the positive label is assigned to all instances, a label-wise "SVM" and a Transformer-based approach, called "BERT".

One of the main advantages of Transformer-based LM approaches is their ability to capture complex linguistic structures and dependencies, which can be difficult to model using traditional approaches. In general, LM models can learn to understand context, ambiguity, and figurative language, which are all important aspects in argumentation mining, which is reflected by the published results as well, where "BERT" significantly outperforms the other approaches. This approach utilizes the BERT language model (Devlin et al., 2018) that uses stacked Transformer-based encoders (Vaswani et al., 2017), pre-trained on a large corpus of text data.

RoBERTa (Liu et al., 2019) is a variation of the BERT LM that was designed to improve upon some of its limitations, using a similar architecture, pre-trained on a larger and more diverse corpus of text data, with longer sequences and fewer masking tasks. This approach is intended to help RoBERTa capture more complex linguistic patterns and relationships than BERT. RoBERTa also uses a different pre-training objective, which involves training the model to predict the correct order of sentences in a document. This allows the model to better understand the relationships between different sentences in a document and to capture a wider range of linguistic knowledge.

By fine-tuning the pre-trained RoBERTa model on a specific task, the model can be optimized to better handle the specific requirements of that task and can result in improved performance. Accordingly, second-phase pre-training (Gururangan et al., 2020) can further improve an LM's performance with domain or task-adaptive pre-training that allows the model to learn task-specific features and patterns that are not captured by the general language model. The transfer of knowledge (transfer learning) allows a pre-trained LM to adapt to a specific task with less labeled training data and build upon the wide range of linguistic patterns and relationships previously learned.

3 System Overview

In this section, we describe our proposed Transformer-based system for the identification of human values behind argument in detail, where, given a conclusion, a stance and a premise, the input is classified into one of the 20 pre-defined values categories. Our system consists of an LM adaption (TAPT pre-train), data transformation, OVA training and tuning phase, as shown in Figure 1.

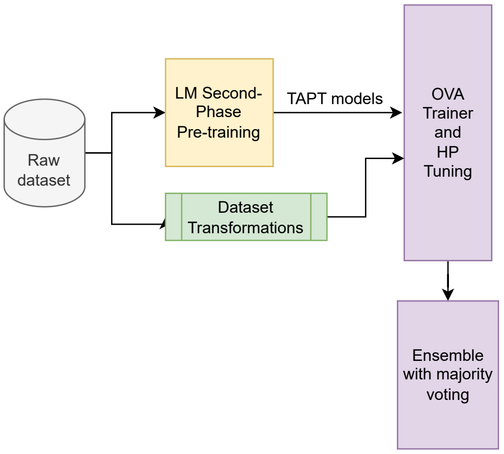
Raw dataset
LM Second-Phase Pre-training
TAPT models
Dataset Transformations
OVA Trainer and HP Tuning
Ensemble with majority voting

Figure 1: Overview of our proposed ensemble system with second-phase pre-training (task-adaption) and majority voting.

3.1 Language model alignment (Task adaption

Task adaptation refers to the process of second phase pre-training a LM with domain-specific unlabeled data that can potentially lead to performance improvements in that specific topic or domain (Gururangan et al., 2020). Towards that end, we implemented a slightly different approach where instead of aligning the LM on the task dataset, we trained the underlying language model on a different task. BERT-based models are trained using two types of sentences, Sentence A and Sentence B with the first being a sentence of a given input sequence, while the latter is a second sentence. This approach is commonly used in question answering or text classification tasks, where the input consists of a pair of sentences, where sentence A is a question or a prompt, and sentence B is the text to be classified or used to answer the question. In some cases, Sentence B might be the next sentence that follows Sentence A, but in other cases, it might be a sentence that is randomly chosen from the same document as Sentence A.

During the pre-training phase, BERT-based models are trained to learn a joint representation of both Sentence A and Sentence B using either Masked Language Modeling (MLM) or Next Sentence Prediction (NSP) tasks. In the MLM task, the model is trained to predict the masked words in Sentence A, while in the NSP task, the model is trained to predict whether Sentence B follows Sentence A in the original text or not.

