Assignment2 : Human Value Detection

Reza Shatery, Fatemeh Bozorgi, Kankana Ghosh
Master's Degree in Artificial Intelligence, University of Bologna
{ reza.shatery, fatemeh.bozorgi, kankana.ghosh }@ studio.unibo.it

Abstract

This report addresses the Human Value Detection Challenge, where the objective is to classify, given a textual argument and a human value category, classify whether or not the argument draws on that category. Human values behind natural language arguments, such as to have *'freedom of thought'* or to be *'broad-minded'* are commonly accepted answers and logic to why something is desirable in the ethical sense and are thus essential both in real world argumentation and theoretical argumentation frameworks. Our goal is to perform automatic multi label classification using several neural models considering solely level 3 value categories. Through our experimentation, we achieved a maximum F1-score of 0.88 and an average of 0.77.

1 Introduction

The original paper (Kiesel et al., 2022) studies the human values behind natural language arguments. The authors introduced a comprehensive taxonomy comprising 54 values and curated a dataset of 5270 arguments from four geographical cultures, manually annotated for human values. They compared three approaches,BERT, SVM, and 1-Baseline with training/testing on 'Premise' arguments for category wise classification.

In line with their work, we considered only level 3 value categories and compared the classification over three models, Baselines: *Uniform* , *Majority* classifier and *BERT*. We extended their approach by adding three different variants of BERT:

BERT w/ C: a BERT-based classifier that receives an argument conclusion as input.

BERT w/ CP: adding argument premise as an additional input.

BERT w/ CPS: adding argument premise-to-conclusion stance as an additional input. We fine-tuned multi-label roberta-large with batch size 16 and learning rate 2^{-5} (5 epochs). Our experimentation primarily succeeded in enhancing the test macro average F1-score from 0.71 to 0.77 for level 3 categories. While the classification results across various variants did not exhibit substantial differences, we observed improved scores when classifying 'Premise' and 'Conclusion' arguments compared to using only 'Premise' as input.

2 System description

2.1 Baseline

The baseline model was designed as an original contribution for the project. The architecture utilizes scikit-learn's *DummyClassifier* for random and majority classification. The training process is conducted independently for each category using specified random seeds to ensure reproducibility. Trained models are assessed on test sets, calculating F1-scores for each category and macro-average F1-score.

2.2 BERT

The BERT model architecture is adapted from the original BERT paper. It employs the transformers library, specifically the Pre-trained BERT model for sequence classification. The model is trained using a custom training loop with tokenization and encoding handled by the transformers library.

Data preprocessing involves transforming raw text inputs into a format suitable for training applying tokenization and encoding. To handle different input variants, we create a 'text' column by concatenating specified input columns, where 'Stance' column is encoded to neumerical format and maps the data into a DatasetDict.

Model Training utilizes the Hugging Face *Trainer* with a custom loss computation. We create a *TrainingArguments* to access all the points of customization during training. The process includes tokenization, encoding, and optimization training the BERT model on the prepared dataset,incorporating custom configurations such as evaluation strategy, learning rate, and model saving. The model is evaluated periodically during training, and the best model is saved based on the macro-average F1 score.

Model Prediction involves tokenization, encoding, and using the custom *MultiLabelTrainer* for predictions. We classify arguments using the trained BERT model and evaluate on the test set calculating F1-scores for each category and macro-average F1-score. We include an option for error analysis such as precision-recall curves and confusion matrices, for better performance analysis.

While the BERT model leverages the architecture and functions from the original paper, significant adaptations were made to align it with the project's context, making it an heavily adapted, project-specific implementation.

3 Experimental setup and results

3.1 Baseline

The baseline model is trained for each category independently utilizing scikit-learn's *DummyClassifier* with for *random and majority* classification. It is used with default settings with three different random state (seed) to control the randomness. Trained models were saved in the specified model directory for use during prediction.

3.2 BERT

We utilized Hugging Face *AutoModelForSequenceClassification* with pre-trained *'roberta-large'*, with a custom *MultiLabelTrainer* class extending the Trainer class from the transformers library. It overrides the compute-loss method for custom loss computation using a combination of Binary Cross Entropy which is implemented as *BCEWithLogitsLoss* in PyTorch. We fine-tuned the model with *batch size 16, learning rate* 2^{-5} *(5 epochs) and weight Decay 0.01.* BERT input data was tokenized using the *AutoTokenizer* from HuggingFace. The trained BERT model was saved in the specified model directory and used *'macro-average F1-score'* for selecting the best model.

Overall a common structure was maintained for both experiments to facilitate easy comparison. F1-score was used for evaluation on each category, and the macro-average F1-score was calculated. The code supports training and evaluation for three different seeds, enabling a robust analysis of model performance.

Category	F1 Score					
	Uniform	Majority	BERT	w/CPS	w/CP	w/P
Openness to change	0.41	0.0	0.68		0.68	0.66
Self-enhancement	0.45	0.0	0.68		0.68	0.68
Conversation	0.58	0.81	0.83		0.83	0.83
Self-transcendence	0.68	0.89	0.88		0.88	0.89
avg-f1-score	0.52	0.43	0.77		0.77	0.76

Table 1: F1 Scores for different Models and variants

4 Discussion

Quantitative Results: Our experiments show significant progress in automatic multi-label classification for level 3 value categories. The baseline model, using random and majority classifiers, had an average F1-score of 0.52. In contrast, our BERT-based model improved this to 0.77, emphasizing the effectiveness of advanced transformer models in understanding human values from text. BERT consistently outperformed the baseline across all categories as mentioned in Table 1, the result of the test set. This indicates our model effectively captures diverse human values expressed in different arguments. The introduced BERT variants didn't significantly improve over the base BERT model.
Error Analysis: Our error analysis highlighted challenges in accurately classifying arguments related to 'Openness to change' and 'Self-enhancement,' where F1-scores were lower. These categories contained subtle nuances and context-specific expressions, challenging the model's generalization.

Future improvements could involve adding more context-aware features, exploring different pre-training techniques, or diversifying the dataset to better capture the intricacies of human values.

5 Conclusion

In summary, our project successfully tackled the Human Value Detection Challenge, showing a boost in F1-scores compared to basic models. The BERT-based method proved effective in categorizing level 3 values in text.

Our solution struggles with context-specific language, notably in 'Openness to change' and 'Self-enhancement.' To enhance our model, we could refine its architecture to better grasp context or broaden the dataset to include more linguistic variations. Despite these challenges, our work provides a foundation for further research in automating the interpretation of human values from natural language arguments.

References

Johannes Kiesel, Milad Alshomary, Nicolas Handke, Xiaoni Cai, Henning Wachsmuth, and Benno Stein. 2022. Identifying the human values behind arguments. in smaranda muresan, preslav nakov, and aline villavicencio, editors, 60th annual meeting of the association for computational linguistics (acl 2022). In *ACL (2022)*, page 4459–4471. Association for Computational Linguistics.