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 [5] [7] the state of the art models for text classification are transformer based architectures. Our idea was to use several different pretrained transformer architectures downloaded from HuggingFace, namely Generative Pre-trained Transformer 2 (GPT-2) [6] Bidirectional Encoder Representations from Transformers (BERT) [8], A Robustly Optimized BERT Pretraining Approach (RoBERTa) [4] and A Lite BERT for Self-supervised Learning of Language Representations (Albert) [3].

As the article about GPT-2 [6] 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 [2] or GPT-NEOX [1]. GPT-2 small was used with its 124 million parameters.

BERT was introduced by Google AI Language in [8]. In 2019, it was the best performing model in 11 NLP tasks. Just as GPT-2, BERT is also pretrained in a self-supervised manner while alleviating the usual LSTM based models' unidirectional constraint, by using Masked Language Modeling. Bert consists of 12 transformer blocks, 12 attention heads, and 110 million parameters, which is even less than the parameter amount of GPT-2.

RoBERTa [4] is a variant of the BERT model developed by researchers at Facebook AI. Similar to BERT, RoBERTa is a transformer-based model that is trained to perform natural language processing tasks such as language translation and text classification. However, RoBERTa was designed to be more robust and efficient than BERT by making several improvements to the original model. RoBERTa was trained on a much larger dataset than BERT, which included a combination of publicly available datasets and internal data from Facebook. This larger training dataset allowed RoBERTa to learn more about the structure and patterns of natural language, leading to improved performance on a variety of tasks.

Albert [3] was published by members of the Google Research team, similar to BERT. Usually increasing the parameter space in language models result in improved performance, however the computational costs increase with it as well. With Albert, the goal was to introduce parameter reduction techniques to lower the size and compute requirements of a BERT-like language model while not performing worse on downstream NLP tasks. The Albert model we used had 12 million parameters, almost 90% less than the original BERT.

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 - W5H8NT

- Literature search
- Data exploration
- RoBERTa training and evaluation

2.3 Milán Kriston - Z3M7ZI

- Literature search
- Data pipeline
- BERT training and evaluation

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
BERT-2@cat@500	0.778	0.779	0.783	0.778	0.161
BERT-2@bin@500	0.620	0.615	0.628	0.69	0.620
RoBERTa@cat@500	0.799	0.8	0.817	0.799	0.161
RoBERTa@bin@500	0.674	0.671	0.686	0.674	0.620
Albert-v2@cat@100	0.544	0.538	0.550	0.544	0.0146
Albert-v2@cat@250	0.681	0.678	0.691	0.681	0.136
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 [7] 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. As for the BERT and RoBERTa models, we didn't feel the need to experiment with the sequence lengths, since it was proven with GPT-2 and Albert that models with the length of 500 perform the best. As Google Colaboratory doesn't share what kind of GPU it is using, inference times are hard to compare. In the case of BERT, the performance between the binary and categorical output models was more significant than with GPT-2. The reason for such big differences could be further investigated in the future. It is also worth mentioning that because of the limited resources, the BERT models were only trained for 4 epochs. RoBERTa performed slightly better than BERT in both binary and categorical output cases. This is possible because RoBERTa was trained using a different optimization procedure than BERT, which involved using a dynamic learning rate scheduler and larger batch sizes, allowing it to learn more quickly and efficiently.

4 Conclusion and Future work

4.1 Conclusion

The project was exciting where we gained a lot of knowledge regarding state of the art NLP models, their architectures and their applications, and we applied this newly acquired knowledge to an interesting practical task.

For every model, we used Kaggle, however the biggest difficulty creating the solution with the BERT model, is that HuggingFace's transformer library was not working properly with that particular model, so it had to be trained using Google Colaboratory. Another significant obstacle was that nearly all models that we were using were quite computationally heavy, with hundreds of millions of parameters. Approximately a single epoch took 2 hours (depending on model and environment) which made the training phase slow. To alleviate this inconvenience with the training, for every architecture, the base models were used, which are the least parameter heavy. Because of the lack of appropriate hardware, we had to use the 30 hours free weekly training time provided by Kaggle and a local NVIDIA GeForce GTX 1070 Ti with 8GB of VRAM. In some cases the batch size needed to be downscaled to 4 as the memory of the local training environment was not enough to serve the models.

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 has a tanh based one, which made adjusting the architecture for a common training/testing loop a bit more inconvenient. We also tried to use a very simple LSTM-based baseline model, it suffered similar issues as Albert, but on a higher degree. Since HuggingFace has a well defined API and as the main task was to work with state of the art models, the baseline idea was discarded so we can focus more on the transformers.

4.2 Future work

Every model, especially the BERT could be further trained. With the BERT model the losses were still decreasing after 4 epochs, but the compute cost of Google Colaboratory made it hard to train. Furthermore every model could be enhanced with more hyper-parameter tuning.

4.3 Git repository

The source code of this project can be found here: git repository

References

1. Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., He, H., Leahy, C., McDonell, K., Phang, J., Pieler, M., Prashanth, U.S., Purohit, S., Reynolds, L., Tow, J., Wang, B., Weinbach, S.: Gpt-neox-20b: An open-source autoregressive language model (2022). https://doi.org/10.48550/ARXIV.2204.06745, https://arxiv.org/abs/2204.06745
2. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners (2020). https://doi.org/10.48550/ARXIV.2005.14165, https://arxiv.org/abs/2005.14165
3. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: A lite bert for self-supervised learning of language representations (2019). https://doi.org/10.48550/ARXIV.1909.11942, https://arxiv.org/abs/1909.11942
4. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach (2019). https://doi.org/10.48550/ARXIV.1907.11692, https://arxiv.org/abs/1907.11692
5. Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., Gao, J.: Deep learning based text classification: A comprehensive review (2020). https://doi.org/10.48550/ARXIV.2004.03705, https://arxiv.org/abs/2004.03705
6. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019)
7. dos Santos, V., Paraboni, I.: Myers-briggs personality classification from social media text using pre-trained language models. JUCS - Journal of Universal Computer Science **28**(4), 378–395 (apr 2022). https://doi.org/10.3897/jucs.70941, https://doi.org/10.3897%2Fjucs.70941
8. Toutanova, J.D.M.W.C.K.L.K.: Google ai language: Bert: Pre-training of deep bidirectional transformers for language understanding (2022)