Identification of Human Values behind Arguments
NLP Course Project

Alessandro Lombardini, Giacomo Melacini, Matteo Rossi Reich and **Lorenzo Tribuiani**
Master's Degree in Artificial Intelligence, University of Bologna
{ alessandr.lombardin3, giacomo.melacini, matteo.rossireich, lorenzo.tribuiani }@studio.unibo.it

Abstract

Based on the Human Value Detection challenge 2023, this project discusses the implementation of NLP models capable to classify *Human values* on which a specific text relies on. This task uses a set of 20 value categories compiled from the social science literature and described in the paper *Identifying the Human Values behind Arguments* (Kiesel et al., 2022).

1 Introduction

Human value detection represents a specific and particular NLP task involving different aspects of the subject from which the solving process takes inspiration. This task could easily be compared with a *sequence clustering* process since the possible labels are not easily grouped uniquely. For this reason, the whole paper is based on a dataset with a hierarchical multi-level labeling system with 54 different labels of which, for the aim of this project, only the 20 inner categories are taken into account. This brings the whole project much closer to a *sequence classification* over a specific set of possible classes. According to this, the solving project is widely based on the *Huggingface*'s **Model For Sequence Classification**. Those models are based on pre-trained systems that can be fine tuned on the specific data in order to achieve a better understanding of the task itself. The process has been divided into different steps:

- *models selection*: The first step has involved testing different pre-trained model in order to select (based on the baseline) the best performing ones. This process has lead to the selection of four different base models:
  - *bert-base-uncased*
  - *bert-large-uncased*
  - *distilbert-base*
  - *roberta-base*
- *Hyperparameters tuning*: After choosing the most promising models different runs has been made with different parameter in order to detect the best performing $model - parameter$ set.
- *Best model evaluation and error analysis*: Once the best model has been selected a series of evaluation and error analysis has been conducted.

2 Background

One of the challenges we faced during this project was keeping track of all the different training runs. Being able to effectively visualize the effects of the tuning makes a big difference. For this reasons, we decided to use Weight&Biases to keep track of all the results in one place and leverage on its visualization capabilities. Moreover, it does provide us with a convenient way to store the checkpoints of our models, this allows us to fetch them at need while doing error analysis.

Another challenge of this project regards the class imbalance in the splits. In order to address this problem, some custom losses were used, like a weighted loss. The idea it to assign a higher weight to samples of less frequent classes and vice-versa.

The other loss which was used is the distribution balanced loss. By integrating rebalanced weighting and negative tolerant regularization (NTR), distribution balanced loss first reduces redundant information of label co-occurrence, which is critical in the multi-label scenario, and then explicitly assigns lower weight on "easy-to-classify" negative instances (Huang et al., 2021).

Speeding up training is always beneficial, especially when it comes with little drawbacks, therefore mixed-precision training was used. Using a numerical format with 16-bit floating point precision has three main benefits: smaller memory requirements which allows bigger models, less

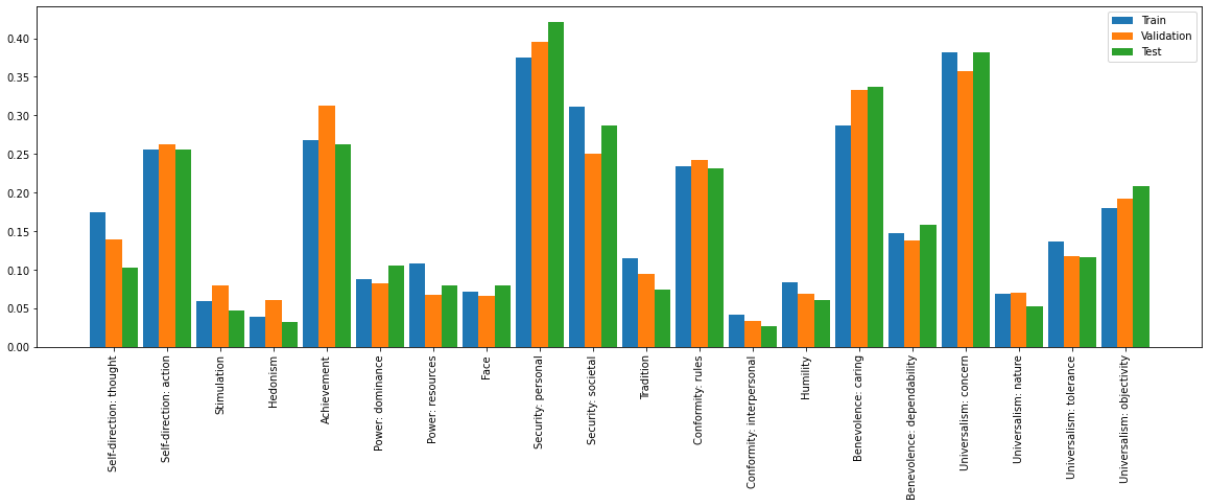


Figure 1: Distribution of the labels in the train, validation and test set.

memory bandwidth which speeds up data transfer operations and faster mathematical operations. Mixed precision training allows to use full precision only in needed steps ensuring no accuracy is lost(Micikevicius et al., 2017).

