Identifying the human values behind arguments

Paul Fuhr

Abstract

This project studies different models to classify values behind human arguments. Extending the work of [1], we're augmenting their dataset by paraphrasing each textual argument using the Pegasus paraphraser from HuggingFace. We're evaluating different models trained on the initial dataset and, respectively or, on the augmented dataset. The main goal is to repeat the notions of the Information Retrieval class and explore additional machine learning techniques.

1 Introduction

We're working upon the work of [1], which provided the following definition of a value. A value is a (1) belief (2) pertaining to desirable end states or modes of conduct, that (3) transcends specific situations, (4) guides selection or evaluation of behavior, people, and events, and (5) is ordered by importance relative to other values to form a system of value priorities." [1] This is better understood by the following example, also given in [1] . "Social media is good for us. Though it might make people less polite, it makes our lives much easier." The definition is matched for the value of "having a comfortable life": The belief (1), that a comfortable life is desirable (2) in general (3), evaluates behavior, e.g in the example it selects use of social media as good behavior. Moreover "having a comfortable life" is prioritized over other values, e.g here over "being polite" (5) inside the value system of the person making the statement.

The authors of [1] provide (a) "a multi-level taxonomy of 54 human values taken from four authoritative cross-cultural science studies and (b) a dataset consisting of 5270 arguments from the US, Africa, China and India each of which manually annotated with value labels." The 54 values are then grouped into 20 value categories. In order to compare different approaches to identifying the human values behind arguments, the authors of [1] created the competition SemEval Task 4.

The task was to map textual arguments to vectors $\mathbf{x} \in \{0,1\}^{20}$, indicating the absence or presence of a human value category.

2 Goals of the project

The first goal of the project is to explore different methods to solve the task using machine learning and contribute to the effort of identifying the human values behind arguments. The second goal of the project is to repeat some of the contents of the information retrieval class, explore new machine learning techniques and practice python. (As I am a mathematics student who mainly coded with C++ before, this is my first bigger python project.)

3 Proposed methodology

We will try 2 different model architectures to solve the task.

3.1 BOW Model

The input sequences are being represented by a binary vector using the Bag of words approach. This representation is passed to a binary classifier, namely a simple neural network with one hidden layer followed by a ReLU activation. To classify 20 categories we use 20 of these binary classifiers which are trained sequentially. I shall abbreviate this model with 20BC.

3.2 Bert Model

The bert-base-uncased transformer of the huggingface library creates a rich representation of the input sequence. [2] This representation is forwarded through a linear classifier. I shall abbreviate this model with TC.

4 Experimental results

4.1 Datasets

4.1.1 First Dataset

I used the dataset (arguments.tsv) provided on the information retrieval website consisting of 5270 textual arguments with corresponding labels. The first 4517 arguments coming from the USA were used for training (80%) and validation (5%) and the remaining 753 arguments ranging from the USA, China and Africa were used for evaluation (15%) of the models performance.

4.1.2 Second Dataset

Because the dataset is small for a complex task, we created the second dataset paraphrases.tsv by paraphrasing the premise and conclusion of each argument in the dataset arguments.tsv using the Pegasus Paraphraser from the huggingface library. For example "algorithmic trading allows for more timely, accurate, and efficient trading." was paraphrased to "More timely, accurate, and efficient trading can be achieved with the use of algorithmic trading." In the following, we shall use this convention: Every model for which I don't specify the training dataset is trained on the first dataset.

”On augmented data” means that the model is trained on the training part of the first dataset united with the first 4517 paraphrases.

4.2 First Experiment

As a first experiment, we trained the three models 20BC, TC and 20BCoad [1] on their respective datasets with the hyperparameters listed in Table 1. We used early stopping when the loss on the validation set rose for patience := 3 consecutive epochs.

Table 1 Hyperparameters

Models	optimizer	learning rate	patience[1]	loss function	hidden units
20BC, 20BCoad	SGD[2]	0.1	3	CrossEntropyLoss	128
TC	SGD[2]	0.05[3]	3	BCEwithLogitsLoss	None

[1]early stopping parameter

[2]We also played a bit with Adam, but in the tests SGD had better results.

[3]I planned to use learning rate = 0.1 for the Transformer Classifier as well, but this led to fast convergence to a constant function effectively classifying every argument as the zero vector.

4.2.1 Results

During the experiment we noted first that the dataset is biased towards 0 on each category. As can be seen in Figure 1, in many categories the proportion of positive labels is lower than 0.1. It is reasonable to fear that this absence of positive labels will lead to poor model performance. As can be seen in Figure 2, on categories 3, 4, 6, 8, 11, 12, 13, 14, 16, 18, 19, 20 the model 20BC is acting very similar to the zero map with a few quite random positive predictions, because it has a low percentage of positive predictions and low recall. For TC and 20BCoad one observes similar behavior, though the effect is a little smaller for the transformer classifier.

We further looked at the F1-score[2], as a trade-off between precision and recall for each category separately. In Figure 3, one can observe the following: Firstly, the biased dataset coincides with poor performance in the above-mentioned categories. Secondly, 20BCoad is not performing better across all categories than 20BC, although it was trained on a larger Dataset. For example, on categories 2, 5, 9 and 10 it is performing worse, whereas on categories 6, 7, 8, 11, 12, 18 and 19, it is performing better.

Overall however, 20BC achieves a slight higher F1-score than 20BCoad, while TC performs significantly better than both models as can be seen in Table 2.

4.3 Second Experiment

The F1-scores of TC on some categories (2, 5, 7, 9, 10, 15, 17) were above 0.4, so looked quite promising. Thus, I was interested in further optimizing the performance of a transformer based model on a single value category. I adapted the architecture of

[1]this means 20 BC trained on augmented data

[2]If the F1-score is undefined (precision undefined or (precision = 0 and recall = 0), it shall be defined as 0.