We followed a supervised training approach to task adaption by training the model for the classification of the stance using the premise as Sentence A and the conclusion as Sentence B. Based on the argument presented in Table 1, the premise, “It should be allowed if the student wants to pray as long as it is not interfering with his classes” is used as Sentence A, while the conclusion, “We should prohibit school prayer” is used as Sentence B. By training the LM on this type of input, it can learn to classify the stance of a given text based on the relationship between the premise and conclusion. In this example, the model should predict that the stance is against school prayer since the premise argues for allowing it, while the conclusion argues for prohibiting it. By training on both Sentence A and Sentence B, BERT-based models can learn to understand the relationship between different sentences in a given text and capture the contextual meaning of the input sequence.

3.2 Per Label Task adapted models

Our proposed system is built on the task-adapted RoBERTa base language model, where we train separate binary classification models for each label. We use an OVA approach, where each model is trained to differentiate between instances of one class and instances of all other classes combined, effectively identifying instances of its corresponding label, while ignoring instances of all other labels.

To form the final input sentence string, we implement prompt engineering by concatenating the conclusion C to the premise P, using a connecting phrase that reflects the stance St and connects the premise and conclusion. Specifically, we use the format $S = C + R_{St} + P$, where R_{St} represents the appropriate connecting phrase. This process enables the model to take into account the relationship between the premise and conclusion, and the stance expressed in the connecting phrase.

After forming the input sentence, it is passed through the model’s transformer layers, which takes in the tokenized sentence and the attention mask. The resulting output is then processed using either mean pooling or the model’s pooled output and passed through a classification layer to produce the final per label binary output prediction.

3.3 Ensemble Module with Majority Voting

Given the total number of labels $N_L = 20$, we trained and tuned two distinct models for each label, resulting in a total of 40 models. We then grouped these models into three sets of OVA classifiers. Each set followed the architecture described in Section 3.2, and was individually hyperparameter-tuned, but trained on a different subset of the task dataset. One set was trained on the original dataset without any sampling, one set was trained on a down-sampled dataset, and the final set consisted of the best-performing models from the first two sets, which could either be non-sampled or down-sampled models.

During inference, we used each binary classifier corresponding to each class to predict the probability that a given sample belongs to that class. To generate the final prediction, we employed a majority voting approach. Specifically, we assigned a binary label based on a fine-tuned threshold, and the final label was determined by the majority vote among the three sets of models.

4 Experimental Setup

4.1 Dataset and Evaluation Methodology

We transformed the dataset for two different approaches by replacing the stance with a connecting phrase and by balancing the dataset by down-sampling to the majority label instances for a per label (OVA) approach. By replacing the stance from a set of phrases, as shown in Table 2, we concatenate conclusion and premise sequences with a randomly selected connecting phrase. For example, the argument presented in Table 1 would become “We should prohibit school prayer *so it is wrong to say that* it should be allowed if the student wants to pray as long as it is not interfering with his classes”. This process allows the model to learn from a continuous context and learn semantically relevant representations that take the complete argument into account. On the other hand, transforming a dataset for OVA involves converting the original multi-class labels into binary labels to create a set of binary labeled datasets, each corresponding to a single class.

Label	Phrases
against	“so it is not valid to say that” “so it is wrong that”
in favor of	“so” “thus” “therefore” “. Subsequently” “. As a result” “. So it is valid to say that” “, so it is true that”

Table 2: Pool of phrases that would connect the conclusion with the premise depending on the label.

The dataset had pre-defined train, validation and test splits, with labels for both the training and development sets. We created a development set from the train dataset by splitting it into 80% for training and 20% for development and used the provided validation set as test set. We follow the tasks evaluation strategy using the label-wise F1-score and its means over all labels (macro-averaged F1), which is the harmonic mean of the Precision and Recall metrics, applying the same weight to all classes.

