Assignment 2

Edoardo Saturno, Riccardo Marvasi, Lucia Gasperini Arianna Albertazzi
Master's Degree in Artificial Intelligence, University of Bologna
{ riccardo.marvasi, edoardo.saturno, lucia.gasperini, arianna.albertazzi3 }@studio.unibo.it

Abstract

This report presents an analysis of performance for models involved in a text classification task. It specifically focuses on human values in natural language arguments (Johannes Kiesel, 2022). The objective is to develop and train an architecture for a multi-label multi-class classification task. The approach involves the use of word Embeddings obtained from BERT architectures. This kind of representation of the data will be used as input data for four different classification heads, one for each of the four considered categories. The entire architecture performance will then be evaluated.

1 Introduction

In recent years, the performance of various models designed for classifying argumentative text into categories has achieved significant milestones. Transformer-based models, exemplified by BERT (Jacob Devlin, 2019), are particularly notable for their prowess in bidirectional context understanding and semantic relationship capture, making them suitable for many NLP tasks. Support Vector Machines (SVM) (M.A. Hearst, 1998) aim to discover optimal hyperplanes for class separation but can be computationally expensive, especially when dealing with large Datasets. Other strategies, in cases of limited computational resources, involves the use of simpler models that implement Logistic Regression, although they assume linear relationships, thus limiting their expressiveness compared to more complex models. This study specifically concentrates on Pre-trained BERT models utilizing the "bert-base-uncased" model (Victor Sanh, 2020). The task involves employing distinct binary classification heads for each category, where the input of the classification heads will be the semantically rich bert embeddings.

We want to perform identification of four different macro categories of "human values", defined as *openness to change*, *self-enhancement*, *conservation*, and *self-transcendence*. In order to perform a robust analysis we used three different seeds for initialization of the Bert models. After training, at evaluation time, we looked at the average performance on the three seeds.

2 System description

In this section the sequential steps used for implementing the strategy will be shown. Initially, the data was extracted from the given files, with an additional step of combining lower-order categories into macro categories. The next step involved organizing the newly structured data into a Pandas Dataframe, which will include the true labels corresponding to the macro categories *Openness to change*, *Self-enhancement*, *Conservation*, and *Self-transcendence*. Subsequently, preprocessing was performed by converting the data to lowercase to improve tokenization and, in order to reduce computational effort, a length analysis was conducted on the data: truncation was applied based on an empirical criterion, limiting the number of words. Following data preparation, the models were created. They included a Random Uniform Classifier, Majority Classifier, Binary Linear Classifier, and Bert Models. The Bert models were defined based on the input they received at training time (C, CP, CPS). Embeddings for the input data were obtained for all Bert models for all the random seeds, and these embeddings were fed into the classification heads of each Bert model obtaining what we will call "Bert-based" models. The performance on the validation set was then compared with that of the majority and random classifiers. In the end, the study involved comparing the performance of Bert-based models with respect to other Bert and the baselines. This comprehensive analysis aimed to provide insights into the effectiveness and variability of Bert-based models in classifying the macro categories under different conditions.

3 Experimental setup and results

The observed disparities in performance primarily arose from the input kind employed for embedding generation. BertC models exclusively utilize the tokenized form of conclusions while BertCPs employ the tokenized concatenation of premises and conclusions. As the name suggests BertCPS models additionally concatenate Stance values to the tokenized CP. Also we systematically explored variations in learning rate, batch size, and the number of epochs during hyperparameters tuning. Ultimately, the F1 score was adopted as the evaluation metric. For which regards the specific strategies we tried to use we can highlights the fact that in order to reduce Embeddings dimension, truncation was implemented after a specified word count for both Conclusion and Premise sequences. Notably, empirical analysis revealed that 99 percentage of the data has a length below 20 for conclusions and 90 percentage below 36 for premises. Consequently, a maximum sequence length of 20 for C, 56 for CP, and 57 for CPS was introduced. This strategy effectively prevents utilising up to 200 padding tokens for each sequence based on the maximum sentence length that could be found in the data before applying truncation. We also used *SGD optimizer* in the training of the classifiers. In the end the following hyperparameters values were used : *200* epochs and *0.003* as learning rate. Result can be observed in Table 1.

4 Discussion

The obtained results reveal a consistent improvement in performance as the complexity of the input provided to the models increases. BertC demonstrates sometimes superiority over the baseline classifiers in certain classes, but falls short on achieving this across all classes. It can notably be surpassed by the majority classifier in the most frequent class and by the random classifier in the less represented one. BertCP exhibits greater resilience, consistently outperforming the classifiers in nearly all cases, with only occasional exceptions when compared to the majority classifier in the most frequent class. A comparable, but better, pattern of performance is observed for CPS, maintaining effectiveness across various classes. Notably, the CPS model emerges as the top-performing model, owing to the richness of information encapsulated in its Embeddings. The inclusion of Stance adds valuable insights into the relationship between Premise and Conclusion, fostering a more comprehensive understanding of the entire sentence from the model's perspective and resulting in a more balanced classification. A trend that can be observed is that models receiving simpler inputs tend to exhibit a "majority classifier" behavior, wherein a substantial number of examples are classified into the most frequent classes. Contrarily, models handling more complex inputs, such as CP and CPS, strongly reduce this behaviour. An example of that can be observed for C models: it is evident that they are biased towards the majority classes, outperforming CPS. However, when examining the less frequent classes, a significant drop in performance, as high as 0.4, is observed. Looking ahead, future investigations could explore the implications of Downsampling on model performance, particularly in addressing challenges related to less represented categories.

5 Conclusion

Summarizing, *CP* and *CPS* models demonstrate good performance with robust Macro F1 scores, highlighting their balanced proficiency. On the other hand *BertC* tends to slightly mimic a majority classifier. As anticipated, the richness of the input significantly influences the performance of the corresponding models. One of the main limitations is associated with the maximum sequence length of the sentences. Specifically, there is a varying maximum length for each set (train, test, val), and a common value must be chosen among the three. In this case, the minimum of these three values is selected.

Category	**Bert C2**	**Bert CP1**	**Bert CPS3**	**Random Uniform Classifier**	**Majority Classifier**
Openness to Change	0.173	0.498	0.585	0.413	0
Self-Enhancement	0.548	0.587	0.619	0.449	0
Conservation	0.838	0.839	0.835	0.592	0.838
Self-Transcendence	0.874	0.871	0.864	0.612	0.874
Macro F1	0.551	0.699	0.726	0.516	0.428

Table 1: F1 scores for each category and macro F1 scores for the seed with the highest performance for each model.

References

Kenton Lee Kristina Toutanova Jacob Devlin, Ming-Wei Chang. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Nicolas Handke Xiaoni Cai Henning Wachsmuth Benno SteinJohannes Kiesel Milad Alshomary Nicolas Handke Xiaoni Cai Henning Wachsmuth Benno Stein Johannes Kiesel, Milad Alshomary. 2022. Identifying the human values behind arguments. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

E. Osuna J. Platt B. Scholkopf M.A. Hearst, S.T. Dumais. 1998. Support vector machines. *IEEE Intelligent Systems and their Applications*.

Julien Chaumond Thomas Wolf Victor Sanh, Lysandre Debut. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *EMC*^{2} : $5thEditionCo - locatedwithNeurIPS'19$.