Efficiency of Bagging Ensemble Method on Modern Transformers

Brian Murtagh
Pontificia Universidad Católica de Chile
bmurtagh@uc.cl

Pablo Brancoli
Pontificia Universidad Católica de Chile
pbrancoli@uc.cl

Lucas Rodríguez
Pontificia Universidad Católica de Chile
ljrodriguez1@uc.cl

Abstract

"Two heads are better than one", this old saying expresses the idea that governs the famous "ensemble methods" in machine learning. In general terms, these methods are based on the hypothesis that the combination of several models can produce a new, much more powerful model.
On the other hand, during the last years, the Natural Language Processing study area has been shocked by the arrival of the innovative transformers, these deep learning models have the advantage that they take all their inputs completely, unlike the old recurrent neuronal networks, which depended on taking inputs sequentially. This is a great contribution to the world of NLP because the transformers allow greater parallelization, reducing training times and improving results.
During this study we sought to mix these two ideas, ensemble models and transformers, for this first an analysis was made of how different transformers behaved in different datasets, in order to choose the most suitable transformer for the ensemble.
Once chosen, an analysis was made of how this transformer behaved in different ensemble architectures. Surprisingly, the results were not as expected, and it was concluded that the transformers, being so complicated, work better on their own than mixing them together.

Keywords

machine learning, transformers, NLP, ensemble methods

ACM Reference Format:
Brian Murtagh, Pablo Brancoli, and Lucas Rodríguez. 2021. Efficiency of Bagging Ensemble Method on Modern Transformers. In *Deep Learning: Term Project, June 16, 2022, Santiago, Chile.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1234567.1234567

1 State of the art

The objective of this research is to provide a general understanding of the performance that different modern transformers can have on various SuperGLUE tasks, as well as to propose an architecture that improves the results through the "bagging" ensemble methodology. Therefore, it is necessary to establish a basic understanding of these elements.

Transformers are deep learning models based on a self-attention mechanism, where each part of the data has the same importance, unlike recurrent networks, which follow a sequential process. Its main application is the field of natural language processing, and this is the reason why we use it to analyze the SuperGLUE dataset. The main difference they have with recurrent neural networks is that they process the data inputs sequentially, while the transformers process all the input at once, thus giving the same importance to each data and also allowing a greater parallelization reducing training times.

There has been multiple papers testing transformers on GLUE and SuperGLUE dataset[3,8,9], proving the increasing persective of transformers in NLP. As SuperGLUE is a new standard benchmark in the field, we proposed inspecting different transformers for finding the most suitable candidate for a dual bagging ensamble and testing if it improves the accuracy of the transformer alone.

In the field we can find similar approaches, such as in the work of Julian Risch and Ralf Krestel[10] where they developed a Bagging Bert for a more robust aggression detection, the difference is they joined up to 15 models of BERT having positive results in accuracy but more important a major boost in the stability of accuracy, reflecting promising results by mixing transformers.

In other investigations[11], also Boosting methods have been tested for Bert in the GLUE dataset, the investigation which was based on passing the output through several BERT transformers, which shared weights. All this with the purpose of enhancing the original BERT, where they came to the conclusion that their own model reached a higher performance than that of the original BERT.

1.1 Models

- BERT[3]: Is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers.As a result, the pre-trained BERT model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial tasks specific architecture modifications.

- RoBERTa[4]: Its a extension of BERT, it modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates.

RecSys, 2022-1, Santiago, Chile

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1234567.1234567

- DeBERTa[5]: Its model architecture that improves BERT and RoBERTa using two novel techniques. The first is the disentangled attention mechanism, where each word is represented using two vectors that encode its content and position. The second is an enhanced mask decoder is used to replace the output softmax layer to predict the masked tokens for model pretraining.

- AlBERT[6]: Its a Lite version of BERT, reducing the memory consumption and increasing the training speed, with two parameter-reduction techniques.

  Due to hardware limitation all models were used with their base pre-trained format and not bigger formats available, which greatly impacts transformer performance and decreasing this research accuracy below baseline benchmarks[8,9]

1.2 Bagging[7]

With the purpose of improving the results of the models, an investigation of the bagging ensemble method was made, bagging its a technique that helps to improve the performance and accuracy using the predictions of different models to get a better one. It is used to deal with bias-variance trade-offs and reduces the variance of a prediction model. Bagging avoids overfitting of data and is used for both regression and classification models, specifically for decision tree algorithms.