4.2 Training

We trained each model on a single label and tuned the hyperparameters using the Optuna library (Akiba et al., 2019) using the search space as shown in Table 3 and trained for 100 epochs with an early stopping patience of 20. The best hyperparameters for each per label model are shown in the Appendix A.1 (Table 6).

Parameter	Search space
dropout	0.2 ... 0.25
learning_rate	1e-6 ... 1e-5
weight_decay	1e-4 ... 1e-3
warmup_steps	0.5, 1 ... 10 epoch(s)
batch_size	160, 192, 224, 256
max_norm	1.0, 2.0, 3.0
threshold	0.3 ... 0.45

Table 3: Hyperparameter search space for each binary classification model.

We trained the models without sampling and with down-sampling to create the two sets of OVA classifiers and used the best performing model per label as to create the third set. Additionally, we experimented with different pre-trained language models, such as BERT (Devlin et al., 2018), AlBERT (Lan et al., 2019), MPnet (Yee et al., 2019), XLnet (Yang et al., 2019) and DistilBERT, DistilRoBERTa (Sanh et al., 2019), both base and the large variant of RoBERTa (Liu et al., 2019). We task-adapted these models and trained them on the downstream task using the best per label hyperparameters and compared their average performance on the labeled validation set.

We implemented our described system with the Python programming language (3.8.16) and the PyTorch (1.10.2) and Transformers (4.23.1) libraries on a single computer with a 24-core Intel CPU and two Nvidia RTX A6000 graphics cards. The code is available at: https://github.com/d1mitriz/aristoxenus-semeval23-task4/

5 Results

This section describes the overall results compared to the best approach and our experimental results that led to our proposed system.

5.1 Overall results

In the ValueEval task, 40 teams submitted a total of 182 entries, including those from the organizers. Table 4 shows the official results for our system, as well as the baselines (1-Baseline and BERT), the top-performing approach, and the best-performing systems for each category. Our system achieved a macro-F1 score of 0.47 on the official test set, outperforming both baselines and the BERT approach in 19 out of 20 value categories.

Despite our system's strong performance compared to the baselines, it fell short compared to the best approach. Nevertheless, our approach demonstrates that an ensemble approach utilizing different training regimens using an OVA strategy can improve over a single multi-label classifier.

5.2 Experimental results

Initially, we developed a single multi-label classifier that could predict all value categories using a single architecture and classification head. Our goal was to explore how the model could leverage the inter-dependencies between the categories and the nature of the data. However, this approach resulted in the lowest performance among our experiments.

Despite our initial expectations, we found that the single classifier struggled to capture the subtle differences between the value categories and the complex relationships between them. Additionally, the relatively large number of categories and the imbalanced distribution of the data made it challenging for the model to learn meaningful representations for each category. As a result, we decided to explore alternative approaches, such as using separate classifiers for each category and incorporating additional features to improve the model's performance.

To address the limitations of the single classifier approach, we decided to split the responsibility across 20 models, each focused on predicting a single label using an One-Versus-All strategy with a majority vote system. The foundation of our approach was the underlying LM that generated semantically and contextually relevant embeddings for the input data.

To determine the best LM for this task and setting, we experimented with various base and large versions and evaluated their *adaptation* capabilities on the evaluation set. Table 5 summarizes the results of the LM experiments. Overall, we found that the base version of RoBERTa achieved the best results in terms of macro-F1 score.

Furthermore, we observed that fine-tuning the LM to the task-specific data improved its performance, suggesting that the LM could effectively learn to represent the unique features and nuances of this task. These findings informed our final system, which incorporated a second phase pre-trained RoBERTa-based model fine-tuned on the ValueEval dataset and achieved competitive results in the task.

6 Conclusion

In this paper, we describe our Transformer-based Language Model system for the ValueEval task, which utilizes second-phase pre-training in an One-Versus-All (OVA) setting to identify the human values expressed in arguments. We task-adapt the RoBERTa LM to the domain by training the model to predict the stance that connects the conclusion to the premise. Furthermore, we transform the input data to better capture the semantic and contextual information in a continuous way, by replacing the stance with a connecting phrase. Our system predicts based on a majority vote from predictions by an ensemble of three different sets of per label models. We show that the task-adaption improves on the systems performance, indicating the language models can learn to generate better embeddings by aligning them to this task. A possible direction for future work would be to investigate the impact of different language models as well as data transformation techniques on the systems predictive capabilities.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2623–2631, New York, NY, USA. Association for Computing Machinery.

