Sentiment Analysis using transformers in Python

Naman Garg

ngarg3@binghamton.edu

Abstract

Recently, natural language processing using transformers has gained a great deal of attention in the field of machine learning. Transformers give better accuracy and optimization on text-based classification compared to other sequential models such as recurrent networks. In this paper, we explore different transformer models that capture the sentiment information of individual words with the whole sentence. We use an emotion dataset that classifies the emotions for sentences to predict sentiments.

Introduction

Sentiment analysis in today's world has a wide variety of applications. Companies, especially for advertising and product, use sentiment analysis to evaluate and improve their performance. The companies analyze the sentiments of their customers using product reviews. Sentiment analysis is also used to help predict market performance in the financial sectors using textual data, such as financial news articles that capture people's sentiment and use it to determine the decisions they might take based on the articles, which can help in improving investment performance. Furthermore, writing assistance tools such as Grammarly, use sentiment analysis to determine emotions in the text documents used and help guide a user to edit them according to the user's preferences. However, a good sentiment analysis tool is not widely available and can result in great performance and accuracy on small paragraphs or sentences. This paper shows an implementation of sentiment analysis using transformer models that have been pre-trained on large unlabelled language corpus.

Related Work

Transformers

Transformers, as mentioned in the 'Attention is all you need' paper by Vaswani et al. (2017), is an architecture that transforms one sequence into another one using an encoder and a decoder. Vaswani et al., in their paper, describe that the transformer model reads the entire sequence of tokens at once. This architecture differs from other sequence models as it doesn't use recurrent networks such as LSTMs (long short-term memory) which read the tokens sequentially (left-to-right or right-to-left).

BERT

The BERT model is a Bidirectional Encoder Representations from Transformers, developed by Google. This model is pre-trained on a large amount of unlabeled data, which then can be fine-tuned using supervised learning with fewer labeled samples. The pre-trained model uses 'contextualized word embeddings', (Peters et al.,2018) which means that the words are encoded based on their context or meaning. These tokens are pre-processed such that 80% are replaced with a "[MASK]" token(Horev, 2018). During training, the model tries to predict the masked tokens as a self-supervised learning task. The general transformer model uses encoder and decoder networks, whereas the BERT model only uses an encoder to learn language representation from the input text (Devlin et al., 2019).

Implementation/Approach

Model

A Robustly optimized BERT approach (RoBERTa), a model developed by Facebook, is an improved retrained implementation of the BERT model with more data and computation power. Liu et al. in their paper describing the RoBERTa model, mention that the BERT model is "significantly undertrained". The RoBERTa model removes 'the Next Sentence Prediction (NSP) task' from BERT's pre-training and introduces the concept of dynamic masking which changes the masked token during the training (Liu et al., 2019). The RoBERTa model, as shown in the table below, is pre-trained on nearly ten times more data than the original BERT model. Furthermore, the model is trained with longer sequences and a dynamic masking pattern.

Model	data	bsz	steps
RoBERTa			
with BOOKS + WIKI	16GB	8K	100K
+ additional data (§3.2)	160GB	8K	100K
+ pretrain longer	160GB	8K	300K
+ pretrain even longer	160GB	8K	500K
$\text{BERT}_{\text{LARGE}}$			
with BOOKS + WIKI	13GB	256	1M

Table 1. Development set results for RoBERTa (Liu et. al., 2019)

Data

The dataset is preprocessed for emotion recognition by Saravia et al., who used pattern-based representations to classify different emotions in their approach. For this paper, a sample of 20,000 sentences is used. The dataset contains the different sentences classified with emotions :

- sadness 😢
- joy 😃
- love 🥰
- anger 😡
- fear 😱
- surprise 😯

Data Processing

The data is processed into tokens for sentiment analysis using the pre-trained tokenizers for the model in the following steps

- **The tokenizers** process the data by converting the words in the text into tokens.
- The tokens are encoded into the numerical encoding of the tokens in the vocabulary and attention mask which indicates the model which tokens should be attended to (1s), and which should not (0s).
- A vector of fixed size 512 dimensions for tokens is created for each sentence and the tokens are padding to match the dimensions.

Processed Data:

Sentence:	`When was I last outside? I am stuck at home for 2 weeks.`
Tokens:	`['When', 'was', 'I', 'last', 'outside', '?', 'I', 'am', 'stuck', 'at', 'home', 'for', '2', 'weeks', '.']`
Token IDs:	`[1332, 1108, 146, 1314, 1796, 136, 146, 1821, 5342, 1120, 1313, 1111, 123, 2277, 119]`
Attention Mask:	`[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]`

Details

The model uses cross-entropy loss for calculating loss and evaluating the performance.

- The RoBERTa model is trained on the following parameters which are used to fine-tune the model.

```
batch_size=16
warmup_steps=100
Learning rate = 1E-05
Epochs = 4
accumulate_grad_batches=1
Training Time = 23 minutes 8 seconds
```

- The BERT model is trained on the following parameters which are used to fine-tune the model.

```
batch_size=16
warmup_steps=100
Learning Rate = 2E-05
Epochs = 10
Training Time = 43 minutes 23 seconds
```

Results

	precision-recall		f1-score	support
sadness	0.984183	0.958904	0.971379	584
joy	0.963557	0.941595	0.952450	702
love	0.783784	0.895062	0.835735	162
anger	0.916981	0.949219	0.932821	256
fear	0.909524	0.888372	0.898824	215
surprise	0.858824	0.901235	0.879518	81
accuracy			0.936500	2000
macro avg	0.902809	0.922398	0.911788	2000
weighted avg	0.939006	0.936500	0.937292	2000

Table 2. Classification Report for the RoBERTa model for accuracy and performance.

	precision	recall	f1-score	support
sadness	0.940000	0.870370	0.903846	540
joy	0.879518	0.935897	0.906832	780
love	0.846154	0.647059	0.733333	170
anger	0.875000	0.840000	0.857143	250
fear	0.800000	0.842105	0.820513	190
surprise	0.500000	0.714286	0.588235	70
accuracy			0.865000	2000
macro avg	0.806779	0.808286	0.801650	2000
weighted avg	0.871610	0.865000	0.865716	2000

Table 3. Classification Report for the BERT model for accuracy and performance.