In order to speed up convergence and eliminate "hockey stick" loss curves, where in the first few iteration the network is basically just learning the bias, the biases of the classification head were initialized according to the class distribution as suggested in Andrej Karphaty's blog.

3 System description

The experiments have taken place in a pipeline that is developed as follows:

1. The dataset is preprocessed (see section 4 for details);
2. The hyperparameters are chosen (see section 5 for details) and an Hugging Face trainer is created;
3. In order to monitor the training, a Weight & Biases instance is created;
4. The model is trained according to the selected hyperparameters;
5. The results are analyzed (see section 6 for details).

Several architectures have been tried such as bert-base-uncased, distillbert-base-uncased and roberta-base. All these models have been loaded using Hugging Face in a pretrained version. The classification head of those models has been substituted with a linear layer having size 20 (as the number of possible human values). Since the problem is multilabel, a sigmoid activation has been used.

The metric function has been adapted from jesusleal.io by personalizing the computed metrics, while the class used to compute the distribution-balanced loss was taken from the implementation of the paper (Huang et al., 2021).

4 Data

The dataset was accessed from the Human Value Detection 2023 task website. This dataset contains arguments labeled with the human values associated to them. Each argument can have more than one value associated to it. This makes this task a multi-label classification problem. The arguments are divided in three parts: a premise, a stance (in favor or against) and a conclusion. Since the test dataset does not come with its labels, it was decided to further split the validation set into validation and test. The train set comes with 5220 examples, the validation with 1516 and the test set with 380 examples. In order to make the experiments sounder, the models have also being tested with a dataset (100 examples) containing arguments from the recommendation and hotlist section of the Chinese question-answering website Zhihu. The labels distribution (image 1) has been analyzed and it was discovered that the distribution inter-split was similar. On the other hand, the distribution intra-split is not uniform: some values are much more frequent than others. This means that the dataset is not balanced. In order to improve the model it may be necessary to add weights to rebalance the classes.

Also the distribution of the lengths of premises and conclusions have been analyzed. While the conclusions' lengths showed to have a small standard deviation, the premises had some examples with a very high length. This would force a very long padding to the majority of the examples. In order to reduce the padding, and thus speed up learning and save memory, it was decided to truncate the arguments to the 99% percentile of the lengths of the tokenized arguments in the training set as shown in table 1. The arguments have been tokenized using Huggingface's auto-tokenizer with the following format: *{conclusion}SEP{stance}SEP{premise}*. It was decided to truncate the right end of the examples. In this way the information loss would cut just a part of the premises (that, as mentioned before, are the ones with a high variance in length), while the conclusion would remain intact. In this way also the truncated examples should keep information enough to be correctly classified.

50-th	75-th	90-th	99-th
33	40	52	60

Table 1: Percentiles of the lengths of the arguments for all the splits

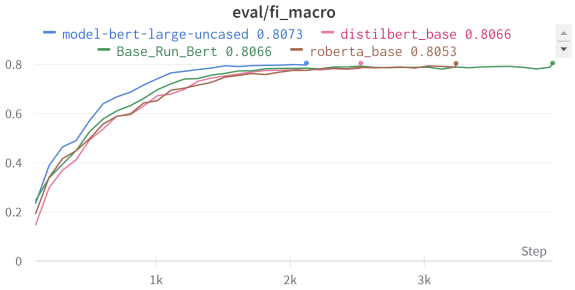
5 Experimental setup and results

In order to compare the results of the of the experiments, it was set up a baseline as per lesson of (Kiesel et al., 2022). It classifies each argument as resorting to all values. All the models described in section 3 have been trained for 30 epochs with early stopping with 4 epochs patience. Initially, the hyperparameters have been set as in (Kiesel et al., 2022), meaning with a batch size of 8, a constant learning rate of 2^{-5}, no weight decay and using BCE loss.

Model	Val	Test
baseline	0.271	0.276
bert-base	0.7891	0.8066
distilbert	0.7847	0.8066
roberta	0.7895	0.8053
bert-large	**0.7983**	**0.8096**

Table 2: F1 score macro of the models with base hyperparameters set

The F1 score macro on the test set for these base runs can be seen in table 2. Moreover, it is possible to see a comparison between the base runs of each architecture in the graph 2a.

