Human Value Detection

Isabella Cadisco

July 29, 2024

1 Introduction

Aim of this project is to explore the use of Transformer based models in the setting of multi-label classification problems.
More details on the specific problem will be provided in the second section of this report, for now the premises of this project are explained.
First, let's give a formal definition of the task: in **multilabel classification**, each example is linked to a set of labels $Y \subseteq L$, where L is a finite set of labels. The objective is to learn a model f that maps each input instance to a set of labels Y. In this setting, input instances are texts and the chosen model f is a Transformer based language model.
Transformer based language models are neural language models built on the Transformer architecture, shown in the Figure 1.

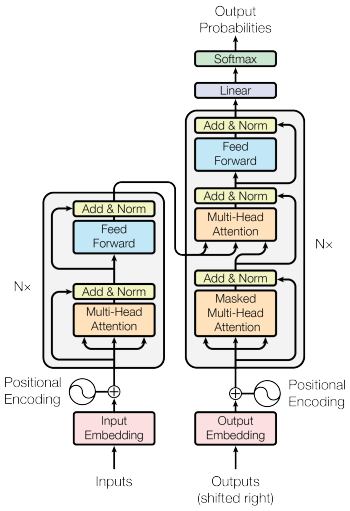

Figure 1: The Transformer - model architecture.

The complete architecture is designed for sequence to sequence tasks, especially translation, and has an **encoder-decoder structure**.

The **encoder** maps an input sequence of symbols to a vector-space representation of it. It is composed of N identical layers, each of them with two sub-layers: a **multi-head self-attention mechanism** and a **fully connected feed forward neural network**.

The **decoder** uses the encoder's representations to generate a target sequence, considering in an auto-regressive way also the tokens previously produced by itself. It is composed of N identical layers, made by three sub-layers: a **multi-head attention mechanism**, performed over the output of the encoder stack, a **self-attention sub-layer**, attending only previous positions in the decoder output, a **fully connected feed forward neural network**.

The **attention mechanism** models dependencies without regard to their distance in the input or output sequences. [1].It operates by projecting inputs into query, key, and value vectors, computing then the pairwise cosine similarity (i.e. the dot product between the two vectors) between each query and key, scaling this value and then applying a softmax to get a value between zero and one, also known as attention weight. This attention weight is then multiplied by the value vector to extract features with high attention.

The Transformer architecture has multi-headed self-attention: multiple sets of self-attention weights (or heads) are learned in parallel independently of each other. The number of attention heads included in the attention layer varies from model to model, but number in the range 12-100 are common. Each self-attention head will learn a different aspect of language (e.g. the relationship between people entities, the rhyme between words, ...). This happens during training and is not possible to dictate ahead of time what aspects of language the attention heads will learn, in fact, the weights of each head are randomly initiated and given sufficient training data and time, each will learn different aspects of language.

2 Research Question and Methodology

The multi-label classification problem that this project aims to solve is the one proposed by the **SemEval 2023 challenge**, in particular *task 4: Identification of Human Values behind Arguments*. The input sequences x are textual arguments, the target of classification is a vector $y = [0, 1]^n$, indicating absence or presence of n human values. **Human values** are defined, according to (Kiesel et al., 2022) [2] as *commonly accepted answers to why some option is desirable in the ethical sense.* The considered set of values is the one from (Schwartz et al., 2012) [3] plus nine

other values, resulting in a total of 54 human values, which are the black dots in the centre of Figure 2. Aim of this work is to identify the presence of values belonging to the second level.

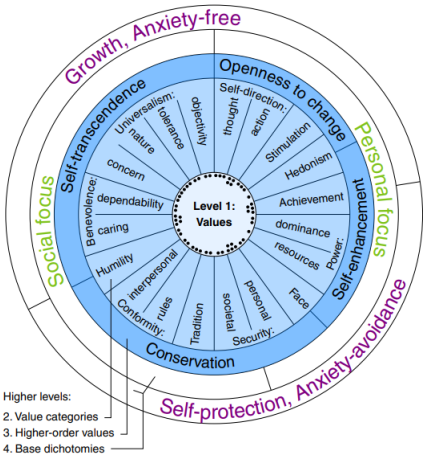

Figure 2: 54 human values (level 1), categorized in the more abstract levels 2-4. Categories that tend to conflict are placed on opposite sites.

