Information Retrieval Exam Project
SemEval task 4: Human Value Detection

Antonio Belotti
mat 960822

Università degli studi di Milano

Abstract. This paper presents various attempts at building a classifier able to detect human values in a textual input. Human values can be thought of as fundamental, underlying beliefs that move people to make a certain decision or idea, instead of another. A set of human values can be the foundation to express any concept or stance about any subject, and as such, detecting them from text can be difficult. The best-performing model produced with this work, a fine-tuned DeBerta v3 classifier, achieved an average macro F1 score of ≈ 0.49 on the test set.

1 Introduction

This paper describes my attempts at solving the SemEval 2023 Task 4 challenge: "identifying human values in text".[1] The authors of the challenge, also creators of the dataset, provided a classifier yielding a performance baseline. This project aims to build a classifier that outperforms that result.

A *human value* is a belief that influences how one forms an opinion about a topic, how one behaves, and how one thinks about others. Each person has their own set of beliefs, with different priorities, more important values override the effect of less important values. Some values are incompatible and conflict with other values. The formalization of human values is a complex task that pertains to the field of social studies. [3] describes human values, their hierarchical categorization, the history of the studies, and provides more references.

This paper is organized as follows. Section 2 describes the task and the available dataset. Section 3 describes the experiments, training, and evaluation of all produced models. Section 4 evaluates the results and concludes the paper.

The code for this work is publicly available on Git Hub. [2]

2 Automatic detection of human values

Identifying human values from text requires detecting hidden between-the-lines properties. Consider the example phrase "no matter they felt forced to commit it:

[1] ttps://touche.webis.de/semeval23/touche23-web/

[2] https://github.com/antoniobelotti/HVD

anyone who commits a crime should be prosecuted": there is no explicit mention of the pertaining human values (*behaving properly*, wanting a *safe country*), nor their priorities. One can try to infer this relation, using a labeled dataset to build an automatic classifier.

Dataset The Touché23-ValueEval Dataset [5], is a labeled dataset of 9324 arguments from six different sources. It is an expansion of the same authors' previous dataset [3]. Each sample is structured in 4 parts: stance, premise, conclusion, and labels. A complete example of one dataset row is provided in table 1. There are 20 possible human values, organized in a hierarchy (figure 1 in appendix A). Classification can be performed at various levels: I will focus on the fine-grained 20 categories, referred to as *level 1*.

I downloaded the dataset from the huggingface hub and used the *main* split.[3]

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."
Stance	"in favor of"
Conclusion	"We should ban human cloning"
Labels	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Table 1. Dataset sample

Considering the available data, the task at hand is a multi-label classification problem: each input text can be linked to multiple human values. The training data is imbalanced (label distribution in figure 2 appendix A).

Baseline performance [3, 5] provide the results of a baseline model: a *"fine-tuned multi-label bert-base-uncased with batch size 8 and learning rate* 2^{-5} *(20 epochs)"* achieved a macro F1 score of 0.26 on the test set. The rest of this paper will describe various attempts to build a better classifier.

3 Experiments

This section describes the challenges faced in the construction of a solution and the steps that lead to the definition of the final model. To answer a couple of questions about the correct methodology to follow, I trained some smaller models to guide the process.

[3] `https://huggingface.co/datasets/webis/Touche23-ValueEval`

Evaluation strategy The baseline model is evaluated using the macro F1 score on the test set. This metric is the unweighted average of the F1 score for each label. All the following experiments will be evaluated according to the same metric.

What data to use? Considering the provided dataset, the most important part is the premise: it is the raw text "containing" the human values we want to identify. The stance is the polarity of the argument, but does it tell something about the underlying human values? The conclusion is often a summarized version of the premise. We are only interested in identifying the presence of values, not the specific argument each sample is about. The winning submission [7] for the SemEval task, as hinted by [3], used a different approach, concatenating premise stance and conclusion into a single string to obtain a more human-like sentence. Concatenating the three parts, however, does not always yield a human-like sentence. To have some insight into what might be the best approach, I trained two models, one for each approach. Both models, fine-tuned DistilBERT[6] classifiers, have identical structure and training parameters. The training parameters are shown in table 4 appendix B. Repeating the process five times with different random seeds, the average macro F1 score for the model trained on premise, stance, and conclusion achieved a slightly better score (≈ 0.4285) compared to the model trained on only the premise (≈ 0.4267). While this test is not proof that one method is better than the other, and the performance difference is negligible, I used the concatenated premise, stance, and conclusion for the rest of the experiments.