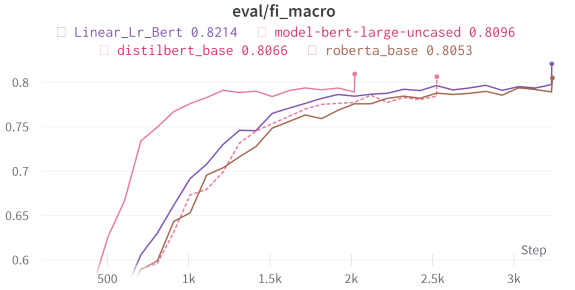


(a) f1 score comparison between base runs of each architecture

The hyper-parameter tuning has been carried out on the following parameters:

- models: *distilbert base uncased*, *bert base uncased*, *roberta base* and *bert large uncased*
- learning rate scheduler: constant and linear
- weight decay
- loss function: BCE loss, BCE loss with class weights and distribution balanced loss

Performing grid search would have been cumbersome, therefore we decided to explore the hyperparameters space based on the most promising results according to the validation and test scores. In the following graph a comparison over the f1 score between the most promising models according to *f1 macro* (2b) is shown.



(b) f1 score comparison between the four best models

It is possible to see that Bert with a linear learning rate is the best performer by some margin, as also shown in table 3.

Model	Val	Test	Zhihu
baseline	0.271	0.276	0.19
bert-base	**0.7977**	**0.8214**	**0.9**
distilbert	0.7847	0.8066	0.847
roberta	0.7895	0.8053	0.898
bert-large	0.7894	0.8096	0.865

Table 3: F1 score macro of the best model for each architecture

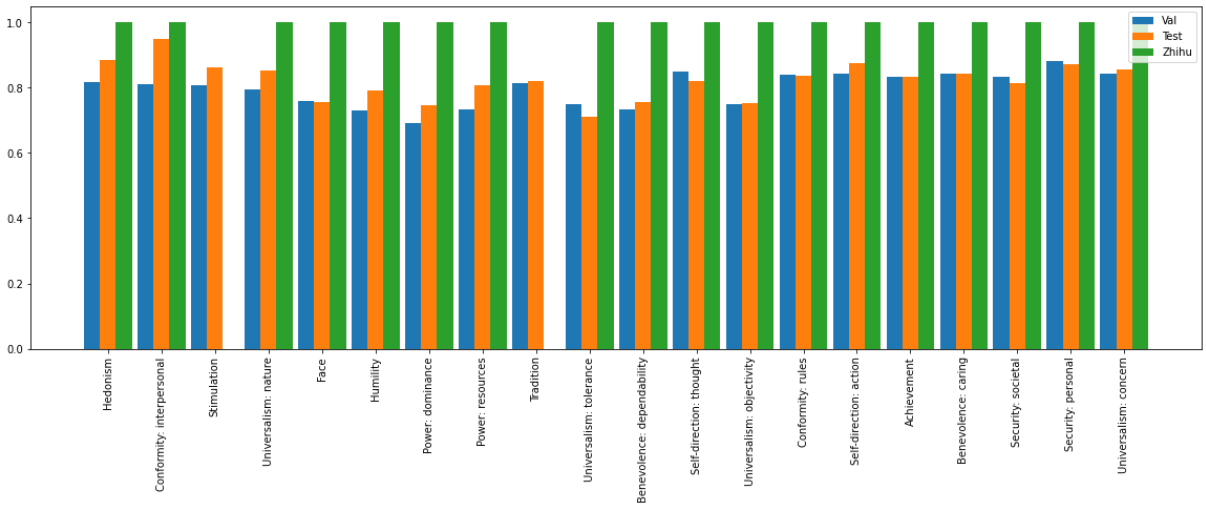
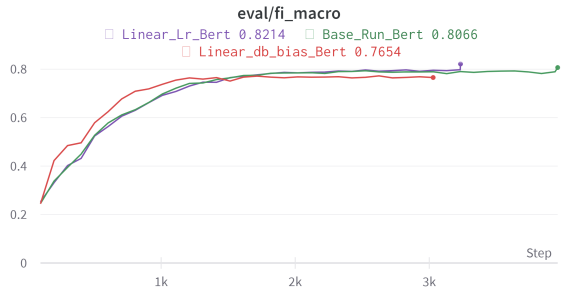


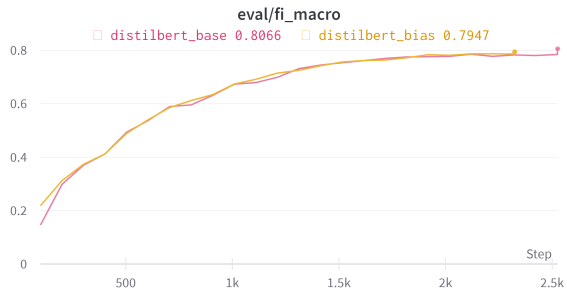
Figure 2: F1-score for each label in validation, test and Zhihu dataset. The labels are sorted based on their frequencies in the training set.

