Human Value Detection with Hierarchical Multi-label Text Classification
NLP Course Project

Elisa Ancarani, Enrico Benedetti, Stefano Fantazzini, and **Irene Gentilini**
Master's Degree in Artificial Intelligence, University of Bologna
{ elisa.ancarani4, enrico.benedetti5, stefano.fantazzini, irene.gentilini2 }@studio.unibo.it

Abstract

This work explores the task of identifying human values in text arguments that was proposed in the paper by (Kiesel et al., 2022). We propose a hierarchical approach for multi-label text classification (HMTC), partially following the one proposed in (Cerri et al., 2014), with five different variations. In our experiments, we test the effects of Multi-Layer Perceptrons (MLP), Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) models as classifiers, and BERT for embedding text. We have found that the task may be too difficult for achieving good results, as our hierarchical architectures were not able to learn to better classify fine-grained values with respect to the previous flat approach, maybe due to the limited data available. However, we do not rule out that the use of other hierarchical approaches could be applied to this task more successfully.

1 Introduction

Value Theory, also called Axiology, can be described as the intent to classify what things or actions are good, and how good they are (Schroeder, 2021). These things or actions are referred to as values, and they are a central concept in the field of social science and play an important role in other disciplines as well. The theory of values concerns the structure of human values, which is similar across culturally diverse groups. This suggests the existence of a universal organization of human motivations, without denying the difference between individuals and groups in the attribution of relative importance to values (Schwartz, 2012). Human values play an important role in most aspects of human communication, from marketing to religion. In particular, human values are prominent in discussions, and it is therefore interesting to know whether it is possible to extract them from arguments in an automatic way.

The task at hand falls under the category of argument mining, a research field of natural language processing, which consists of the automatic identification and extraction of structured arguments expressed in unstructured natural language text, (Lawrence and Reed, 2019).

At the same time, this task is most similar to opinion mining, which shares some analogies with argument mining (Lippi and Torroni, 2016), and whose objective is to identify and extract people's thoughts, without focusing on the reason for these thoughts, unlike argument mining (Lawrence and Reed, 2020). A known approach to opinion mining is to use Support Vector Machines. This kind of model was even used in the original work we are expanding (Kiesel et al., 2022). However, this paper expressed that a promising approach would be to rely on Hierarchical classification, hence why we worked on building a Hierarchical Multi-label Text Classification (HMTC) model. Our dataset is made up of values structured hierarchically over 4 different levels, and arguments that need to be classified as one or more of said values at a time, in each hierarchical level. For this reason, an HMTC approach appears to be an interesting possibility for solving this task. We tested 5 models, all based on HMTC. The first one is our baseline model as it simply employs the usage of a text vectorization layer. The second one incorporates skip connections, which allow to keep the original input at every layer of the hierarchy in the whole network. The third one implements BERT, a transformer-based model used instead of the Text Vectorization Layer for the encoding of the input. The fourth and fifth models employ respectively a convolutional neural network (CNN) and a long short term memory network (LSTM).

The dataset we used was the Webis-ArgValues-22, which is organized in hierarchical levels, so our training was set up such that each level was trained separately with a dedicated Network. To take ad-

vantage of the hierarchical structure, the output of a level is then given as input to the following level.

The results of the models were underwhelming, and there could be many reasons for it. The main problem seems to be that our dataset is quite small, and that some labels at the lower level have less support compared to the higher ones. Aside from this, it is possible that the poor results may be due to the complexity of the model. A telltale sign of this is that the results for level 4 are similar to those of the original paper, but they degrade along the other hierarchical levels.

2 Background

The Webis Group widens its research into several artificial intelligence fields and organizes numerous initiatives, including Touché, a series of scientific events and shared tasks on computational argumentation and causality.

Human value detection is a task proposed by (Kiesel et al., 2022) from which arose a challenge presented by Touché for the 17th International Workshop on Semantic Evaluation (Mirzakhmedova et al., 2023). The original work discusses three natural language processing and machine learning-based approaches which are BERT, support vector machines (SVM) and a baseline approach which consists in classifying each argument as resorting to all values.

