Human Value Detection: a brief summary of simple neural network and transformer architectures

Vincenzo Collura, Gianmarco Pappacoda, and **Anthea Silvia Sasdelli**
Master's Degree in Artificial Intelligence, University of Bologna
{ vincenzo.collura2, gianmarco.pappacoda, anthea.sasdelli }@studio.unibo.it

Abstract

Human values are "trans-situational goals(...) that serve as guiding principles in the life of a person or group"(Schwartz et al., 2012). Therefore, humans tend to express natural language arguments guided by underlying values which are dependent on a number of factors such as the cultural context and transcend situations. The purpose of the challenge proposed by (Kiesel et al., 2022) is: given a textual argument and a human value category, to classify whether or not the argument draws on the Schwartz categories(Schwartz et al., 2012) of human values. The authors of this report propose a number of neural architectures to be evaluated for the task. Furthermore, the proposed architectures include "simpler" neural architectures as well as more complex ones, so-called "large language models" powered by the underlying architecture known as "transformers".

1 Introduction

As human values are underlying driving forces of humans' decisions, understanding them and being able to predict whether a human value is present or not in a given argument as such driving force is an important feat. Being able to rightly classify the speech values of an interlocutor can be very useful but also very complex. As a matter of fact, the sheer amount of emotions that populate the emotional spectrum of humans makes this operation very difficult.

The challenge proposed by (Kiesel et al., 2022) is: given a textual argument expressed in natural language and a human value decide whether or not the textual argument draws from the given human value.

The problem can be easily framed as a multi-label classification one.

The provided dataset is composed of 7289 arguments, expressed as triplets each composed of a *Premise*, a *Conclusion* and a *Stance* (i.e. whether the conclusion is in favour of or against the premise).

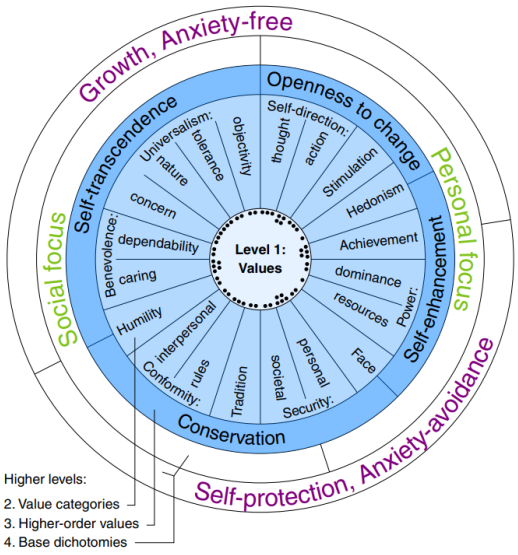

Figure 1: Image of the 20 value categories. Categories that tend to conflict are placed on opposite sites. Illustration adapted from (Kiesel et al., 2022) and (Schwartz et al., 2012)

Each argument is uniquely identified and is associated with 20 "level2" labels (a form of aggregation over "raw" human values). The test dataset does not contain such labels.

This project goal is to assess the performance of "simpler" neural architectures in conjunction with a dense vector representation of words and more complex pre-trained large language models architectures known as "transformers".

GloVe(Pennington et al., 2014) has been selected as dense vector representation the simpler neural architectures. Transformers-based solutions already use specific embeddings.

All the proposed solutions are trained, evaluated

against validation and test sets using the metrics proposed by the paper authors (F1-score, Precision and Recall using macro average).

2 Background

To proceed in the analysis and the implementation of this project the (Kiesel et al., 2022) paper was fundamental. The advantage given by a customized dataset of 5270 arguments from four geographical cultures, manually annotated for human values allows to overcome the obstacle of the large variety of the human values, by accounting for many different cultural contexts and providing the survey with cultural-specific questions.

The aforementioned work is based on (Schwartz et al., 2012) that establishes the cornerstones for the classes of human values to be defined.

"The theory defines and orders 19 values on the continuum based on their compatible and conflicting motivations, expression of self-protection vs. growth, and personal vs. social focus." (Schwartz et al., 2012)

Another important psychological paper concerning human values is "The Nature of Human Values"(Rokeach, 1976). He estimates the total number of human values to be fewer than hundreds, and develops a practical survey of 36 values that distinguishes between values pertaining to desirable end states and desirable behaviour.

It has been noted that in the data the premise or the conclusion can be used multiple times with different respective stances.