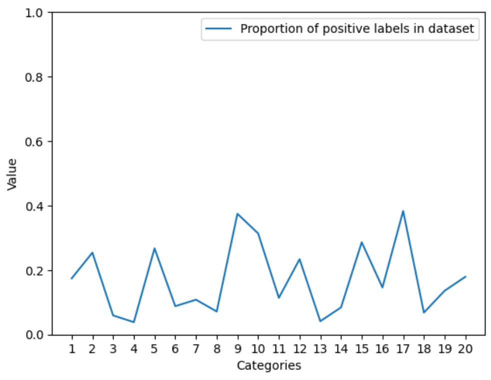

Fig. 1 Proportion of positive labels

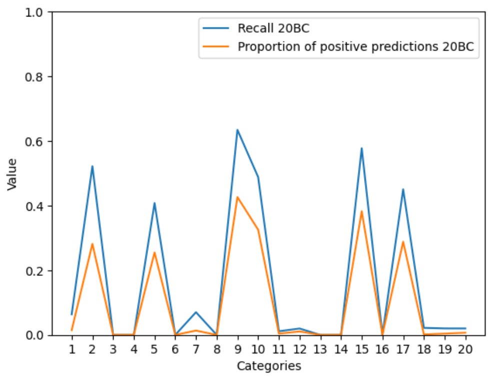

Fig. 2 20 BC

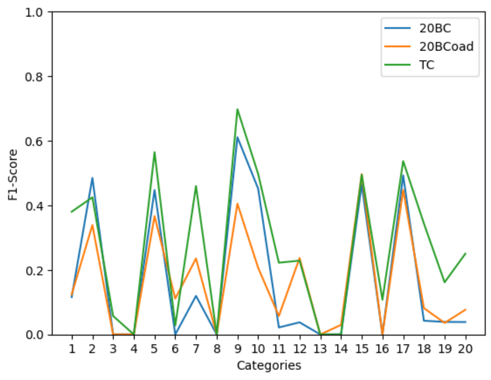

Fig. 3 F1-Scores

Table 2 F1-scores

Model	F1 score overall
20BC	0.17
20BCoad	0.16
TC	0.27

the model, adding one hidden layer with a modifiable number of hidden units followed by a ReLU activation. Working with just one category, the output dimension was changed to 2. Let us thus call this model TC2. As the results of Experiment 1 (4.2) were the most promising for category 9, I only worked on category 9 in the following. I tried out the following 2 ideas:

4.3.1 Hyperparameter Optimization

Working with fixed optimizer SGD and loss function CrossEntropyLoss, TC2 had the modifiable hyperparameters: learning rate, patience and hidden units. According to the RandomSearch Algorithm, we proceeded as follows: From the hyperparameter searchspace defined in Table 3, we retrieved 20 random hyperparameters. We then trained the model [3] and evaluated each configuration of hyperparameters using F1-score on the evaluation dataset. We obtained the hyperparameters listed in Table 4 as the best configuration with an F1-score of 0.705. This score is also a tiny bit higher than the F1-score of TC on category 9 (0.697).

Table 3 Hyperparameter Searchspace

Parameter	Set of values
hidden size	$64, 128, 192, 256$
learning rate	$0.001, 0.006, 0.011, ...0.046$
patience	$1, 2, 3, 4$

Table 4 Best configuration of hyperparameters

Parameter	hidden size	learning rate	patience
Value	64	0.016	4

4.3.2 1:1-ratio

Based on the assumption, that for a binary classification a 1:1-ratio of positive to negative labels is optimal, I extended the dataset with randomly chosen paraphrases with label 1 to create this exact 1:1-ratio. I quickly abandoned this approach, because

[3] with early stopping

I wanted to look at Hyperparameter optimization as well. However, the results I obtained by playing a bit with the model were not better than the ones obtained by the TC on category 9 in the first experiment.

5 Conclusion, Future Work and Ideas

We can be confident that the work contributes to the effort of understanding how to detect human values in arguments.

We can be content with the F1-score of 0.7 received on category 9 in section 4.3.1 as the dataset was quite small, but not very biased. I believe that given a bigger and relatively unbiased dataset, it should be possible to achieve a higher score such as 0.9 and higher. Obviously this is not the case for all the categories. I hypothesize that some categories might just be ”more difficult to learn”. A possible approach to estimate this difficulty could be to determine the quality of the available data for a given category with similar, but more advanced methods than used in section 4.2.1 and compare it to the performance of different models on that category.

Furthermore, it surprised me that 20BCoad did not perform better than 20BC. I would be curious to understand why: Did it overfit because of the bigger training dataset? Is the size of the dataset really so relevant, or have other parameters such as the type of encoding and complexity of the model a more determinatory effect? Maybe the use of the augmented data does not improve F1-score, but other metrics, such as either Precision or Recall?

Moreover, It would interest me to continue with the hyperparameter optimization begun in section 4.3.1, extending the search space by e.g including other optimizers, such as Adam and trying other more sophisticated hyperparameter optimization algorithms such as Bayesian Optimization and evaluation methods such as k-cross-validation. Hyperparameter variation could also help with finding hyperparameters that work better with the augmented dataset.

Lastly, my mom works in Pharmacovigilance at a Pharma company. In the company, actual people have to review snippets of text for a mention of side effects, a very similar problem to the detection of values in arguments. I would be interested in finding a dataset and creating and optimizing a model to help solve this task.

References

[1] Johannes Kiesel, N.H.X.C.H.W. Milad Alshomary, Stein, B.: Identifying the human values behind arguments. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics **1: Long Papers**, 4459–4471 (2022) https://doi.org/10.18653/v1/2022.acl-long.306

[2] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)