A promising approach for multi-label text classification we have relied on is to consider the classes involved in the classification as hierarchically structured, namely hierarchical multi-label text classification (HMTC). As labels are hierarchically interrelated, it is challenging to learn a mapping between arguments and labels and it can also lead to finding an intrinsic hierarchical structure between the arguments (Chen et al., 2019). Moreover, in the hierarchical method the structure of our data is taken into account, preventing labels from being processed separately, resulting in the need to train a classifier for each label (Nayak et al., 2013).

In their survey, (Jr and Freitas, 2011), presented the hierarchical classification (HC) task from many different points of view, and we focused on the fact that these methods are able to optimize a loss function either locally or globally. In the medical domain, (Cerri et al., 2014) described a local-based HMTC with Multi Layer Perceptrons (MLP) trained incrementally level by level in the hierarchy, for the purpose of tackling a protein function and gene prediction task. (Wehrmann et al., 2018) employs two versions of hierarchical multi-label classification (HMC), where each layer corresponds to one level of the label hierarchy, to optimize the loss functions both globally and locally, with a feed-forward network in the first version, and a recurrent architecture inspired on Long Short-Term Memory (LSTM) in the second (Hochreiter and Schmidhuber, 1997a). (Zhou et al., 2015) combined the strengths of two architectures for text classification purposes, respectively a convolutional neural network (CNN) to extract a sequence of higher-level phrase representations which are fed into an LSTM to obtain the representation of the sentences. The work of (Gong et al., 2020) designed a Hierarchical Graph Transformer for HMTC and discussed the usage of a weighted cross entropy loss to tackle the problem of uneven distribution of label categories. In this project, we took inspiration from these works by examining the different nuances in order to implement different hierarchical approaches to fit our problem.

3 System description

3.1 Base architecture & training process

The main architecture we studied in this work is a neural network for HMLTC from (Cerri et al., 2014). See Figure 1 for an illustration of the architecture. The model is made from a cascade of multi-label classifiers (their output activation is the sigmoid function as the labels are non-exclusive). In the original work of (Cerri et al., 2014), those classifiers were Multi-Layer Perceptrons (MLP), where each of them is associated with a level of the hierarchy (see Section 4 and Figure 2). This structure allows each block to compute probabilities for the labels of the corresponding level, and feed those scores to the following sub-network in the sequence, which will predict finer-grained labels using the information coming from the output layers of the previous classifiers.

We trained our networks following the method in (Cerri et al., 2014). Starting from the multi-label classifier for the highest level in the taxonomy and ending with the classifier for the human values, each classification block was trained one at a time on the training data. In our case, instead of building the architecture by training each block and attaching them one after the other, we decided to build the full architecture and training multiple times with

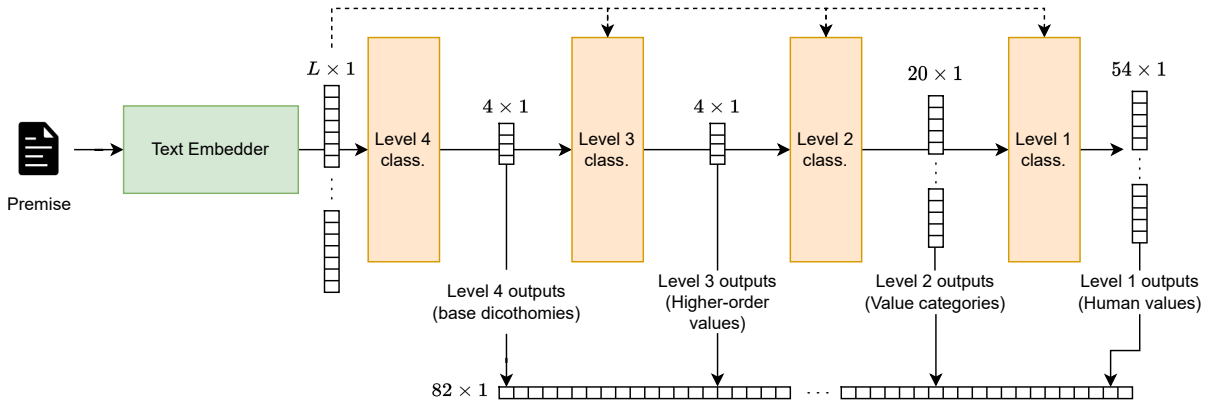

Figure 1: Our network architecture. The dotted line represents a residual connection from the text embedding to the inputs of downstream classifiers that is present in some variations of the models.

the same data, keeping all classifier blocks frozen except for the one that was training at that time.