What happens is the reframing of the text, which means the change of perspective with respect to an issue. In fact, it is possible that, from the same premise, the consequent conclusion captures different aspects or topics. This is important also for the understanding of the data in the preprocessing. This topic is developed in the paper (Chen et al., 2021)

3 System description

In order to ensure reproducibility a seed (i.e. number) has been set for all the random number generators (RNGs) involved in the process. While the authors have striven to ensure reproducibility, some employed libraries such as HuggingFace exhibited a fluctuating behaviour with respect to this issue.

The first performed step was the download of the dataset, the arguments and the labels ones, each already splitted in train, test and validation.

As test set labels are not available unless actively participating in the challenge, the authors have decided to discard the test set and produce another test set by randomly selecting samples from the training set up to $\approx 15\%$ its size (≈ 800 samples).

In order to better grasp the nature and the distribution of the data, an explanatory data analysis phase has been carried out, revealing the dataset labels to be unevenly distributed (please refer to Figure 2).

Another aspect that was checked is the the distribution of sentences length (accounting for the summation of premise, stance and conclusion): in training and validation sets are, as hypothesized, comparable. This ensures the validation set is employable to perform validation during training. Moreover, it is possible to cut the sequences in order to reduce the computational power needed to perform both training and inference. In order to perform the cut, the 99th percentile of the lengths has been selected. Needless to say, were the distribution of the documents sentences to change, the maximum allowed sentence length would have to be adjusted accordingly.

The system has been split into two different sections:

- Simpler network architectures employing dense vector representations (GloVe only)
- Transformer-based models repurposed (i.e. fine-tuned) for the task

In both sections the training data has been preprocessed by concatenating the stance (converted to a single word: favor/against), the conclusion and the stance. The input data is then converted to lower-case.

In the first section the data is also stripped of punctuation as it could impair model training. The lower-case transformation is very important for the first section since GloVe, that is the chosen embedding method, only has lower-case mappings. This passage is fundamental in order to check and find real Out-Of-Vocabulary (OOV) words contained in the data.

The word embedding has been implemented as part of the neural network in the form of a neural network layer. The embedding layer depends

on the building process of the embedding matrix which in turn depends on the vocabulary. During this step words whose embedding is known are translated, words whose embedding is not known (OOVs) are translated into vectors whose dimensions have a uniform distribution.

The first proposed architecture, "BiLSTM" is a simple Bidirectional LSTM layer with 36 units, its output is passed across an average pooling and a max pooling layer in a parallel way, the output is then feed into a fully-connected layer using ReLU as activation function.

The second proposed architecture, "CNNText" implements a series of convolutional layers with 36 filters and increasing sizes of kernels (1,2,3,5), no padding, stride $= 1$. The input is flattened and passed into the first layer. Once passed the last layer

The aforementioned two architectures are adaptations of (Agarwal, 2020).

The second section describes the implementation of transformer-based solutions. The authors proposed `BERTTiny`(Bhargava et al., 2021; Turc et al., 2019) $\approx$ $4.5M$ parameters and `DistilRoBERTa-base`(Sanh et al., 2019) ($\approx$ $82M$ parameters). The chosen architectures are among the smallest transformers while delivering near-SOTA performance in many tasks.

In both transformer architectures, the final part of the neural network (often referred to as "head") has been cut off and replaced by fully-connected layers and a dropout layer in order to repurpose the language models.

4 Data

The dataset used for this project was provided by Touché23-Human-Value-Detection (Kiesel et al., 2022).
The original dataset is pre-split into training, test and validation splits, however the test split is not labelled, therefore the authors have decided to discard it in favour of a new test set obtained by randomly extracting samples from the training set up to $\approx 15\%$ of its size (≈ 800 samples).

The dataset is further split into two main sections, the first one containing arguments as triplets and the second one containing the labels.

Each sample in the first section is composed of

- the argument ID
- the conclusion, which is the final stance with respect to the premise.
- the stance, which has only two values: "in favor of" or "against". It connects the conclusion and the premise by specifying whether the former is in favor of or against the latter.
- the premise, which is a text or a sentence that gives the context, the core, of the argument.

The second section contains labels. Each sample is composed by 21 features, representing the argument ID followed by the 20 "level2" value categories. Each feature can either be 1: the argument draws on that category, or 0 the argument does not draw on that category. Categories are not mutually exclusive.

