ghostwriter19 @ ATE_ABSITA: Zero-Shot and ONNX to Speed up BERT on Sentiment Analysis Tasks at EVALITA 2020

Mauro Bennici
You Are My Guide
Torino
mauro@youaremyguide.com

Abstract[1]

English. With the arrival of BERT[2] in 2018, NLP research has taken a significant step forward. However, the necessary computing power has grown accordingly. Various distillation and optimization systems have been adopted but are costly in terms of cost-benefit ratio. The most important improvements are obtained by creating increasingly complex models with more layers and parameters.

In this research, we will see how, by mixing transfer learning, zero-shot learning, and ONNX runtime[3], we can access the power of BERT right now, optimizing time and resources, achieving noticeable results on day one.

Italiano. Con l'arrivo di BERT nel 2018, la ricerca nel campo dell'NLP ha fatto un notevole passo in avanti. La potenza di calcolo necessaria però è cresciuta di conseguenza. Diversi sistemi di distillazione e di ottimizzazione sono stati adottati ma risultano onerosi in termini di rapporto costo benefici. I vantaggi di maggior rilievo si ottengono creando modelli sempre più complessi con un maggior numero di layers e di parametri.

In questa ricerca vedremo come mixando transfer learning, zero-shot learning e ONNX runtime si può accedere alla potenza di BERT da subito, ottimizzando tempo e risorse, raggiungendo risultati apprezzabili al day one.

1 Introduction

In a process with data that change very quickly and the need to resort to complete training in the shortest possible time, transfer learning techniques have made possible a fast fine-tuning of BERT models. The distillation of a model made it possible to decrease the load and the times without significantly losing accuracy. These models, therefore, require, at least, constant fine-tuning training. In addition, a BERT model specially designed for the Italian language and with a vocabulary containing technical terms increases its effectiveness.

Constant and multi-disciplinary training requires specific skills and tailor-made services. In this research, we will see an effective way to make both things possible. The idea is to use a way to exchange AI models between library and frameworks, the ONNX project, and a runtime, the ONNX runtime project, to optimize inference for many platforms, languages and hardware. The ONNX runtime is still working to optimize the training directly in the ONNX format.
The second goal is to find a viable alternative with acceptable performance at the start of a new project while waiting for a trained BERT model.

The research was carried out for the ATE ABSITA (de Mattei et al., 2020) task in the EVALITA 2020 (Basile et al., 2020), using all 3 available sub-tasks.

2 Description of the system

To start using a sentiment analysis system, we need several elements. Certainly, a starting dataset

[2] https://github.com/google-research/bert
[3] https://microsoft.github.io/onnxruntime/

with the related labels. In the tasks of the challenge, we have the reviews of 23 different products. Each review has a corresponding rating assigned by the end-user. For each review, it was required to extract the aspects contained in it. By aspect, we mean every opinion word that expresses a sentiment polarity. Finally, each aspect was classified as a pair of values: positive or negative, for 4 possible states.
Imagine a system that receives an unspecified number of reviews in real-time with new products and different categories. We find ourselves in the situation of always having to fine-tune our models.
The complexity of BERT makes training time difficult for constant alignment. Being able to reduce the training time, or being able to put in place an alternative in the meantime, new perspectives open up, such as:

- Made inference calls before a full trained model is completed.
- Training of the new model.
- Running the BERT model.
- (optional) reclassify recent product reviews after the model update.

In this perspective, in order to validate my hypotheses, I used the AlBERTo (Polignano at al., 2019) model, used in the baseline, and Ktrain[4], a wrapper for TensorFlow[5], with the autofit option.

The first submission, called ghostwriter19_a, was obtained training all the models with the Ktrain framework.
The results for the three tasks for the second submission, called ghostwriter19_b, were obtained in two different way:

- for the first two tasks, I used the model of the first submission but exported on ONNX and ran with the ONNX runtime.
- for the third task, I trained the model with TensorFlow using a Zero-Shot learner [ZSL] (Brown et al, 2020).

To test the models, I used two different machines with Ubuntu 20.04 LTS:

- 6 vCPU on Intel Xeon E5-2690 v4 - 112GB with P100 (GPU)
- 14 cores on Intel Xeon E5-2690 v4 - 32GB (CPU)

2.1 Task 1 – ATE: Aspect Term Extraction

To identify an aspect, the dataset contains a label for every single word with three possible values:

- B for Begin of an aspect.
- I for Inside an aspect.
- Or for Outside, not in an aspect.