As we descended the levels of our taxonomy, we noticed that the output activations of the classifiers were particularly low, meaning that our models were not able to learn and correctly predict those labels with high confidence. As a result, we employed a technique for choosing thresholds by adapting an online tutorial about threshold moving (Brownlee, 2021) for the multi-label case. This method allowed for the automatic selection of the threshold that granted the best F_1-score on the training set, for each label separately. These thresholds were then fixed and used in the final evaluations.

Using TensorFlow and Keras, we implemented the architecture and then experimented by putting different blocks in place of the MLP block and the input processing block, and changing their parameters (such as the number of layers and their dimension). We implemented the training and evaluation loop building upon a TensorFlow tutorial (Chollet, 2020).

3.2 Architecture variations

In the following section, we delve into the variations of the architectures we tested.

1. The architecture from (Cerri et al., 2014). To process text, we have added a Text Vectorizer layer from Keras. The classifiers are MLP layers with one hidden layer and an output layer with a number of neurons appropriate to the number of labels to predict at each level.
2. The same classifiers as before, but the text embedder is a pre-trained BERT (Devlin et al., 2018) from HuggingFace. We take the hidden representation of BERT's output [CLS] token as text embedding, which can be interpreted as a vector representation of the whole input sequence for classification, as suggested in (Rogge, 2020).
3. Using the same classifier structure as model 1, we implemented skip connections to pass, as an additional input to each MLP layer, the vectorized input sequence. This approach aimed at reinforcing the initial input to each MLP block, expecting it would provide more context and improve the prediction of finer-grained values.
4. The skip connection architecture with the Text Vectorizer followed by a trainable embedding layer. Each classifier module is a CNN-based network consisting of 32 convolution filters with kernel size 3×3 and stride 1. Subsequently, max-pooling and flattening are applied and finally the output is given by a dense layer.
5. The skip connection architecture with the Text Vectorizer followed by a trainable embedding layer. Each classifier module has a dropout layer, followed by an LSTM (Hochreiter and Schmidhuber, 1997b) and a dense output layer.

The variations with LSTMs and CNNs were inspired by the work of (Zhou et al., 2015), in which the authors employ those types of networks in the task of text classification, and by the blog post of (Camacho, 2018).

4 Data

The dataset we used to experiment with our hierarchical approaches is Webis-ArgValues-22 (Kiesel

et al., 2022). In it, 5270 total arguments were annotated by crowd workers and labeled for the values each argument resorted to.

In the following we will give a brief description of the dataset structure.

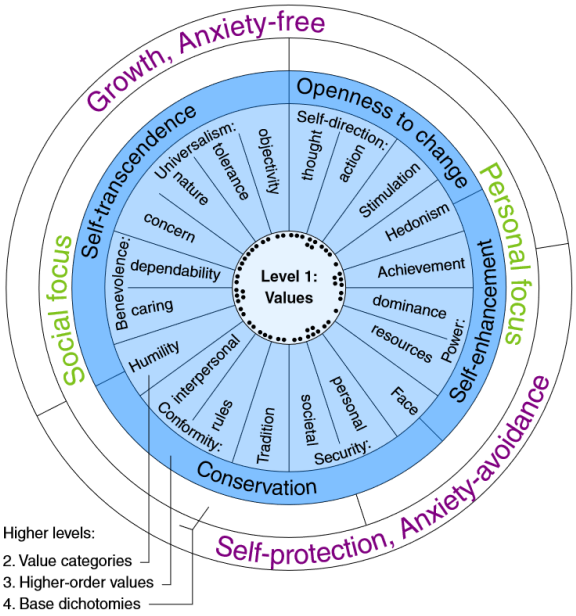

Figure 2: Schema of the taxonomy levels used by the original paper and our work. Image taken from (Kiesel et al., 2022).

4.1 Dataset structure

Each argument in the dataset contains a premise, a conclusion and a stance. The premise is a sentence in natural language that agrees with a conclusion if the stance is *"in favor"*, or goes against the conclusion if the stance is *"against"*.