5 Experimental setup and results

A seed has been set for all the involved frameworks as to ensure reproducibility. To accomplish the goal of this project two variants of neural networks have been chosen: simple neural networks with dense vector representations and transformers-powered large language models. The starting idea being to start the implementation with a simpler architecture, and then compare the results with a more complex system, such as large language models.

In the first solution the data was preprocessed, in order to check the maximum length of the sentences in the arguments datasets and the distribution of sentences length in training and validation sets. Then the vocabulary was built, using GloVe(Pennington et al., 2014) as embedding with dimension equal to 300 and always taking into account and adding the OOV words at each step (training, validation, test).

As the neural networks output are in the form of logits, a sigmoid function has been used on the outputs. This projects the outputs into the $[0, 1]$ interval. This, however poses another problem, the choice of a *threshold hyperparameter* to convert the predictions into their crisp label counterparts.

Each neural network has been trained using early stopping, adaptive learning rate and Adam(Kingma and Ba, 2014) optimizer. The learning rate has been changed during the experiments for each model, in the end the authors deemed learning rate $= 10^{-3}$ to be suited for both.

For the second approach, the `DistilRoBERTa-base` and `BERTTiny` models have been chosen. In this case the data has been preprocessed in a different way, since `DistilRoBERTa-base` and `BERTTiny` have their own embeddings.

In order to have a better structured input for the model, the data is encoded in the following way: first the Stance is transformed in a special token, like a boolean variable, that can be only "favor" or "against"; this is followed by the Conclusion and the Premise, all separated by a white space.

For both solutions, many experiments have been carried out: in order to check the versatility of the systems, the weights have been frozen. This solution was immediately discarded, as the results obtained for both models were poor, while with unfrozen weights the overall performance was improved.

The loss functions used for training the neural networks is the cross-entropy. In information theory, the cross-entropy between two probability distributions `p` and `q` over the same underlying set of events measures the average number of bits needed to identify an event drawn from the set if a coding scheme used for the set is optimized for an estimated probability distribution `q`, rather than the true distribution `p`.

$$H(p, q) = -\sum_{x \in \mathcal{X}} p(x) \log q(x) \quad (1)$$

$$H(p, q) = -\int_{\mathcal{X}} P(x) \log Q(x)\, dr(x) \quad (2)$$

The cross entropy loss function (Equation 1 for discrete, equation 2 for continuous) is instead a metric used to measure how well a classification model in machine learning performs. The loss (or error) is measured as a number between 0 and 1, with 0 being a perfect model. The goal is generally to get the model as close to 0 as possible. The HuggingFace training method for this task defaults to binary cross-entropy with logits, which has been empirically observed to be worse than cross entropy in this task, the authors have therefore implemented and used cross entropy only.

The proposed metrics used were F1-Score, precision and recall with macro average. Macro-level metrics provide a view across the entire organization or function, since the macro metrics are calculated as the arithmetic mean of individual classes scores.

The macro-average precision and recall score is calculated as the arithmetic mean of individual classes' precision and recall scores. The macro-average F1-score is calculated as the arithmetic mean of individual classes' F1-score.

Model	F1	precision	recall
BiLSTM	0.25	0.19	0.35
CNNText	0.46	0.40	0.56
BERTTiny	0.39	0.39	0.43
DistilRoBERTa-base	0.48	0.54	0.53

Table 1: Experimental results on the test set (macro)

The overall performance of the models can be further explored on a per-label basis in the 3 figure.

6 Discussion

Fine-tuning transformers models by freezing intermediate layers did not prove to be a good strategy for training. Instead, performing a full training process with lower learning rates yielded better results. In particular, the embedding layers training was observed to be extremely important. This is partly explicable by OOVs but more intuitively by the fact the embedding is the very representation of language components a model has.

Most models could easily exceed the chosen baseline: Table 2 (provided by (Kiesel et al., 2022)). While this has been the case for most, the `BiLSTM` model which leveraged a recurrent neural network design performed considerably worse than all the other models.

The `CNNText` model performed incredibly well despite its simple architecture and the use of convolutions that is usually associated with computer vision tasks.

The best model, `DistilRoBERTa-base`, has been trained for 15 epochs with learning_rate $= 2 \cdot 10^{-5}$, the chosen model belongs to the last epoch and the validation loss for the last epoch still showed ongoing improvement, thus leading the authors to believe the model to be under-trained and potentially even more capable.