Augmentation The training dataset is relatively small and has very few examples of less-represented labels. As an attempt to improve classification performance, I tried augmenting the original training dataset with various data augmentation techniques: token masking, text-to-text summarization, and back-translation. Since the available data is multi-label, it is not easily possible to augment only the most underrepresented classes, so I used the whole training set as input to the augmentation procedures. This left me with many possible combinations of training datasets, each containing more or less artificially augmented samples. To somehow measure if the augmented training data can be helpful to improve performances, I once again compared multiple fine-tuned DistilBERT models, training on various subsets of data. Training hyperparameters can be found in table 4 appendix B. Results in table 2 suggest no augmentation method is useful.

3.1 Models Overview

All the models described in this section have been trained on the original dataset, with no augmentation technique, using premise, stance, and conclusion concatenated in a single string.

summarized	masked	backtransl.	original and summarized	original and masked	original and backtransl.
0.406	0.396	0.419	0.421	0.413	0.422

Table 2. Average macro F1 score on the various dataset

Non-neural models In order to differentiate from the dataset author's model, I tried multiple non-neural models: random forest, logistic regression, and support vector classifier. All these models usually perform better if the data is pre-processed, for example, to remove punctuation and stop words. In this particular case, the pre-processing pipeline consists of decontraction, punctuation removal, numbers removal, stop-words removal, and stemming. After pre-processing, the data is encoded using a tf-idf vectorizer and passed to the model. In order to build multi-label classifiers, I used a one-vs-rest strategy. I used a coarse-grained grid search with 5-fold stratified cross-validation to select the best-performing model of this group. The grid search parameters can be found in table 5 appendix B. The resulting model is a support vector classifier with a linear kernel and parameter $C = 1$, achieving a test macro f1 score of ≈ 0.28, which is marginally better compared to the baseline model.

Transformers-based models I fine-tuned multiple models based on pre-trained language models, *xlnet-base-cased* [8], *deberta-v3-large*[1, 2], and *roberta-large* [4] trained from scratch by the authors of [7]. [4] Each model has a corresponding tokenizer to encode the input text, and as is the case with transformers model, no traditional data cleaning is needed. For each base model, I repeated the training process three times (because of limited resources and time) with different seeds and considered the average macro f1 score. The results are shown in table 3, while the training parameters are shown in table 5 appendix B.

The best-performing models are based on DeBertaV3 and Roberta, with an average macro F1 score of ≈ 0.489 across the 3 runs.

Ensembling Testing the method used by [7], I ensembled three transformers-based classifiers (using the same base models from the previous experiment) with the following strategy. First, train all classifiers on the test dataset. Then use the averaged predictions on a 300-sample holdout set to find the best global threshold for all labels. Finally, use the best-performing threshold to predict the final test set, again averaging the predictions of all the models. This approach resulted in a macro f1 score of ≈ 0.482, showing no particular improvement.

[4] `https://huggingface.co/danschr/roberta-large-BS_16-EPOCHS_8-LR_5e-05-ACC_GRAD_2-MAX_LENGTH_165`

Base Model	**Avg. macro F1**
DeBerta v3	≈ 0.489
RoBerta	≈ 0.489
XlNet	≈ 0.45

Table 3. Average macro F1 score from 3 repeated runs

4 Conclusion

Identifying human values in the text is a difficult task. Fine-tuning big pre-trained language models achieved better performances, while ensembling, non-neural machine learning models, and data augmentation proved ineffective. Building a model based on a bigger pre-trainer language model might be an “easy” way to obtain a better performance.

From a practical perspective, regarding my experiments, repeating the same process multiple times with different seeds does look redundant and does not make the result more robust.

The best performance achieved is a macro F1 score value of ≈ 0.49. From [7] we know better results are achievable. Future work could extend the approach by using different language models or by employing different techniques. Even bigger improvements could come from a bigger dataset with quality samples. Lastly, having more domain-specific knowledge about human values could prove useful to better understand and exploit available data.

References

[1] Pengcheng He, Jianfeng Gao, and Weizhu Chen. *DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing*. 2021. arXiv: `2111.09543 [cs.CL]`.