The RoBERTa model, as observed from the result shown above tables (Table 2, Table 3), gives significant accuracy and performance (**93.6 %**) compared to the BERT model (**86.5%**) for this data. This result is parallel to the ones observed by Liu et. al in their paper. This shows that the RoBERTa model is an effective model that can be used for sentiment analysis and provides substantially better performance compared to other transformer and recurrent network models.

The accuracy for sentiment analysis in the RoBERTa model even goes as far as **97%** for an emotion (**sadness**). However, for the emotions such as fear, love and surprise the model gives lower accuracy (less than 90%) than other emotions -joy, anger, and sadness- as there are fewer labeled samples available in the dataset for these emotions. Furthermore, it is easier for the model to confuse emotions such as joy and love as they contain nearly similar word embeddings as words used to identify these emotions have similar meanings.

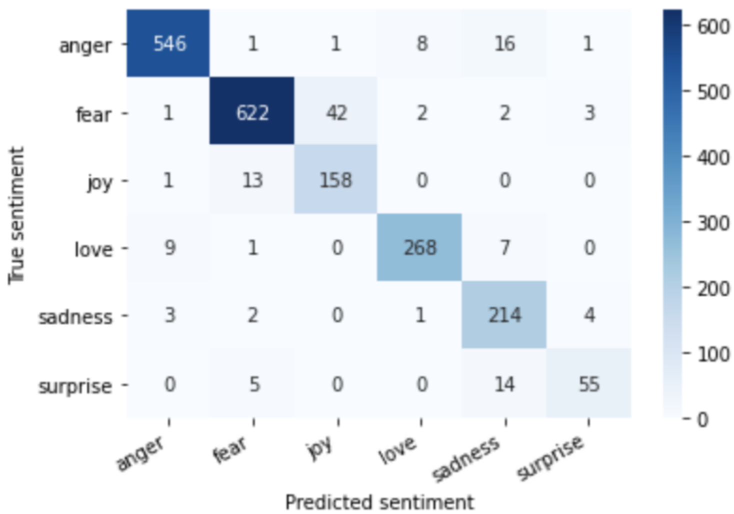

Fig. Pandas confusion matrix representation for prediction of each emotion for the RoBERTa model.

The following function below can be used to predict the sentiment on any real world sentence.

```
Module - trained model
label2int = {"sadness": 0,"joy": 1,"love": 2, "anger": 3, "fear": 4,
"surprise": 5 }
def get_reply(msg):
       module.cuda()
       model = module.eval()
       enc = tokenizer.encode_plus(msg)
       X = torch.tensor(enc["input_ids"]).unsqueeze(0).cuda()
       attn =
      torch.tensor(enc["attention_mask"]).unsqueeze(0).cuda()
       output = module((X,attn))
       _, pred_label = torch.max(output.data, 1)
       prediction=list(label2int.keys())[pred_label.item()]
      print(prediction)

get_reply("Alexa play a romantic song on Spotify")
-> love
get_reply("I am trying my best in life, but I am still lacking in
everything that I try to accomplish.")
-> sadness
```

Experiment Examples:

I would show up every once in a while to an empty classroom and just return to my office shrugging my shoulders and feeling somewhat of a guilty pleasure in not having to teach a class.
predicted: sadness
Expected sadness

I sat down and asked my mind to shut up not feeling particularly hopeful that anything would come of it.
predicted: joy
Expected joy

I will attempt drugs if you specifically ask for them because I am feeling dangerous.
predicted: anger
Expected anger

I am feeling up to it I will publish the next installment of this wonderful horror serial.
predicted: joy
Expected joy

I was feeling a bit dull and we were trying to figure out exactly why.
predicted: sadness
Expected sadness

Conclusion/Future work

In conclusion, the sentiment analysis implemented using the RoBERTa model gives significant performance and accuracy on the emotion dataset as compared to the BERT model. Also, the model works well with real-world text data. For future work, we can apply and compare other transformer models for the sentiment analysis such as the T5-base model, another transformer model developed by Google. Also, Sanh et al. in their paper describe a pre-trained model DistilBERT which contains 40% fewer parameters than the BERT model and runs 60% faster while preserving over 95% of BERT's performance (2019).

Work Cited

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", arXiv:1810.04805 [cs.CL] 2019.

Saravia, Elvis, Liu, Hsien-Chi Toby, Huang, Yen-Hao, Wu, Junlin, and Chen, Yi-Shin. Contextualized Affect Representations for Emotion Recognition. In Conference on Empirical Methods in Natural Language Processing, 2018.

Rani Horev. BERT Explained: State of the art language model for NLP, 2018. https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. "RoBERTa: A Robustly Optimized BERT Pretraining Approach", arXiv:1907.11692 [cs.CL], 2019.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, "Deep contextualized word representations", arXiv:1802.05365 [cs.CL], 2018.

Victor Sanh, Lysandre Debut, Julien Chaumond and Thomas Wolf. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter", arXiv:1910.01108 [cs.CL], 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In Advances in neural information processing systems, arXiv:1706.03762 [cs.CL], 2017.