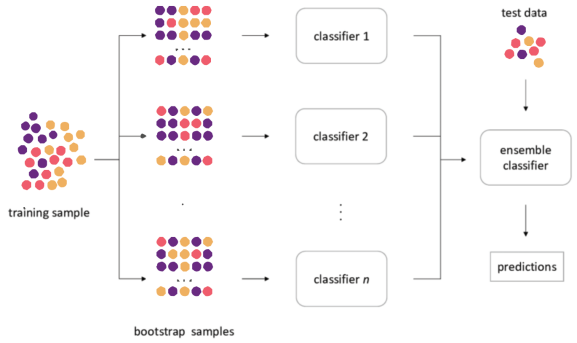

Figure 1: Bagging Ensemble Method

1.3 Ensembling Strategy[7]

The motivation of the ensembling is to research the performance of dual transformers and how its accuracy varies under different aggregate algorithms. In this research there are 3 ensembles developed, were all three include a dual RoBERTa transformers that receive the tokenized string of a respective SuperGLUE task, the encoders output is later given to a aggregate algorithm to combine the two outputs into a final answer. The different aggregate algorithms are as follow:

- BI-LSTM: This method concatenates the last hidden states of the two RoBERTa, and then passes them through a BI-LSTM layer, which transforms the output from 1 dimension to 80, after which a flatten layer is applied to prepare the results for the linear layer. of 245760 x 2, finally a softmax log function is applied.

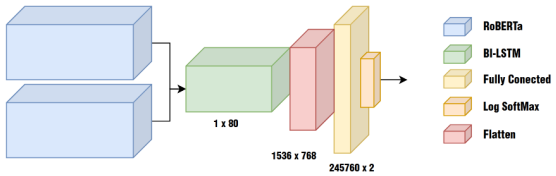

Figure 2: BI-LSTM Architecture

- Fully Connected: This method concatenates the last hidden states of the two RoBERTa, to then pass them through a linear layer of 1536x700, the output of this layer is normalized, and then go through a linear layer again this time of 700x2, to finally apply a softmax log function.

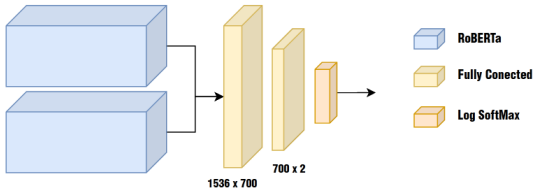

Figure 3: Fully Connected Architecture

- Max Out: This method simply consists of a 4x2 linear layer where it receives the outputs of the two models and concatenates them to form 4 values, these are passed to the linear layer and return an output of dimension 2, simulating No/Yes.

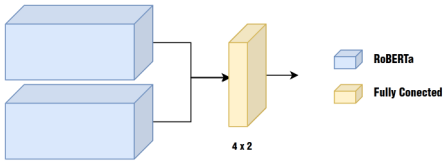

Figure 4: Max Out Architecture

2 SuperGLUE Dataset[1]

We based our investigation on the SuperGLUE dataset, that is a extension of GLUE Benchmark, which offers a single number metric that summarizes progress on a diverse set of tasks in the Natural Language Processing area.

2.1 BoolQ

Is a Question-Answering dataset, where each example consists in a premise, a question about that premise and an answer yes/no. The source is google search engine, where users search a question, and the premise is paired with a Wikipedia article.
Premise: Bullsnake – Bullsnakes are very powerful constrictors who eat small mammals, such as mice, moles, rats, pocket gophers, ground squirrels, and rabbits, as well as ground nesting birds, birds' eggs and lizards. Their climbing proficiency enables them to raid bird nests (and birdhouses) to eat the nestlings or sitting mother. One snake can eat five small birds within 15 minutes. Juvenile bullsnakes depend on small lizards, frogs, and baby mice.
Question: can a bull snake kill a small dog
Answer: Yes

2.2 CB

CommitmentBank, is a corpus of short texts in which at least one sentence contains an embedded clause. Each of these embedded clauses is annotated with the degree to which it appears the person who wrote the text is committed to the truth of the clause.
Premise: Polly had to think quickly. They were still close enough to shore for him to return her to the police if she admitted she was not an experienced ocean sailor.
Hypothesis: Polly was not an experienced ocean sailor
Answer: Yes

2.3 RTE

Recognizing Textual Entailment, is a two class classification task, entailment and no entailment, given a text and a hypothesis, the task must determine if the entailment is true or not.
Premise: Authorities in Brazil say that more than 200 people are being held hostage in a prison in the country's remote, Amazonian-jungle state of Rondonia.
Hypothesis: Authorities in Brazil hold 200 people as hostage.
Entailment: False

3 Research Methodology

The methodology used to achieve the objectives of this research was, in general terms, to search and read recent documents and online resources, then select transformers, which we would analyze with different datasets and hyperparameters, finally, with the model that gave the best result, we analyzed how it behaved with the proposed bagging architecture.