Another aspect that has been noted is the fact that the loss and validation loss are diverging after several epochs for most models. While the chosen metrics (F1-score macro) may keep growing when the model starts overfitting, this does not directly translate into the chosen metric growing against the validation set. Therefore, the method used to select

the model has been selecting the moment in time (epoch) with the lowest validation loss.

Most hyperparameters were chosen empirically using manual tuning, this has led the authors towards the shown results, an extensive hyperparameter search may be conducted in order to squeeze further performance out of existing models.

Model	F1	precision	recall
BERT	0.34	0.39	0.30

Table 2: Baseline model(Kiesel et al., 2022)

7 Conclusion

It has been possible to implement and evaluate the chosen models. Among all the architectures proposed, all of them leveraged pre-trained embeddings. Upon freezing the embeddings weights (i.e. no word representation can be learned) it has been observed that models performed considerably worse. Because of this, the authors have decided not to persist in this practice and have abandoned it early in the process.

Simpler neural architectures have proven to be less effective than pre-trained transformers, nevertheless the `CNNText` performance is very close to the best transformer model (`DistilRoBERTa-base`) while also employing $\approx 66\%$ more parameters.

While `BERTTiny` showed clear signs of overfitting in early stages, lowering the learning rate yielded slower learning, longer times and no significant improvements.

The best model `DistilRoBERTa-base` was trained for 15 epochs, but showed constant yet small improvements, the underlying hypothesis being it has been under-trained, thus showing potential for further improvement.

Up until the last experiment, the `CNNText` model performed significantly better than other solutions, included `BertTiny`. This method shows a lot of potential but is hardly scalable and lacks the generalization provided by large language models, it is therefore interesting to observe it compared to large language models but should not be preferred given its size.

The main issue with these large language models is their use of attention which is quadratic in complexity, thus imposing hard constraints on the training process. Exploring models that employ different forms of attentions, especially if not quadratic in nature, could potentially lead to faster training times while preserving the achieved performance.

References

Rahul Agarwal. 2020. Multiclass Text Classification - Pytorch. https://www.kaggle.com/code/mlwhiz/multiclass-text-classification-pytorch/notebook. [Online; accessed 05-Jan-2023].

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in nli: Ways (not) to go beyond simple heuristics.

Wei-Fan Chen, Khalid Al Khatib, Benno Stein, and Henning Wachsmuth. 2021. Controlled neural sentence-level reframing of news articles. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2683–2693, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Johannes Kiesel, Milad Alshomary, Nicolas Handke, Xiaoni Cai, Henning Wachsmuth, and Benno Stein. 2022. Identifying the human values behind arguments. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4459–4471, Dublin, Ireland. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

M. Rokeach. 1976. *The Nature of Human Values*. Free Press.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Shalom Schwartz, Jan Cieciuch, Michele Vecchione, Eldad Davidov, Ronald Fischer, Constanze Beierlein, Alice Ramos, Markku Verkasalo, Jan-Erik Lönnqvist, Kursad Demirutku, Ozlem dirilen gumus, and Mark Konty. 2012. Refining the theory of basic individual values. *Journal of Personality and Social Psychology*, 103:663–88.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