For every argument, the dataset contains the annotated human value information. The label information is organized following a multi-level taxonomy. The 54 base human values are part of level 1. Then, going up in abstraction, there are 20 value categories in level 2. Level 3 contains 4 higher-order values, and the final levels 4a and 4b consist of 2 base dichotomies, as shown in Figure 2 .

Therefore, there are 82 (2+2+4+20+54) total unique labels. For each element of the taxonomy, a 1 in its column indicates that the element is present, while the value 0 indicates that it is not.

The taxonomy employed in the dataset is based on the Schwartz Value Survey (Schwartz, 2005), and was further revised by (Kiesel et al., 2022) using three other works: the Rokeach Value Survey (Rokeach, 1973), Life Values Inventory (Brown and Crace, 2002), and the Word Values Survey (C. Haerpfer and Puranen, 2020).

Refer to Table 1 for examples from the dataset.

The grey bars of Figure 3 contain information about the unbalanced label distribution for the USA test set. It can be observed that some human values appear very sparingly in the dataset, while others are more frequent, and this applies to the other sources of arguments in a similar fashion.

4.2 Different sources for arguments

The dataset has arguments from various sources: *Africa* (50 arguments), *China* (100 arguments), *India* (100 arguments), and *USA* (5020 arguments). All text contained in the dataset was originally in English or was translated to English from the original language. We highlight, as the original authors did, that no part of the dataset is intended to represent a certain culture.

Table 2 contains more information about the text data for the different parts. It can be observed that the majority of the data comes from a Western background.

4.3 Dataset preprocessing and use

We did not use the information about the conclusion and stance but focused only on the premise and the labels. Our preprocessing on the premises was very simple and consisted in:

- converting words to lowercase when using the BERT tokenizer.
- converting words to lowercase and also removing punctuation when using a text vectorizer.

As for the human values labels:

- we combined the annotated values into a single array of 0s and 1s.
- we merged level 4a and level 4b into a single level 4, as they essentially are at the same height in the taxonomy.

In the previous work (Kiesel et al., 2022), the authors decided to use the non-*USA* parts for benchmarking purposes. We did the same, by training exclusively on a subset of the *USA* arguments and testing our approach on the 4 different sources.

5 Experimental setup and results

The dataset is split into train, validation and test sets, using a function from the original work. The approaches are trained using the 4240 arguments in the training set and validated using the 277 arguments in the validation set, both include exclusively

Argument (stance on conclusion and premise)	Values (level 1)	Dataset part
In favor of "Sudan must advance its industrialization (including agro-industries)" "Industrialization increases the number of 'good' jobs - i.e. sustainable and well-paid -nationwide, and fostering the increase of skilled labour"	Be capable, Have a comfortable life, Have a stable society, Have success, Have wealth	Africa
In favor of "We should abolish the 996 overtime system" "China's 996 overtime system is very inefficient in enhancing your ability to make money. If you rely on 996 to make money, you are essentially making money by working hard, selling your body."	Have a comfortable life, Have success, Have wealth	China
Against "We can rely on Cryptocurrencies like Bitcoins" "If the value of standard currency falls, we can still afford to live in our country, because the fall of currency value impacts all fields. For example, if the prices of groceries rise, so does our salaries. But if the value of cryptocurrencies like bitcoin falls, we will lose our hard-earned money."	Have a comfortable life, Have a stable society, Have wealth	India
Against "We should ban naturopathy" "it provides a useful income for some people"	Have a comfortable life	USA

Table 1: Four example arguments from each of the parts with the corresponding annotated level 1 values. They all contain the value "Have a comfortable life".

Part	Conclusions		Premises		Stances	
	#	Tokens	#	Tokens	# Pros	# Cons
Africa	23	10.6	50	28.1	37	13
China	12	7.3	100	24.5	59	41
India	40	6.6	100	30.3	60	40
USA	71	5.6	5020	18.5	2619	2401
Total	146	5.6	5270	18.9	2775	2495

Table 2: Numbers of unique conclusions and premises for each part of the contributed dataset, their mean number of space-separated tokens, and stance distribution. Taken from (Kiesel et al., 2022).

the USA part. Then, the testing phase is performed separately on each part.