For example, the review "*La borraccia termica svolge egregiamente il proprio compito di mantenere la temperatura, calda o fretta che sia. La costruzione è ottimale e ben rifinita. Acquisto straconsigliato!"* is labeled as:

La	borraccia	termica	svolge	egregiamente	il	proprio
O	O	O	O	O	O	O

compito	di	mantenere	la	temperatura,	*calda*	o
O	O	B	I	I	O	O

fretta	che	sia.	La	costruzione	è	ottimale
O	O	O	O	B	O	O

e	ben	rifinita.	Acquisto	straconsigliato!
O	O	O	O	O

The model will be evaluated with the F1-score. The score results from the full matched aspects, the partial matched ones, and the missed ones.

The preliminary results with the Ktrain model were encouraging (table 1).

Model	**F1-Score**
ghostwriter19_a	0.6152
Baseline	0.2556

Table 1: Task 1 DEV results

At this point, the model has been exported with ONNX in maximum compatibility mode. The model ran with the ONNX runtime optimized for CPU.

The performances have remained unchanged, but the speed of inference has significantly improved (table 2).

[4] https://github.com/amaiya/ktrain

[5] https://www.tensorflow.org/

Model	Query per second
ghostwriter19_a CPU	4
ghostwriter19_b CPU with ONNX runtime	68
ghostwriter19_a GPU	124
ghostwriter19_b GPU with ONNX runtime	217

Table 2: Performance comparison on Task 1

The improvement is 17x for the CPU version and 1.75x for the GPU version.

2.2 Task 2 – ABSA: Aspect-based Sentiment Analysis

For this task, the aspects identified in Task 1 have been used. This implies that an error in Task 1 will have a decisive impact on Task number 2.

The aspect can be classified as:

- positive (POS:true,NEG:false)
- negative (POS:false,NEG:true)
- mixed polarity(POS:true, NEG:true)
- neutral polarity (POS:false, NEG:false)

As showed to the image from the challenge website[6]:

Aspect terms	Positive	Negative
mantenere la temperatura	1	0
costruzione	1	0

The results on the DEV test outperform the baseline (table 3).

Model	F1-Score
ghostwriter19_a	0.6019
Baseline	0.2

Table 3: Task 2 DEV results

Also, for this task, the performance is improved with the use of ONNX runtime (table 4).

[6] http://www.di.uniba.it/~swap/ate_absita/task.html

Model	Query per second
ghostwriter19_a CPU	3
ghostwriter19_b CPU with ONNX runtime	56
ghostwriter19_a GPU	97
ghostwriter19_b GPU with ONNX runtime	154

Table 4: Performance comparison on Task 2

The improvement is 9.5x for the CPU version and 1.59x for the GPU version.

2.3 Task 3 – SA: Sentiment Analysis

Task 3 is a classification problem. However, fully understanding the score is not easy. The evaluation operation is carried out by different people and with different styles. A product with a similar review is rated according to the expectations and judgment of other users differently.

Furthermore, in order to obviate the long training time that a constant updating requires, compared with systems used by the previous version of EVALITA, such as an ensemble system with Tree Random Forest and Bi-LSTM (Bennici and Portocarrero, 2018) or with an SVM system (Barbieri et al., 2016), I used a Zero-Shot Learner [ZSL] (Pushp & Srivastava, 2017). A ZSL is a way to make predictions without prior training (Petroni, 2019). ZSL will refer to the embedding of a previous matrix, AlBERTo in this case, and of the proposed labels as a possible result (Schick and Schütze, 2020).
The proposed labels were the possible numbers for evaluation, then the numbers from 1 to 5.

The proposed prediction value is a weighted average of the two values with the highest probability, if and only if the gap between the two values is less than 10^{-3}. Otherwise, only the value with the highest probability will be considered valid.

For this task, I omitted the ONNX runtime test because a stable converter for the ZSL version is not available.

The score for this task is the Root Mean Squared Error between the polarity predicted and the polarity assigned by the user.

Model	RMSE score
ghostwriter19_a	0.6997
ghostwriter19_b	0.8526
Baseline AlBERTo	1.0806

Table 5: Task 3 DEV results

The loss in performance is 18%, but the entire previous training phase is skipped (table 5).

3 Results

The results obtained with the DEV dataset are very positive both in terms of accuracy and performance. ZSL has proven to be an incredible technology to invest in. The Ktrain seems to suffer a heavy overfit.