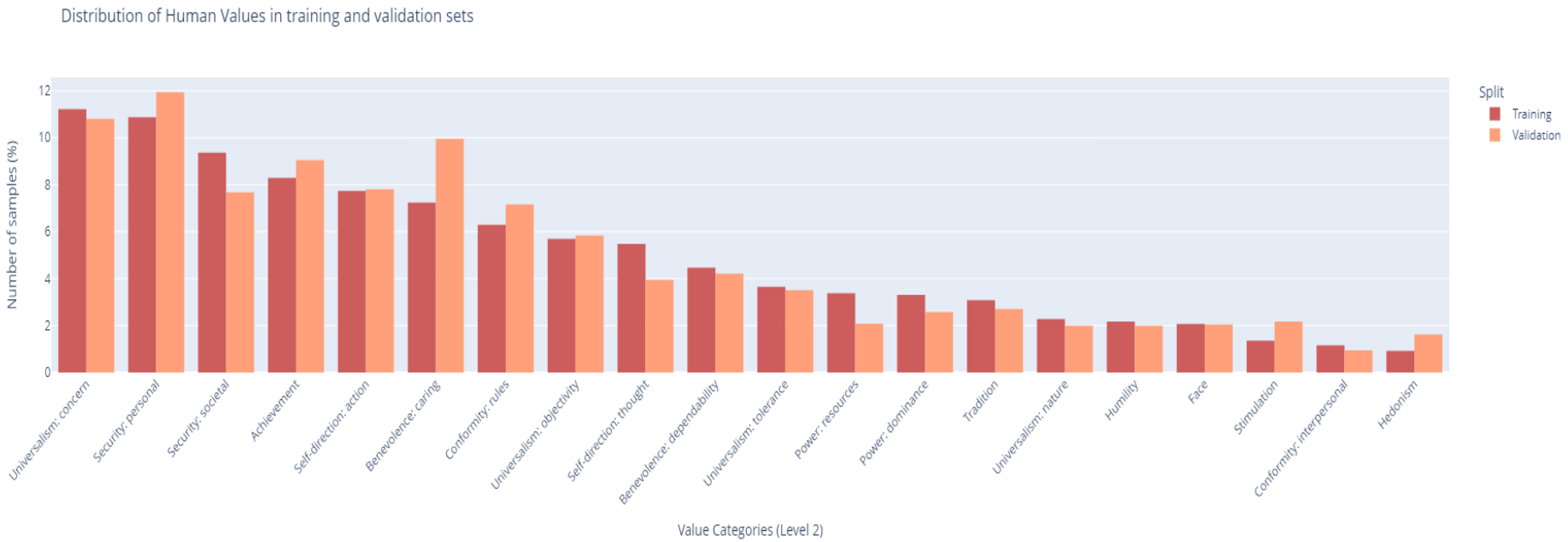

Figure 2: Labels distribution

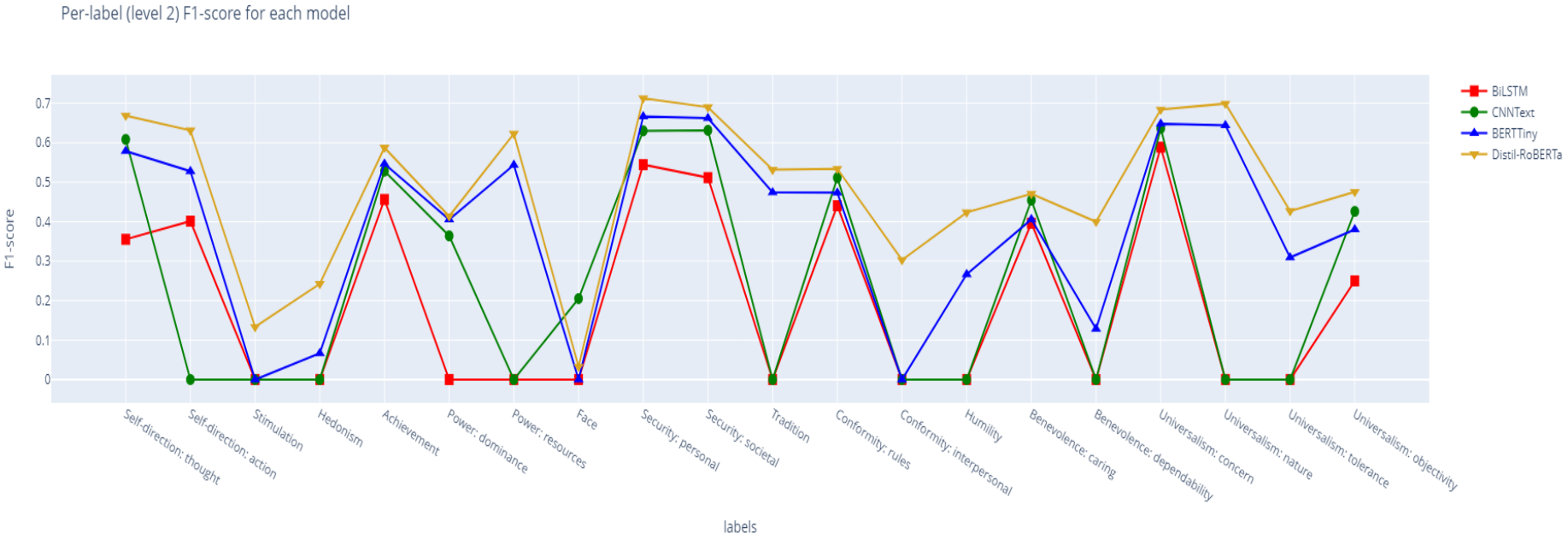

Figure 3: Models performance (F1-Score macro)