5.1 Training function and metrics

To train the models we used a training function that we implemented from scratch taking an example from (Cerri et al., 2014), in order to train each level of the model, corresponding to each level of the value hierarchy, separately. The method works by assigning each hierarchical level with a Multi Level Perceptron (MLP) network responsible for training the level. These networks are trained incrementally, where the input to each level is the result of the MLP Network associated with the previous hierarchical level.

The maximum length of the sentences given as input to the model for training is 46. This value was chosen by taking the length of the longest sequence in the training set +5. This last value is arbitrarily selected to have a bigger margin in order to avoid accidentally cutting off any values.

All the implemented models were trained using the Binary Cross Entropy loss. We also created a Weighted Binary Cross Entropy loss function, where the weights given to the loss of positive label samples are higher than those given to the negative ones. The reason for this experiment was that most human values appear in a small fraction of samples, so we wanted the model to be penalized more for assigning a low probability to a label when it was supposed to be high. We tried training the models with it, but since this version of the loss did not provide any significant benefit, we opted to use the version of the loss without weights instead.

The metric used during the training phase, to evaluate the performances of our models, was the accuracy metric, whereas for the test set we relied on the F_1-score, as it was done in the original paper.

The optimizer used was Adam, and its parameters were all set to the default values, except for the learning rate which was set to e-4 since, after several tests, we observed that the default learning rate of 0.001 was too large to provide any acceptable results.

Each model was individually trained with a structure-related number of epochs. More precisely:

Table 3: Training Hyperparameters.

Model	Skip conn.	Batch Size	Epochs per lev.
HMLP	False	32	10
BERT	False	64	5
HMLP	True	32	10
HMLP CNN	True	32	15
HMLP LSTM	True	64	5

The main differences were in the BERT and LSTM models which were trained for 5 epochs to reduce the training time. The training was very fast and did not require too many resources, except for the model containing BERT, and to a degree the one containing the LSTM layer. This difference is to be expected, since they bring more parameters to the model, making it harder to train, especially the former of the two.

After each epoch the weights were saved onto

an external file if the accuracy turned out to have improved from the previous ones, effectively saving the best weights out of all the epochs. At the end of the training, the best weights were loaded onto the model.

The number of neurons in each architecture was also model-related, seeking to create, even if each architecture was unique, a similar base performance environment to better compare them at inference time. There were used:

Table 4: Models Hyperparmeters

Baseline	**BERT**	**Baseline-Skip**	**CNN-Skip**	**LSTM-Skip**
Hidden dim.	*Hidden dim.*	*Hidden dim.*	*Filter*	*Memory Units*
50	50	[15,15,35,15]	32	200

We also noticed that changing these values did not have any significant impact on the performances.

5.2 Results

We report the results of our architectures, which were trained exclusively on the largest part of the dataset (USA) and tested on the test set, where the 33% is composed of the data from other countries, i.e India, China and Africa. An important aspect to consider is that there are differences in support of the various labels. This led to an uneven distribution of the arguments, making the dataset unbalanced. The results of the training of each level in the hierarchy are reported in the present work in two tables, much like they are presented in the original work, in order to compare the differences. In particular, in Table 5 we present the Precision, Recall and F_1-score on the part of the dataset containing only the USA arguments. Unlike the table in the original work, ours does not show the accuracy, since it does not really represent a good metric for such an unbalanced dataset. In Table 6 we present the F_1-score for each country over each level. Looking at this table in particular, it is noticeable that the USA sample set has better results than the others at all levels, which was expected since the training was performed on a subset of USA samples, so the models learned to generalize them better. It is also reasonable to hypothesize that a larger number of samples possibly helps in achieving better results, since the levels with greater support achieved better results, e.g. level 1, which has many different labels with few samples each, has very low F_1-scores across all the countries and models, while level 4 with its few levels with many samples each, achieved good results.

See Figure 3 for the F_1-scores for each label and level of the taxonomy in the USA part.

6 Discussion

Our results are not better than the ones from the original work, but at the same time they do not skew dramatically from them. The F_1-score is worse when it comes to the lower levels, while level 4, which is the highest in the hierarchy and is not affected by the other levels, has achieved the best results. They are extremely similar to the ones in the original paper which used some flat, non-hierarchical architectures. Looking at our baseline model, indicated as "Hmlp" in our Table 5, the results become worse the lower the level becomes, much like the ones in the original paper if not slightly more. Level 3 is only affected by level 4, therefore the results are still acceptable, even if not desirable. Levels 2 and 1, which are the ones that feel the effect of the hierarchical structure the most, show some unsatisfactory results. This leads us to suppose that the hierarchical structure adds a detrimental degree of complexity. Additionally, all the other 4 variations performed in extremely similar ways, indicating that the results were not really affected by the types of layers the models were made up of, as much as by the hierarchical structure itself.