The adopted approach for multi-label text classification is a neural model made by a **pretrained encoder**, to build an informative representation of the input text, followed by a **dropout layer**, for the purpose of **regularization** and a **linear layer**, to achieve the **classification** goal, mapping the output layer features to an n-dimensional vector, with n number of values to detect.

The pretrained encoder belongs to the **BERT** family, a family of models that uses the Transformer encoder architecture to process each input text token in the full context of all preceding and subsequent tokens, hence the name **Bidirectional Encoder Representations from Transformers** [4] In particular, two experiments were carried out relying on Hugging Face's Transformers library, the first exploiting bert-base-uncased and the second, to speed up the times using its distilled version distilbert-base-uncased.

The BERT model has been pre-trained on large text corpora through Masked language modeling (MLM) and Next Sentence Prediction(NSP); distilBERT is its distilled version thanks to three objectives: a distillation loss, to return the same probabilities as the basic BERT model; the MLM objective, a cosine embedding loss, to generate hidden states as similar as possible to the basic BERT model [5].

The reason for choosing to use distilBERT in the second experiment is that, while maintaining the same hidden size and number of self-attention heads as BERT,

768 and 12 respectively, the number of Transformer blocks is halved from 12 to 6, making fine-tuning faster.

The objective function used for fine-tuning is the **Binary Cross Entropy with Logits Loss** (BCEWithLogitsLoss). It combines a sigmoid layer and the binary cross-entropy loss into a single function, which first applies the sigmoid activation to convert raw logits into probabilities and then computes the binary cross-entropy loss, since the labels are treated as independent one from each other and not mutually exclusive.

For fine-tuning, is used the same **optimizer** BERT was originally trained with: **AdamW**. It is a stochastic gradient descent method doing regularization by weight decay. Weight decay [1]is a regularization technique that adds a penalty proportional to the square of the magnitude of the weights to the loss function, encouraging the model to learn smaller weights, corresponding to less abrupt changes in the output for a small deviation in the input [6]. In AdamW, weight decay is decoupled from the loss function optimization steps, which allows the weight decay factor to be chosen independently of the learning rate, improving generalization [7].

3 Experimental Results

3.1 Data

The dataset for Touché / SemEval 2023 Task 4. ValueEval: Identification of Human Values behind Arguments is based on the original Webis-ArgValues-22. For this project, only *train* and *test* set were used, since no hyperparameters optimization is performed. These two sets have respectively cardinality 5393 and 1576. For each set two tab separated files are provided: one containing the textual arguments; one containing dummy columns indicating the presence or absence of values. Each line represents an argument and it is identified by a unique ID. Thanks to this ID is possible to merge the two sets.

Each textual argument is composed by:

- **Conclusion**: the statement that establishes the context (e.g. *We should ban human cloning*).
- **Premise**: relies on values to argue for or against the conclusion (e.g. *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**: position of the premise towards the conclusion (e.g. *in favor of*).

[1]For Stochastic Gradient Descent methods, l_2 regularization and weight decay coincide.

The purpose of this paper is to identify the presence of second-level values of the taxonomy presented in Figure 2. However, for simplicity, as suggested in the task description, only the six most frequent values in the dataset will be considered, namely: *Self-direction: action*, *Achievement*, *Security: personal*, *Security: societal*, *Benevolence: caring*, *Universalism: concern*.

As can be seen in Figure 3, the distribution of labels in train and test sets is quite similar. Nevertheless, it can be observed that the labels are slightly imbalanced. For completeness, the distribution of the validation set is also shown.

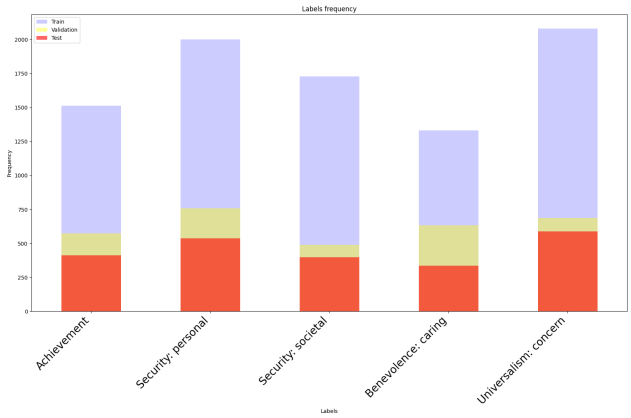

Figure 3: Labels distribution

3.2 Metrics

The evaluation of the model was determined by a collection of different metrics.

