
Elevating Sentiment Analysis with ONNX Runtime: A Success Story

**Description**

# Introduction

In the rapidly evolving world of ecommerce, understanding customer sentiment is paramount for businesses seeking to improve their products and services. With the explosion of online reviews, manually analyzing customer feedback has become daunting. However, thanks to cutting-edge technologies like Large Language Models (LLMs), we can now leverage the power of natural language processing (NLP) to extract insights from customer reviews. This blog post will discuss how using ONNX Runtime made our opinion mining system much faster. This has completely changed how we do sentiment analysis, and weâ??ll explain how it all happened. Keep reading to learn more!

In recent years, NLP models based on the Transformer architecture have led to significant advancements in research and industry. Models like BERT, XLNET, GPT, and XLM have greatly improved state-of-the-art and achieved top positions in tasks such as text classification, named entity recognition, machine translation, sentiment analysis, question answering, and many more. However, these advancements come at a high computational cost. Transformer-based models are usually huge, with a growing number of parameters and training data. For example, the original BERT model had 110 million parameters, while the latest GPT-3 model has a staggering 175 billion parameters, which is about 1700 times more in just two years of research. Training such massive models typically require hundreds of GPUs and several days to complete. Thankfully, transfer learning allows us to download pre-trained models and fine-tune them on our smaller datasets at a lower cost. Once the training is finished, we still have a large model that needs to be deployed for production. However, the inference,

or the process of using the model to make predictions, can be relatively slow compared to smaller models. This slowness can affect the throughput or the speed at which the model can process data to meet our requirements.

## Understanding the Challenge

At the heart of our e-commerce use case was the need to efficiently process large volumes of customer reviews. Initially, our opinion mining system employed the BERT model to extract key information and then applied sentiment analysis using Simple Transformers. However, this process was time-consuming, requiring approximately 2 hours to process just 1,000 reviews on a CPU.

## Exploring ONNX Runtime

Determined to optimize our systemâ??s performance, we conducted extensive research to identify a solution to reduce the processing time without compromising accuracy. Thatâ??s when we came across the Open Neural Network Exchange (ONNX) Runtime, a robust framework designed to accelerate machine learning models.

The ONNX Runtime facilitates the conversion of machine learning models into the efficient ONNX format, enabling faster inference and execution on various hardware platforms. Intrigued by its potential, we decided to experiment with ONNX Runtime and determine whether it could be the game-changer we sought.

