Text classification with RoBERTa

by Roberto Silveira

2 min read • August 19, 2019

CATEGORIES machine_learning, nlp, pytorch

Fine-tuning pytorch-transformers for SequenceClassificatio

As mentioned already in earlier post, I'm a big fan of the work that the Hugging Face is doing to make available latest models to the community. Very recently, they made available Facebook RoBERTa: *A Robustly Optimized BERT Pretraining Approach*[1]. Facebook team proposed several improvements on top of BERT[2], with the main assumption tha BERT model was *"significantly undertrained"*. The modification over BERT include:

1. training the model longer, with bigger batches;
2. removing the next sentence prediction objective;
3. training on longer sequences;
4. dynamically changing the masking pattern applied to the training data;

More details can be found in the paper, we will focus here on a practical application of RoBERTa model using `pytorch-transformers` library: text classification. For this practical application, we are going to use the SNIPs NLU (Natural Language Understanding) dataset[3].

NLU Dataset

The NLU dataset is composed by several intents, for this post we are going to use `2017-06-custom-intent-engines` dataset, that is composed by 7 classes:

- **SearchCreativeWork** (e.g. Find me the I, Robot television show);
- **GetWeather** (e.g. Is it windy in Boston, MA right now?);
- **BookRestaurant** (e.g. I want to book a highly rated restaurant for me and my boyfriend tomorrow night);
- **PlayMusic** (e.g. Play the last track from Beyoncé off Spotify);
- **AddToPlaylist** (e.g. Add Diamonds to my roadtrip playlist);
- **RateBook** (e.g. Give 6 stars to Of Mice and Men);
- **SearchScreeningEvent** (e.g. Check the showtimes for Wonder Woman in Paris);

pytorch-transformers `RobertaForSequenceClassification`

As described in earlier post, `pytorch-transormers` base their API in some main classes, and here it wasn't different:

- RobertaConfig
- RobertaTokenizer
- RobertaModel

All the code on this post can be found in this Colab notebook:

Text Classification with RoBERTa

First things first, we need to import RoBERTa from `pytorch-transformers`, making sure that we are using latest release **1.1.0**:

```
from pytorch_transformers import RobertaModel, RobertaTokenizer
from pytorch_transformers import RobertaForSequenceClassification, RobertaConfig

config = RobertaConfig.from_pretrained('roberta-base')
tokenizer = RobertaTokenizer.from_pretrained('roberta-base')
model = RobertaForSequenceClassification(config)
```

As the NLU dataset has 7 classes (labels), we need to set this in the RoBERTa configuration:

```
config.num_labels = len(list(label_to_ix.values()))
```

```
{
  "attention_probs_dropout_prob": 0.1,
  "finetuning_task": null,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 514,
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "num_labels": 7,
  "output_attentions": false,
  "output_hidden_states": false,
  "torchscript": false,
  "type_vocab_size": 1,
  "vocab_size": 50265
}
```

In this notebook, I used the nice Colab GPU feature, so all the boilerplate code with `.cuda()` is there. Make sure you have the correct device specified [`cpu` , `cuda`] when running/training the classifier.

I fine-tuned the classifier for **3** epochs, using `learning_rate` = **1e-05**, with `Adam` optimizer and `nn.CrossEntropyLoss()` . Depending on the dataset you are dealing, these parameters need to be changed. After the **3** epochs, the train accuracy was **~ 98%**, which is fine considering a small dataset (and probably a bit of overfitting as well).

Here are some results I got using the fine-tuned model with `RobertaForSequenceClassification` :

```
get_reply("play radiohead song")
'PlayMusic'

get_reply("it is rainy in Sao Paulo")
'GetWeather'

get_reply("Book tacos for me tonight")
'BookRestaurant'

get_reply("Book a table for me tonight")
'BookRestaurant'
```

RoBERTo hopes you have enjoyed RoBERTa 😁 and you can use it in your projects!

References

1. **RoBERTa: A Robustly Optimized BERT Pretraining Approach** PDF
   Yinhan Liu and Myle Ott and Naman Goyal and Jingfei Du and Mandar Joshi and Danqi Chen and Omer Levy and Mike Lewis and Luke Zettlemoyer and Veselin Stoyanov, 2019
2. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding** PDF
   Jacob Devlin and Ming-Wei Chang and Kenton Lee and Kristina Toutanova, 2018
3. **Natural Language Understanding benchmark** Link

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, Joseph Dureau, 2018

Share Tweet LinkedIn Reddit

Previous
← *Machine Comprehension with pytorch-transformers*