The **accuracy** will evaluate the number of correct labels over the total number of labels, however this may be misleading in a multi-label setting, where it considers the **exact match** between actual and predicted set of labels for each sample.

To overcome this, **Hamming loss** was also used, taking into account for **partial match** while computing the fraction of incorrectly predicted labels [8].

The **F1 score**, which is the harmonic mean of precision and recall, was also taken into account, thus providing a single score that allows the trade-off between the two. Precision measures the percentage of correctly predicted positive instances out of all predicted positive instances, while recall measures the percentage of correctly predicted positive instances out of all actual positive instances. This score is considered in two versions: **micro**, precision and recall are calculated globally by counting the total of true positives, false negatives and false positives, taking into account label imbalances; **macro**, the metrics are calculated for each

label and then aggregated into the mean. In (Kiesel et al., 2022) [2] this latter version of the F1 score is used, given their specific choice to give the same weight to all label values.

3.3 Value detection from premise only

As reported in (Kiesel et al., 2022) [2], the values are related specifically to the premise of the topic; this first experiment will therefore use only this part to detect them.

The approach is the one described in Section 2, with bert-base-uncased as pre-trained encoder. The dropout probability is set to 30%, the learning rate is $2e-5$, the same as (Kiesel et al., 2022) [2], the weight decay is set to 0.1, the batch size to 32 and the model was trained for 10 epochs. The presence of a label value is detected if its corresponding probability is greater or equal to 0.5.

The aggregate metrics calculated for each training epoch indicate that the model is **overfitting**, as can be seen in Figure 4, particularly after the second epoch.

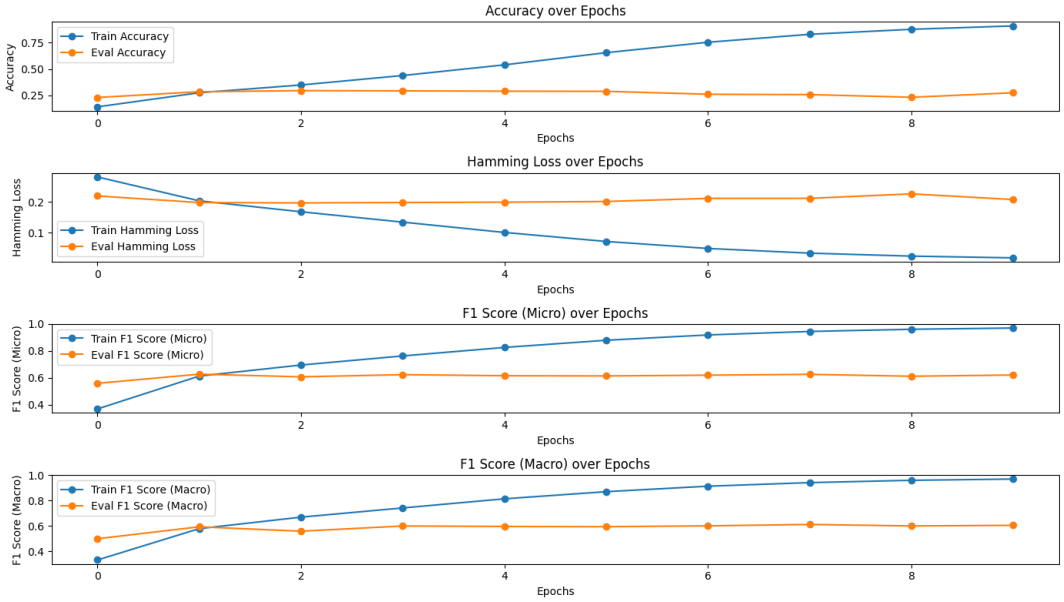

Figure 4: Metrics across training epochs

However, when considering the individual label values, it can be seen that the most represented values in the train dataset, *Security: personal* and *Universalism: concern*: concern,' are identified with an f1 score of about 0.7, as shown in Figure 5.