[2] Pengcheng He et al. “DEBERTA: DECODING-ENHANCED BERT WITH DISENTANGLED ATTENTION”. In: *International Conference on Learning Representations*. 2021. URL: `https://openreview.net/forum?id=XPZIaotutsD`.

[3] Johannes Kiesel et al. “Identifying the Human Values behind Arguments”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 4459–4471. DOI: `10.18653/v1/2022.acl-long.306`. URL: `https://aclanthology.org/2022.acl-long.306`.

[4] Yinhan Liu et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *CoRR* abs/1907.11692 (2019). arXiv: `1907.11692`. URL: `http://arxiv.org/abs/1907.11692`.

[5] Nailia Mirzakhmedova et al. *The Touché23-ValueEval Dataset for Identifying Human Values behind Arguments.* 2023. arXiv: `2301.13771 [cs.CL]`.

[6] Victor Sanh et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *ArXiv* abs/1910.01108 (2019).

[7] Daniel Schroter, Daryna Dementieva, and Georg Groh. *Adam-Smith at SemEval-2023 Task 4: Discovering Human Values in Arguments with Ensembles of Transformer-based Models.* 2023. arXiv: `2305.08625 [cs.CL]`.

[8] Zhilin Yang et al. “XLNet: Generalized Autoregressive Pretraining for Language Understanding”. In: *CoRR* abs/1906.08237 (2019). arXiv: `1906.08237`. URL: `http://arxiv.org/abs/1906.08237`.

Appendix A Dataset description figures

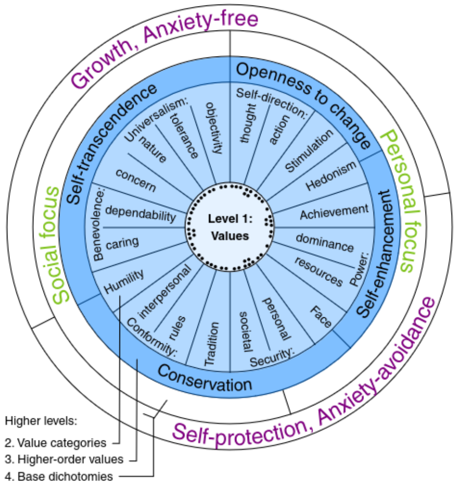

Fig. 1. Hierarchical categorization of human values. [3]

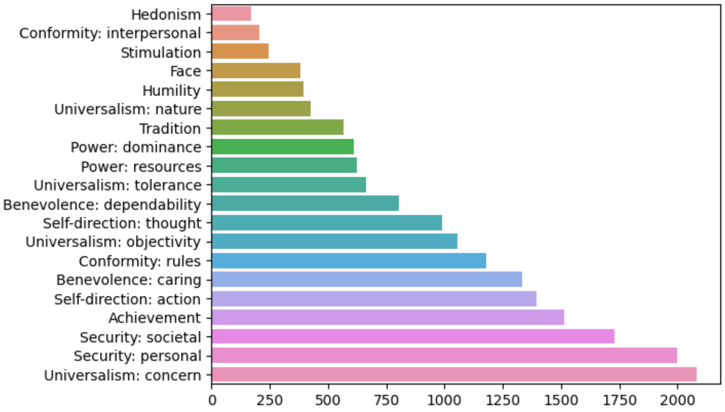

Fig. 2. Label distribution for the training set

Appendix B Model training parameters

Parameter	Value
batch size	8
epochs	10
warmup steps	500
learning rate	5e-5
gradient accumulation steps	2
weight decay	0.01
fp16	true
early stopping callback patience	3

Table 4. DistilBERT models training parameters.

Parameter	Value
batch size	8
epochs*	10
warmup steps	1348
learning rate	5e-5
gradient accumulation steps	1
weight decay	0.01
fp16	true
early stopping callback patience	2

Table 5. * the number of epochs is used to calculate the total number of training steps and the warmup steps. In practice, early stopping prevents from doing all the steps.

Model	Parameter	Value
RandomForestClassifier	-	-
LogisticRegression	C	0.1
	C	1
	C	10
SVC	C	0.1
	C	1
	C	10
	Kernel	linear
	Kernel	rbf

Table 6. Grid search parameters for non-neural model selection