Nevertheless, there could be some additional problems, specifically concerning the data. The fact that the lower levels did not achieve good results is probably not only due to the structure of the model, but also due to the small size of the dataset. The reason why level 4 performs better than the others could be that it is the level with the least number of labels, each with a higher number of samples. The lower levels have progressively more labels, and therefore less and less support for each label in the level. This leads the models to struggle to generalize the data. This very situation is reflected in the original paper as well. It could be that the models we tested do not have enough capacity to capture the more fine-grained human values because there is not enough data that can teach the model to classify those values from text, especially since those values are kind of hidden in the text itself. Another problem is that the data from different countries seem to not be completely aligned since the models do not perform on

Model	Level 1			Level 2			Level 3			Level 4		
	P	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
Hmlp	0.08	0.78	0.14	0.17	0.92	0.28	0.60	1.00	0.73	0.88	1.00	0.93
Hmlp_skip	0.08	0.66	0.13	0.18	0.84	0.28	0.60	1.00	0.73	0.88	1.00	0.93
Hmlp_skip_cnn	0.10	0.31	0.13	0.20	0.50	0.27	0.64	0.79	0.71	0.89	0.98	0.93
Hmlp_skip_lstm	0.08	0.67	0.14	0.17	0.72	0.27	0.60	0.97	0.73	0.88	1.00	0.93
Bert	0.08	0.71	0.14	0.18	0.85	0.28	0.60	0.97	0.73	0.88	1.00	0.93

Table 5: Macro precision (P), recall (R), F_1-score (F_1) on the USA test set over all labels by level.

Model	Level 1				Level 2				Level 3				Level 4			
	Afr.	**Chi.**	**Ind.**	**USA**	**Afr.**	**Chi.**	**Ind.**	**USA**	**Afr.**	**Chi.**	**Ind.**	**USA**	**Afr.**	**Chi.**	**Ind.**	**USA**
Hmlp	0.08	0.08	0.08	0.14	0.19	0.19	0.19	0.28	0.60	0.64	0.59	0.73	0.85	0.89	0.84	0.93
Hmlp_skip	0.09	0.09	0.09	0.13	0.19	0.18	0.19	0.28	0.60	0.64	0.59	0.73	0.85	0.89	0.84	0.93
Hmlp_skip_cnn	0.09	0.11	0.11	0.13	0.21	0.19	0.21	0.27	0.59	0.60	0.61	0.71	0.85	0.89	0.84	0.93
Hmlp_skip_lstm	0.09	0.09	0.09	0.14	0.20	0.19	0.19	0.27	0.59	0.64	0.55	0.73	0.82	0.88	0.76	0.93
Bert	0.09	0.09	0.09	0.14	0.20	0.19	0.19	0.28	0.59	0.61	0.60	0.73	0.85	0.89	0.84	0.93

Table 6: Macro F_1-score on each test set over all labels by level, divided by country.

the other countries as well as on the USA part, to which the subset of training data belonged to. This can be seen in Table 6.

When it comes to the training phase, we noticed that running the code for many epochs led to some overfitting. We tried to avoid it by decrementing the number of hidden dimensions like in the original paper, but that did not solve the problem. In future developments of the work, it could be useful to look into a way to reduce said overfitting, in order to train the models for longer, possibly obtaining some better results.

Before the adaptive thresholding method was applied we observed that, across all our models and datasets, the output activations for most of the labels from lower levels were always quite lower than 0.5, which would mean that without an adaptive method, the models would never predict a positive. The adaptive method, though, caused the opposite behavior for some of those labels, which would always be predicted to be present because their thresholds were set to 0, which technically maximizes the F_1 score by achieving perfect recall. The decision of thresholds for classifiers is a research area in itself, as there is no solution that fits all cases. The choice for the number of false positives and false negatives depends on the particular application. Models that always output the same answer, though, are not particularly useful.