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*.

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.

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

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
Ours	.47	.58	.66	.09	.25	.58	.07	.50	.29	.75	.61	.56	.51	.52	.27	.49	.20	.76	.77	.34	.40

Table 4: Achieved F_1-score of team aristoxenus 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.

Language Model	**macro-F1**
BERT-base	0.5621
TAPT-BERT-base	0.5655
RoBERTa-base	0.5960
TAPT-RoBERTa-base	**0.6217**
RoBERTa-large	0.4789
TAPT-RoBERTa-large	0. 5798
DistilBERT-base	0.5661
TAPT-DistilBERT-base	0.5630
DistilRoBERTa-base	0.5681
TAPT-DistilRoBERTa-base	0.5707
XLnet-base	0.5534
TAPT-XLnet-base	0.5701
MPnet-base	0.5580
TAPT-MPnet-base	0.5749
AlBERT-base	0.5578
TAPT-AlBERT-base	0.5513

Table 5: Experimental results of the different pre-trained language models and their task-adapted counterparts. TAPT in the model names, denotes Task Adapted Pre-Trained and bolt the best result. All results are based on the averaged macro-F1 score on the validation dataset.

values behind arguments. In *Proceedings of the 17th International Workshop on Semantic Evaluation*, Toronto, Canada. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations.

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.

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.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*.

A Appendix

A.1 Hyperparameters

Table 6 shows the hyperparameters for each model and label based on an extensive hyperparameter search as described in Section 4.2.

Label	dropout	learning_rate	weight_decay	warmup_steps	batch_size	max_norm	threshold
Self-direction: thought	0.2346715251	1.7632e-06	1e-3	1	192	3.0	0.44
Self-direction: action	0.2392132739	1.9734e-06	1e-3	2	160	2.0	0.38
Stimulation	0.2472472153	2.75292e-05	1e-3	10	160	3.0	0.44
Hedonism	0.2721909443	2.4005e-06	1e-3	1	192	1.0	0.32
Achievement	0.2270782147	6.6925e-06	1e-3	1	224	1.0	0.41
Power: dominance	0.2560592491	1.53487e-05	1e-3	10	192	3.0	0.33
Power: resources	0.2208751188	4.6915e-06	1e-3	1	224	3.0	0.42
Face	0.2262817676	3.82328e-05	1e-3	1	192	2.0	0.35
Security: personal	0.2845563742	1.42097e-05	1e-3	2	160	3.0	0.33
Security: societal	0.2468968908	8.324e-06	1e-3	2	192	2.0	0.43
Tradition	0.2071445925	4.45584e-05	1e-3	2	224	2.0	0.45
Conformity: rules	0.2284257602	9.5028e-06	1e-3	2	160	3.0	0.32
Conformity: interpersonal	0.2013318981	4.63617e-05	1e-3	9	192	1.0	0.34
Humility	0.2905871324	1.28707e-05	1e-3	1	224	2.0	0.45
Benevolence: caring	0.2246546093	4.90387e-05	1e-3	2	192	3.0	0.42
Benevolence: dependability	0.2686675537	4.32361e-05	1e-3	2	192	2.0	0.33
Universalism: concern	0.243886994	2.56778e-05	1e-3	8	192	1.0	0.34
Universalism: nature	0.2628180128	4.96647e-05	1e-3	2	224	1.0	0.41
Universalism: tolerance	0.2228968031	4.3012e-06	1e-3	2	160	1.0	0.33
Universalism: objectivity	0.2252762583	8.3819e-06	1e-3	2	224	3.0	0.32

Table 6: Results of optimal hyperparamaters obtained from tuning with Optuna for each label and model.