

Classification of Myers–Briggs Type Indicator personality types using Natural Language Processing

Andor Kiss, Dóra Bányai, Milán Kriston, and Zoltán Kádár

Eötvös Loránd University

1 Literature search

As indicated in papers [?] [?] the state of the art models for text classification are transformer based architectures. Our idea was to use an LSTM-based baseline model and four different pretrained transformer architectures downloaded from huggingface, namely Generative Pre-trained Transformer 2 (GPT-2) [?] BERT, RoBERTa and Albert. As the article about GPT-2 [?] shows, the architecture was pretrained in a self-supervised way to understand the language first, and later finetuned for specific tasks, which is the reason why we used it in this project, leveraging the potential of it to model the English language, and providing it with inputs and labels for our specific classification task. We are aware of the fact that GPT-2 is not the current state of the art model in OpenAI's series of language models, however due to computational costs, we did not want to use a too large model with several billion parameters like GPT-3 [?] or GPT-NEOX [?]. GPT-2 small was used with its 124 million parameters.

2 Individual contributions

2.1 Andor Kiss - TXC54G

- Team leader tasks - git repo, weekly report to supervisor, Google Docs, LaTeX template
- Literature search
- Data exploration
- Data pipeline
- GPT-2 training and evaluation

2.2 Dóra Bányai - NEPTUN

- Literature search
- Data exploration
- RoBERTa

2.3 Milán Kriston - NEPTUN

- Literature search
- Data pipeline
- BERT

2.4 Zoltán Kádár - OTO3RC

- Literature search
- Data exploration
- Albert model training and evaluation
- (LSTM-based baseline model)

3 Results

Table 1. Evaluation results

Model	Accuracy	F1 score	Precision	Recall	Execution speed
GPT-2@cat@100	0.549	0.543	0.55	0.549	0.0467
GPT-2@bin@100	0.524	0.516	0.540	0.524	0.0324
GPT-2@cat@250	0.70	0.70	0.70	0.70	0.0479
GPT-2@bin@250	0.69	0.6889	0.6945	0.69	0.0577
GPT-2@cat@500	0.837	0.838	0.84	0.837	0.067
GPT-2@bin@500	0.82	0.819	0.821	0.82	0.0831
Albert-v2@cat@100	0.544	0.538	0.550	0.544	0.0146
Albert-v2@bin@100	TBD	TBD	TBD	TBD	TBD
Albert-v2@cat@250	0.681	0.678	0.691	0.681	0.0136
Albert-v2@bin@250	TBD	TBD	TBD	TBD	TBD
Albert-v2@cat@500	0.888	0.888	0.889	0.888	0.0132
Albert-v2@bin@500	0.841	0.802	0.803	0.804	0.0142

In Table 1 cat represents the model having 16 different output possibilities corresponding to the 16 personality types with softmax output activation, bin represents the model having 4 binary classifiers as the output layer predicting each character in the MBTI type, and the number (100, 250, 500) represents the maximum sequence length the model was trained with. The execution speed column indicates how quickly a batch is processed by the model in seconds. It was measured on an NVIDIA P100 GPU during inference.

In paper [?] the models had 4 binary classifiers as output, thus showing the metrics separately for each character in the MBTI type, however since in this project we used both categorical and binary outputs, a fairer comparison is done by converting the binary outputs to categorical with some post-processing steps and calculating the error using that format.

As the table shows, the sequence length of 500 was the best performing by far, while the categorical outputs outperformed the binary outputs by a small

margin. Compared to the difference in performance, the difference between the models in terms of execution speed was negligible.

4 Conclusion and Future work

4.1 Conclusion

TODO - write something about the difficulties during the project

The main difficulty with these transformer models are the training time and model size. So base models were used, which is the least parameter heavy. With the lack of appropriate hardware, used the 30 hours weekly training time of Kaggle and Geforce 1070 Ti GTX locally with 8 gigs of RAM. In some cases the batch size needed to be downscaled to 4 as the RAM for local training was not enough to serve the models.

Albert model difficulties - Even though all of the transformer models used are conforming with the HuggingFace transformer library, there are small differences which can make comparison and the ability to use a common base code difficult. For example GPT-2 doesn't have an activation function at the end of the transformer model, but Albert had a tanh based one, so the same code implemented in the binary classifier in the torch module needed to be adjusted. Luckily there were options. These small differences made a lot of hours extra troubleshooting and documentation reading necessary.

Baseline model difficulties - We tried to use a very simple baseline model, it suffered similar issues as Albert, but on a higher degree. HuggingFace has a well defined. As the main task described to work on cutting edge, it could be discarded as transformer based models focused on.

4.2 Future work

TODO