The research aims not to have a relevant model but to prove that a model could be production-ready with fewer resources and time.
However, in all three tasks, the models outperformed the baseline with a significant gap in terms of accuracy/RMSE.

3.1 Results for Task 1

The final results with the TEST dataset are:

Model	F1 score
ghostwriter19_a_D	0.6152
ghostwriter19_a_T	0.5399
Baseline AlBERTo	0.2556

Table 6: TEST dataset results for Task 1

The results are about 12% lower than those obtained in the research phase (table 6).
It will be interesting to continue experimenting with different ONNX options to find a better combination of compatibility and performance.

3.2 Results for Task 2

The final results with the TEST dataset are:

Model	F1 score
ghostwriter19_a_D	0.6019
ghostwriter19_b_T	0.4994
Baseline AlBERTo	0.2

Table 7: TEST dataset results for Task 2

The loss from DEV to TEST is about 17% (table 7). However, the percentage of the difference between the results of Tasks 1 and 2 have been maintained with the DEV and TEST datasets.

This is in line with expectations, worse model performance in Task 1 impacted Task 2 proportionally. In return, working on a better model will improve both tasks.

3.3 Results for Task 3

For Task 3 we have:

Model	RMSE score
ghostwriter19_a_D	0.6997
ghostwriter19_b_D	0.8526
ghostwriter19_a_T	0.81394
ghostwriter19_b_T	0.83479
Baseline AlBERTo	1.0806

Table 8: TEST dataset results for Task 3

The difference between the DEV and TEST datasets is marked here only for trained model, 14% (table 8). The untrained one performed slightly better, 2%, with the TEST dataset.

This result confirms that an underperforming model has the same performance of a model that use ZSL, as assumed.

The price to pay, however, is that the average inference time for the ZSL is 157x higher than the pure TensorFlow model obtained with Ktrain.

4 Conclusion

The results demonstrated that it is possible to create hybrid systems for training and inference to make the power of BERT more accessible.

In the time it takes to train a new and optimized model, an untrained ZSL model can make up for it in the meantime.

Optimizing, and in future training, our models to be intrinsically optimized for the platform and framework we have chosen to use does not affect performance and future use.

The improvements obtained in the use of ONNX runtime for these Italian tasks are in line with what Microsoft demonstrated, for the English language, at the beginning of 2020 (Ning at al., 2020).

The next step is to make the ONNX export work with a Zero-Shot learner [ZSL] in order to compensate, at least in part, for the more significant resources that this inevitably introduces.

References

Barbieri, F., Basile, V., Croce, D., Nissim, M., Novielli, N., & Patti, V. (2016). Overview of the Evalita 2016 SENTIment POLarity Classification Task. In *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*. CEUR-WS.org.

Basile, V., Croce, D., Di Maro, M., & Passaro, L. (2020). EVALITA 2020: Overview of the 7th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. In *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, CEUR-WS.org.

Bennici, M., & Portocarrero, X. S. (2018). Ensemble for aspect-based sentiment analysis. In Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso, editors, *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18)*. CEUR-WS.org.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., . . . Amodei, D. (2020, July 22). Language Models are Few-Shot Learners. https://arxiv.org/abs/2005.14165

de Mattei, L., de Martino, G., Iovine, A., Miaschi, A., Polignano, M., & Rambelli, G. (2020). Overview of the EVALITA 2020 Aspect Term Extraction and Aspect-based Sentiment Analysis (ATE_ABSITA) Task. In *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*, CEUR-WS.org.

Ning, E., Yan, N., Zhu, J., & Li, J. (2020, January 31). Microsoft open sources breakthrough optimizations for transformer inference on GPU and CPU. https://cloudblogs.microsoft.com/open-source/2020/01/21/microsoft-onnx-open-source-optimizations-transformer-inference-gpu-cpu/

Polignano, M., Basile, P., de Gemmis, M., Semeraro, G., & Basile, V. (2019). Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets. In *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019)*. CEUR-WS.org.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., & Riedel, S. (2019, September 04). Language Models as Knowledge Bases? https://arxiv.org/abs/1909.01066

Pushp, P. K., & Srivastava, M. M. (2017, December 23). Train Once, Test Anywhere: Zero-Shot Learning for Text Classification. https://arxiv.org/abs/1712.05972

Schick, T., & Schütze, H. (2020, April 27). Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference. https://arxiv.org/abs/2001.07676