With a bigger dataset, it is likely that this problem would not occur altogether.

This project has still a lot of room for improvement. The first thing to come to mind is to expand the data. There is a newer version of the data (Mirzakhmedova et al., 2023), that has approximately double the samples. We did not, however, use this dataset, since our goal was to implement hierarchical models with a more complex underlying structure, and this newer data only contained levels 2 and 3 which would have not made the most out of a hierarchical model. This data has the potential to be structured with level 4 as well, so it could be interesting to test it on a variation of our model with one less level.

7 Conclusion

In this work we tried five different variations of hierarchical architectures to extract human values from unstructured arguments. The results turned out to not be as good as we hoped for since none of the models were able to be confident in their positive predictions for the more fine-grained levels. This may be due to the fact that the complex nature of this task, along with a dataset that has very little supporting data for many labels, does not allow the HMLTC models to learn the innermost levels of the hierarchy. We still believe that hierarchical approaches could still be a valid option for this type of problem especially if a way to avoid overfitting was found. Further work may be to implement, for each level, one binary classifier per label in order to shift the focus to a binary classification problem, while maintaining a hierarchical approach. Moreover, it may be worth it to also concentrate on just a subset of the labels. Working with more samples in order to better train our models and finding techniques to reduce the issue of overfitting could also lead to an improvement in the results.