The articles consulted served as a means to learn about the state of the art and the different transformers. This stage of the research also included searching for implemented code online, mainly in repositories published by authors of research in the field. This helped us define the objective of our research and guide us to be able to implement our own bagging model.

After defining the transformers and datasets, the first implementation was carried out. First of all, an analysis was developed for each model in each dataset, with a batch size of 8 and a learning rate of 1e-5, all this in order to see which model is the best and which dataset is more suitable for it. From this it was concluded that the best model was DeBERTa, but there was a problem of hardware limitations, because DeBERTa is a very heavy model. When trying to implement the ensemble architecture, when occupying several models, the RAM filled its memory quickly, using AlBERT we had the same problem, so the decision was made to use RoBERTa as the model to be analyzed, since it was the rest with the best results.

We continued with an analysis of hyperparameters for RoBERTa, we specifically analyzed how the results changed with different batch size and learning rates, all this in order to get the best hyperparameters to finally implement the ensemble architecture. After verifying that the best result was with learning rate 1e-5 and batch size 8, these hyperparameters were chosen to continue with the ensemble.

The ensemble consisted of training two Roberta models in parallel, where for the output of each one, a last layer was added that decided which output to take according to the values delivered from each model. The last layers options were a Fully Connected layer, a BI-LSTM layer, and a function that takes the max output. Finally we save the accuracy results of each last layer in order to have a final architecture.

This process allowed an exhaustive analysis of the training parameters and their impact on the results, as well as an overview of the different transformers and their ability to face changes. It also allowed how the transformers behaved against an ensemble architecture such as bagging.

3.1 Metric[1]

To measure all the different models and their configurations, we have the benefit that Super GLUE summarizes all the metrics in one, the accuracy, for this we test the model and check how many predictions were correct, and divide it by the total number of predictions, represented by P. Let L be the correct labels for every example, then the equation it's as follows:

$$Accuracy = \sum_{i=1}^{|P|} \frac{P_i = L_i}{|P|}$$

4 Parameter Analysis

For the purpose of this research, we did not focused on the optimization of the transformers configuration as their setting have already been deeply analyzed by large groups of researchers, rather the research is focused on training parameters to analyze the impact of them on the accuracy

- Learning Rate: It determines the step size in each iteration of the training, while moving toward a loss function. It determines the speed at which the machine "learns", too small values may result in a long training process that could get stuck, whereas a value too large may result in learning a sub-optimal set of weights too fast or an unstable training

process.

- Batch Size: Depicts the number of samples that propagate through the neural network before updating the model parameters. Each batch of samples goes through one full forward and one full backward propagation. When the batch size is big, lesser is the noise in the gradients and so better is the gradient estimate, however it may require a lot of memory consumption.

- Epochs: Is a hyperparameter of gradient descent that controls the number of complete passes through the training dataset. When its big the training completes more passes so it more "trained" but you need to be care full, too many epochs may cause over fitting.

5 Results

The experiments described in section 3 resulted in a significant amount of configurations and scenarios. All of them were measured using Accuracy, and the results were condensed in a Table 1,2,3:

Model Analysis on different datasets

From Table 1 we can draw two conclusions, the model with the best accuracy and the dataset that gives the best results. Regarding the models, we can analyze that Roberta and DeBERTa were the ones with the best results, being the best DeBERTa, which gave us the best result in BoolQ with an accuracy of 85%, this is because they are models based on BERT, which were extended and modified with the purpose of having better results than the original BERT. On the other hand, On the other hand, for the same reasons, we can see that BERT is the one with the worst results, because the other models are improvements of this one.