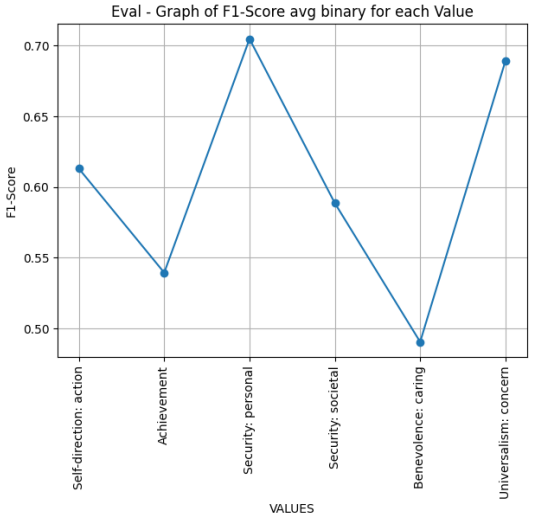

Figure 5: Evaluation F1 per single label

3.4 Value detection from full argument

As reported in (Kiesel et al., 2022) [2], automatic models may improve when they incorporate conclusion and position as context to the premise; this second experiment will therefore use full text to detect human values. The experimental setup is the same as in the previous experiment, except that the pre-trained encoder, which, to speed up finetuning, is distilbert-base-uncased.

Evaluation metrics are reported in in Figures 6 and 7, aggregate per epoch and w.r.t. single labels at the end of training. As can be seen, the behavior of the model does not differ much from the previous experiment.

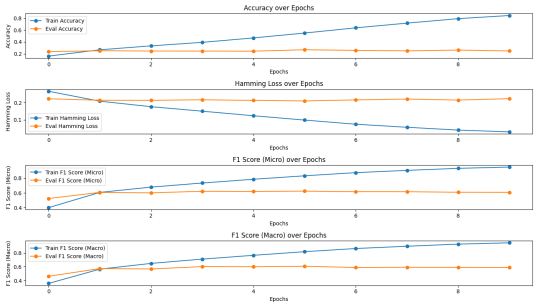

Figure 6: Metrics across training epochs

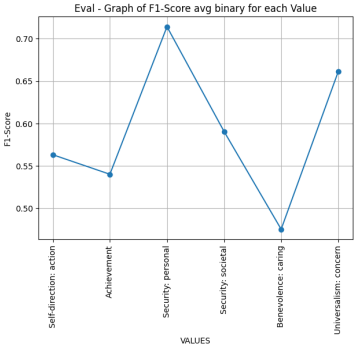

Figure 7: Evaluation F1 per single label

4 Conclusion

Although there are clear signs of overfitting, the results on the most represented label values are encouraging and suggest that more training data for the other values would be helpful in improving the model's abilities without necessarily having to reduce its complexity. Last but not least, hyperparameter optimization was not performed in these experiments, which could bring further improvements, for example, reducing the batch size could improve generalization.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[2] J. Kiesel, M. Alshomary, N. Handke, X. Cai, H. Wachsmuth, and B. Stein, "Identifying the Human Values behind Arguments," in *60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)* (S. Muresan, P. Nakov, and A. Villavicencio, eds.), pp. 4459–4471, Association for Computational Linguistics, May 2022.

[3] S. H. Schwartz, J. Cieciuch, M. Vecchione, E. Davidov, R. Fischer, C. Beierlein, A. Ramos, M. Verkasalo, J.-E. Lönnqvist, K. Demirutku, *et al.*, "Refining the theory of basic individual values.," *Journal of personality and social psychology*, vol. 103, no. 4, p. 663, 2012.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[5] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020.

[6] S. Scardapane, "Alice's adventures in a differentiable wonderland – volume i, a tour of the land," 2024.

[7] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019.

[8] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1–13, 2007.

2

[2]I declare that this material, which I/We now submit for assessment, is entirely my/our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my/our work. I/We understand that plagiarism,

collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me/us or any other person for assessment on this or any other course of study.