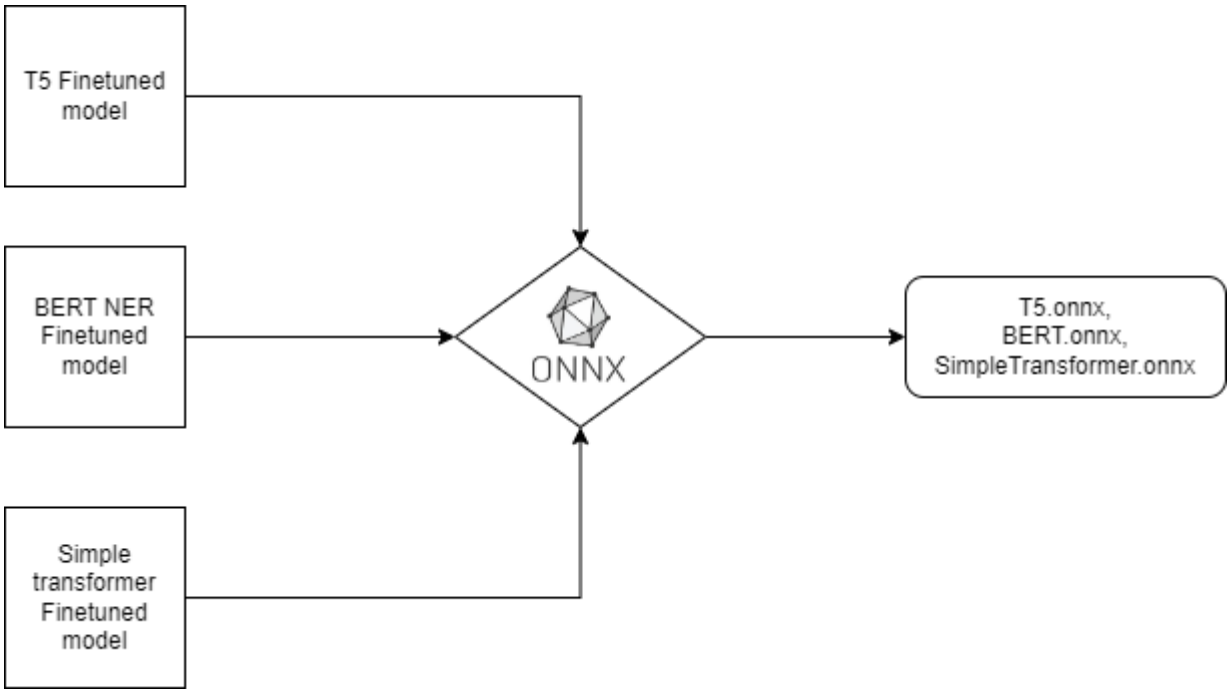
## Conversion to ONNX Models

Our first step was to convert our fine-tuned models BERT, T5, and Simple Transformer into ONNX models. This process involved mapping our existing modelsâ?? intricate neural network architectures and weights to the ONNX format. While this conversion requires some effort, the results were well worth it.

The conversion process typically involves the following steps:

**Exporting Model:** The trained model must be exported from the original framework (e.g., TensorFlow or PyTorch) into a serialized format. For instance, in TensorFlow, the model is saved using the SavedModel format, while in PyTorch, it is saved using the torch. Save function.

**ONNX Conversion:** Next, the serialized model is converted into the ONNX format using the ONNX library. Both TensorFlow and PyTorch have built-in support for ONNX conversion. TensorFlow models can be converted using the tf2onnx package, while PyTorch models can be converted using the torch. onnx.export function.




## The Power of ONNX Inference

After successfully converting our models, we again executed the opinion mining pipeline, this time utilizing the ONNX Runtime for inference. The difference in performance was astounding. Our system now processed 1000 customer reviews in a mere 18 seconds, a remarkable improvement from the previous 133-minute duration on a CPU. The ONNX Runtimeâ??s ability to leverage hardware acceleration and optimize execution made this transformation possible.

| Number of Reviews | Prediction time with ONNX on GPU | Prediction time without ONNX on GPU | Prediction time without ONNX on CPU | Prediction time with ONNX on CPU | Accuracy & F1 with simple transformer bert model | Accuracy & F1 with simple transformer bert onnx model |
|---|---|---|---|---|---|---|
| 200 | 7 s | 146 s | 3420 s (57 minutes) | 3.73s | Acc = 90 , F1 = 88 | Acc= 90, F1 = 88 |
| 500 | 17 s | 338 s | 3780 s (63 minutes) | 9.03s | Acc = 89, F1= 87 | Acc= 89, F1 = 87 |
| 1000 | 36 s | 728 s | 7980 s (133 minutes) | 18.3s | Acc = 89.1 F1 = 88 | Acc = 89.1, F1 = 87 |
| 2000 | 75 s | 1423 s | 14400 s (240 minutes) | 38.07s | Acc=89.25, F1=88 | Acc = 89.25 , F1= 88 |

*CPU specifications â?? Intel(R) Core(TM) i7-10610U CPU @ 1.80GHz 2.30 GHz

Download PDF

## Benefits and Implications

The adoption of ONNX Runtime had far-reaching benefits for our e-commerce use case. Not only did it drastically reduce processing time, but it also opened doors to scalability and cost-efficiency. With accelerated inference, we could handle larger volumes of customer reviews within shorter time frames, enabling real-time analysis and quicker decision-making.

Moreover, the ONNX formatâ??s compatibility with various hardware platforms allowed us to leverage parallel processing capabilities, including GPUs and specialized accelerators. This flexibility expanded our computational options and facilitated future system performance enhancements.

## Limitations and Challenges

ONNX Runtime supports a wide range of machine learning models, but it might only partially cover some custom or specialized model architecture. Some complex models or specific operations may not be fully compatible with ONNX, requiring adjustments or custom implementations to achieve successful conversions. Also, The process of converting models to the ONNX format adds an additional step to the deployment pipeline.

While ONNX is growing in popularity, it might still need to offer the same comprehensive ecosystem, potentially leading to a lack of some specific tools or community support for certain use cases.

## Conclusion

By embracing the power of ONNX Runtime, we revolutionized our approach to opinion mining in the ecommerce domain. Integrating ONNX models into our system dramatically reduced processing time, enabling us to extract insights from customer reviews swiftly and efficiently. With faster inference and optimized execution, our sentiment analysis pipeline became a more agile and scalable solution.

Innovation and advancements in NLP and machine learning continue to transform the ecommerce industry, empowering businesses to understand their customers better and improve their products and services. During our journey of using ONNX Runtime, we discovered the true power of our opinion mining system. It brought a new era of speed and accuracy to our work, improving things. The future of sentiment analysis in ecommerce is undoubtedly exciting, and we eagerly anticipate the next wave of innovation that will continue to shape the landscape.