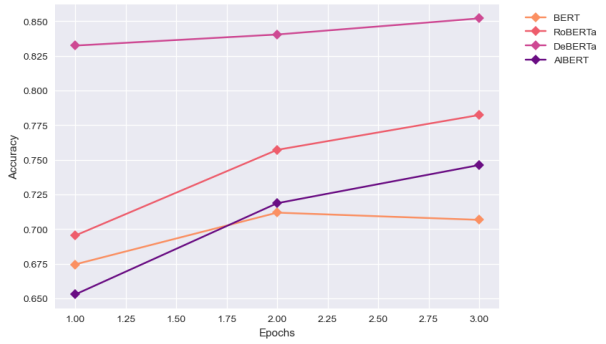

Figure 5: Models accuracy in BoolQ Dataset

We can analyze that in most of the models, except for AlBERT, the dataset with the best results was BoolQ, with a maximum of 85%. This is due to the fact that it occupies the simplest methodology, where by means of a Google Query it obtains the first result of Wikipedia, a web page which has millions of data, for which it will be much easier for the result of the question to be correct, since the other datasets are much more complex, CB is based on a embedded clause so for the model is much more difficult to check if a hypothesis is correct or not, plus RTE is based in two task classification so its also much more complex than BoolQ.

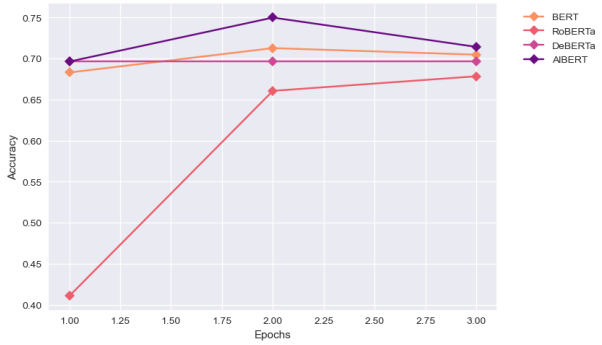

Figure 6: Models accuracy in CB Dataset

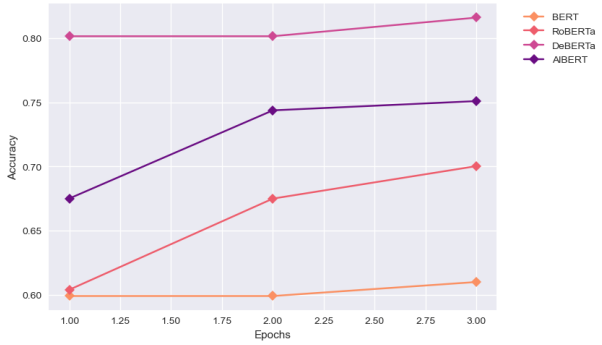

Figure 7: Models accuracy in RTE Dataset

Hyperparameter Analysis for RoBERTa

In table 2 can be analyzed several things, first of all for the learning rate 1e-5 t can be observed that after epoch 4 it begins to suffer from overfitting, because the accuracy begins to drop, so the model is being overtrained.

On the other hand, it can be observed that for a learning rate of 1e-3, the accuracy does not change, this may be due to the fact that the learning rate is quite high, which causes instability in the training process.

Finally, it is observed that the best result is with a batch size of 8 and a learning rate of 5, so these are the hyperparameters that will be used from now on.

Ensemble Architecture Analysis

Table 1: Models Accuracy

Epoch	Model											
	BERT			RoBERTa			DeBERTa			AlBERT		
	Boolq	CB	RTE	Boolq	CB	RTE	Boolq	CB	RTE	Boolq	CB	RTE
1	0,6746	0,6831	0,5992	0,6954	0,4107	0,6041	0,8324	0,6964	0,8014	0,6529	0,6964	0,6751
2	0,7119	0,7128	0,5992	0,7572	0,6607	0,6750	0,8404	0,6964	0,8014	0,7187	0,7500	0,7437
3	0,7067	0,7048	0,6101	0,7823	0,6785	0,7003	0,8520	0,6964	0,8159	0,7462	0,7143	0,7509

Table 2: Hyperparameter Analysis for RoBERTa

Batch Size	Epoch	Learning Rate	
		1e-5	1e-3
8	1	0.693578	0.621713
	2	0.766055	0.621713
	3	0.791437	0.621713
	4	0.802752	0.621713
	5	0.798777	0.621713
16	1	0.716820	0.621713
	2	0.781651	0.621713
	3	0.791743	0.621713
	4	0.802141	0.621713
	5	0.796942	0.621713