Other graphs are presented to show, for each architecture, the difference between various hyperparameters sets(3a, 3b):



(a) f1-macro score comparison between *bert-base-uncased* models. It is shown the base one, one using linear lr scheduler with BCE loss and one using linear lr scheduler with distribution balanced loss

It seems that the the distribution balanced loss has an effect in making the model learn faster but it does also plateau sooner.

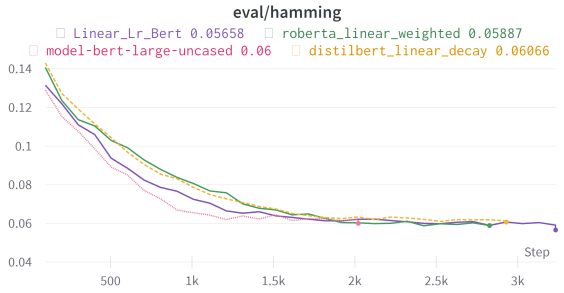


(b) comparison over *bias initialization* and base model for *distilbert-base*

Image 3b shows a comparison between a distilbert model with and without bias initialization.

Model	Val	Test	Zhihu
baseline	0.1617	0.1703	0.1165
bert-base	**0.755**	**0.763**	**1**
distilbert	0.743	0.745	0.994
roberta	0.751	0.758	0.997
bert-large	0.734	0.739	0.99

Table 4: Hamming score of the best models for each architecture



(a) comparison of the different architectures on *hamming loss*

Image 4a shows a comparison of the Hamming loss between different models. The evaluation of the Hamming score on the best models is shown in table 4.

6 Discussion

The results achieved by the models are good. The metrics of the best model greatly outperforms the baseline. After experimenting with different transformer architectures, it was shown that the difference among them, for this specific task, is not large. Still, as it can be seen from table 3, the architecture that performed best based on f1-macro average was Bert (with linear learning rate decay and a BCE

loss). It is interesting to notice that the ranking of the four best models changes depending on the metric we consider. If the hamming loss is taken as main metric, the Roberta based model improves its performance by a great deals 4a.
As it can be seen in image 3b, the initialization of the bias in the classification head made the convergence faster. Still, it does not influence the result in the long term.
Surprisingly, as shown in image 2, the frequency of the labels in the training set does not influence their f1-score: all the labels have similar good performance. This means that the problem of class-imbalance has been greatly dealt. Another thing that it is possible to notice by looking at the image is that the performance on the Zhihu dataset is very good. This shows both that the model is capable of transferring its knowledge to different cultures and that the Zhihu dataset is probably a simpler one w.r.t. the train, validation and test sets.
Here is shown an example of an error done by our best model on the test set:

- *Premise*: The 'War on Drugs' gifted to us by the USA is absolute rubbish. Sugar is more dangerous than Cannabis! How many people have died on 'legal highs'? FAR too many. The death toll from Cannabis use = 0. More lives have been destroyed by drug laws than the drugs themselves. Besides, Cannabis is an herb, not a drug. Many herbs have medicinal properties. If any becomes a health issue, that's how it needs to be addressed, as a health problem, not a crime!

- *Conclusion*: We should legalize non-violent drug use

- *Stance*: in favor of

In such a case identifying the correct labels is not an easy task, in our opinion even an human would have had troubles getting all the correct labels. The model is still able to identify two correct labels and some of those which were incorrectly predicted could, in our opinion, be considered adequate.

Value	Predicted	True
Security: societal	1	0
Benevolence: dependability	1	0
Universalism: concern	1	0
Security: personal	1	1
Universalism: objectivity	1	1
Self-direction: action	0	1
Stimulation	0	1
Hedonism	0	1
Achievement	0	1

7 Conclusion

The classification of Human Values behind arguments is a challenging task, but thanks to huge pre-trained trasformers models at our disposal it was fairly easy to address. Transfer learning allows to deploy solutions very quickly by saving us from prohibitive training and letting us focus on fine-tuning the model on our specific task. Different architectures haven been tried and each of them has been tuned according to a set of hyper-parameters. Their performance has shown to be similarly good. Results have been quite good from the beginning but some future improvement might come from including the semantic meaning of the labels in the classification problem as described in this paper.
Another future challenge would come from including other Human Values, as per (Kiesel et al., 2022), and to classify them based on their hierarchical structure.

References

Yi Huang, Buse Giledereli, Abdullatif Köksal, Arzucan Özgür, and Elif Ozkirimli. 2021. Balancing methods for multi-label text classification with long-tailed class distribution.

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.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2017. Mixed precision training.