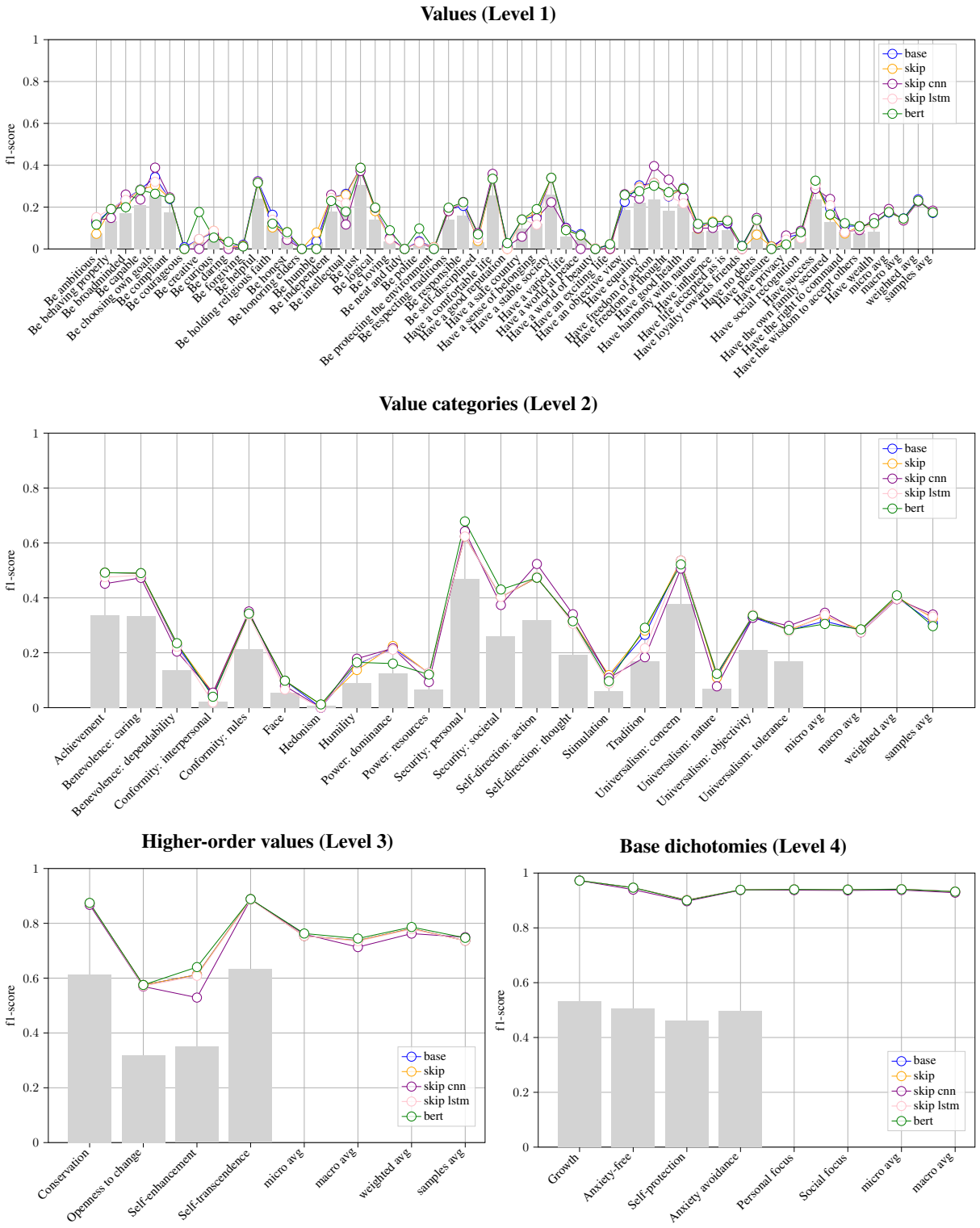

Figure 3: Parallel coordinates plot of F_1-scores on the USA test set over all labels by level. The grey bars show the label distribution, which is equal to the F_1-score of random guessing as per this distribution. (caption from (Kiesel et al., 2022)).

8 Links to external resources

The dataset and Webis-ArgValues-22 and the original paper can be found here.
The code and this report can be found in this GitHub repository.

References

Duane Brown and R. Kelly Crace. 2002. Life values inventory facilitator's guide.

Jason Brownlee. 2021. A gentle introduction to threshold-moving for imbalanced classification.

A. Moreno C. Welzel K. Kizilova Diez-Medrano J. M. Lagos P. Nor-ris E. Ponarin C. Haerpfer, R. Inglehart and B. Puranen. 2020. World values survey: Round seven - country-pooled datafile.

Cezanne Camacho. 2018. Cnns for text classification.

Ricardo Cerri, Rodrigo C. Barros, and André C.P.L.F. de Carvalho. 2014. Hierarchical multi-label classification using local neural networks. *Journal of Computer and System Sciences*, 80(1):39–56.

Boli Chen, Xin Huang, Lin Xiao, Zixin Cai, and Liping Jing. 2019. Hyperbolic interaction model for hierarchical multi-label classification. *CoRR*, abs/1905.10802.

François Chollet. 2020. Writing a training loop from scratch nbsp;: nbsp; tensorflow core.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Jibing Gong, Zhiyong Teng, Qi Teng, Hekai Zhang, Linfeng Du, Shuai Chen, Md Zakirul Alam Bhuiyan, Jianhua Li, Mingsheng Liu, and Hongyuan Ma. 2020. Hierarchical graph transformer-based deep learning model for large-scale multi-label text classification. *IEEE Access*, 8:30885–30896.

Sepp Hochreiter and Jürgen Schmidhuber. 1997a. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Sepp Hochreiter and Jürgen Schmidhuber. 1997b. Long short-term memory. *Neural computation*, 9:1735–80.

Carlos N. Silla Jr and Alex A. Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22(1-2):182–196.

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.

John Lawrence and Chris Reed. 2019. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

John Lawrence and Chris Reed. 2020. Argument Mining: A Survey. *Computational Linguistics*, 45(4):765–818.

Marco Lippi and Paolo Torroni. 2016. Argumentation mining: State of the art and emerging trends. *ACM Trans. Internet Technol.*, 16(2).

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.

Sushobhan Nayak, Raghav Ramesh, and Suril Shah. 2013. A study of multilabel text classification and the effect of label hierarchy.

Niels Rogge. 2020. Difference between cls hidden state and pooled_output · issue #7540 · huggingface/transformers.

Milton Rokeach. 1973. The nature of human values. *New York, Free Press.*

Mark Schroeder. 2021. Value Theory. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Fall 2021 edition. Metaphysics Research Lab, Stanford University.

Shalom H Schwartz. 2005. Schwartz value survey. *Journal of Cross-Cultural Psychology*.

Shalom H. Schwartz. 2012. An overview of the schwartz theory of basic values. *Online Readings in Psychology and Culture*, 2:11.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. 2018. Hierarchical multi-label classification networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5075–5084. PMLR.

Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis C. M. Lau. 2015. A c-lstm neural network for text classification.