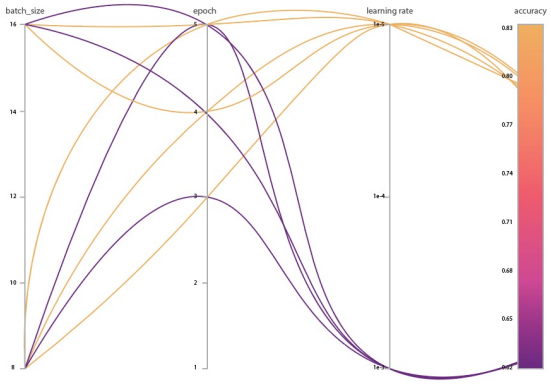

Figure 8: Hyperparameter Analysis for RoBERTa

From table 3 it can be seen that the results were not as expected, the best architecture was that of the Plain column, that is, the RoBERTa model without any modification, so the ensemble methods did not serve to give a better result. This is because the layer that combined the results of each model was probably not very good, and perhaps it would require pretraining.

On the other hand, now only analyzing the ensembles, we can see that the best layer was the Max-Output layer, this could be for the same reason that the ensembles did not work, because Max-Output being just a function, it was the only layer that didn't require pretraining. This fact that they do not require training can also be seen in how their accuracy increases through the epochs, unlike the other models, where it increases much more slowly.

Table 3: Ensemble Architecture Analysis

	Ensemble Architecture			
Epoch	Max-Output	BI-LSTM	FullyConnected	Plain
1	0,6216	0,6216	0,5957	0,6954
2	0,6656	0,6216	0,6155	0,7572
3	0,7179	0,6394	0,6100	0,7823

6 Conclusions

From the results of the experiment, it can be observed that different transformers affect the datasets differently, as well as the results depend a lot on the chosen dataset. It was expected that as the epochs increased, the accuracy would be higher, a statement which was incorrect, since the transformers quickly began to suffer from overfitting when they reached 3-4 epochs of training.

As it could be observed, the best transformer was DeBerta, reaching up to 85% accuracy in the BoolQ dataset, this falls on the nature of the model, which is defined in the same paper[5] as "a model that improves BERT and Roberta" using two novel techniques. Unfortunately, due to the limitations of the study's resources, this model could not be used to investigate how it behaved with ensembles. This is due to the fact that, as it is based on other versions of BERT, it is a much heavier model, and it required large amounts of data. of RAM to be able to be trained.

It can be concluded that the learning rate is an essential hyperparameter when training a model, when looking at the results it can be seen how a very large learning rate quickly generates instability in the training since many steps are skipped, which is why a lot of steps are omitted. information.

The performance of the ensembles with the transformers was not very successful, reaching worse performance than the ensembles by themselves, this could be due to the nature of the transformers that by treating all the inputs as one, they become much more complex systems than those normally used, and layers with BI-LSTM require many more times of training than transformers, so it could be interesting in the future to investigate how ensembles behave with pre-trained combination layers.

Finally, we would like to comment on the relevance of the study of the various challenges that arise from the area of NLP. This area

is constantly growing and new models come out every day, each one more intelligent than the other, in their day they were the transformers and today they are models that they even think they are self-aware such as LaMDA and GPT-3. These systems could play a fundamental role in the coming years both in the world of technology, and in politics or the economy, and if we are not careful they could even become a problem in the future.

7 References

(1) Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., ... Bowman, S. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. Advances in neural information processing systems, 32.

(2) Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

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

(4) Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

(5) He, P., Liu, X., Gao, J., Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654.

(6) Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. (2019). Albert: A lite Bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.

(7) Rakotoson, L., Letaillieur, C., Massip, S., Laleye, F. (2021). BagBERT: BERT-based bagging-stacking for multi-topic classification. arXiv preprint arXiv:2111.05808.

(8) Yi Tay, Zhe Zhao, Dara Bahri, Donald Metzler, Da-Cheng Juan. (2021). Hypergrid Transformers: Towards A Single Model For Multiple Tasks. ICLR.

(9) Jin Y., Hassan H. (2020) FastFormers: Highly Efficient Transformer Models for Natural Language. arXiv:2010.13382v1.

(10) Risch J., Krestel R. (2020). Bagging BERT Models for Robust Aggression Identification. LREC.

(11) Huang, T., She, Q., Zhang, J. (2020). BoostingBERT: Integrating multi-class boosting into BERT for NLP tasks. arXiv preprint arXiv:2009.05959.