SENTIMENT ANALYSIS

Citibeats

Author
Aurora Cobo Aguilera
a.cobo.aguilera@gmail.com
14th July 2023

1 Introduction

The attached notebook solves a Sentiment Analysis work, comparing a baseline based in a traditional model with a Neural Network one.

All experiments are run in a small subset of data in order to speed up the process. In addition all steps are explained in the notebook, leaving the report to answer some proposed questions.

2 Executive summary

The models used in the work are mainly 3:

- TF-IDF vectorization and Logistic regression
- TF-IDF vectorization and SVM
- BERT and LSTM model

The metrics explored during the work are very diverse: accuracy, F1 score, recall, precision, ROC curve, confusion matrix, loss function in the neural network training...

I conclude that the pretrained Neural Network based model perform much better than the baseline. However, due to resources limitations I could not obtain good conclusions as I did not explored well the hyperparameters and configurations as well as all samples information.

In test, the accuracy will always be better in in-domain data than in out-domain data. Here, data augmentation or pretrained model with 3 sentiment classes would have been ideal.

3 Motivation

During the work, I had several issues I will detail next. The main difficulty was due to the high quantity of data and the impossibility to train and process the data with GPU, RAM and suitable resources. I decided to take a subset of samples (1000 samples per class) in training and maintain the original test sets. I also study only the proposed columns features of data, but let the idea of using the date or user ID to find some correlations, such as usual negative comments in a specific user or positive ones in festive periods.

3.1 Model 1: Baseline

As the baseline a use a traditional vectorization of TF-IDF with basic cleaning of tokens, that is removing stopwords and non alphanumeric caracters. In addition, I include some common bi and tri grammars. I also lowercase and use a pretrained model from Spacy to make all this preprocessing.

In a scenario with more time, I would have encoded emoticons, that are super valuable in a sentiment analysis problem. removing them is not the best option.

Because of the same reason of the few samples, my final dictionary is quite poor with few tokens, a possible reason for the vectorization fail in the classification performance.

With the TF-IDF matrix I train two models, a Logistic Regression and a SVM. I follow a cross-validated training of some model hyperparameters.

The accuracy in the test set are quite far from the training one. I need a real training with all samples to fix final conclusions about the usability of this model to solve this task.

3.2 Model 2: Neural Network

In the second model, I take the same set of samples from training and build a Neural Network model thanks to Pytorch library. My model consist on a first tokenization step with BERT, the corresponding Encoder to build the embeddings, a LSTM and a final layer to project data to the desired dimension, the number of classes, in a classification problem.

3.3 Proposal of improvement

In Neural Network model it would have been interesting check at the attention scores in BERT. I might have seen some tokens that are more important in the decision, or some of them that are not being taken into account and should be. I keep this as a possible way of exploring more what the model is doing and how to improve it. Some text feature might be included such as negations, emoticons or expressions.

Other improvements have been exposed during the notebook and the report.

4 Research questions

4.1 What are the accuracy/f1/recall/precision of your model on the Sentiment140 dataset? And on the Dublin dataset?

Here I expose accuracy score although I have computed the others metrics for the baseline model, not for the NN model because of time availability.

Training, Test and Validation belong to Sentiment140 dataset, but validation accuracy is only obtained in the Neural Network model.

	Training	**Test**	**Test-Dublin**	**Validation**
TF-IDF_LR	83.70%	46.79%	38.2%	X
TF-IDF_SVM	92.7%	46.39%	38.23%	X
BERT-LSTM	96.81%	60.24%	44.53%	85.25%

4.2 Are they or are they not the same? How large is the difference? Can you think of the cause? How would you corroborate or refute your hypothesis?

They are obviously not the same. The accuracy is much better in Sentiment dataset than in the Dublin one. It is due to the differences in the samples regarding the training ones, that belong to the Sentiment dataset. You can check different samples to see the differences, regarding the topics or the lenght in the comments. Even using links, hastags, mentions to users may complicate the training.

4.3 What are the main classification challenges?

The main challenges are the few samples I was able to use for the trianing, what provoked a lower accuracy. In addition, the number of classes. I only have 2 classes in training and 3 in test, but I had no indication regarding if I could use external dataset to make some augmentation for the neutral class. Another perspective I woudl, have follow is to use a 3-classes pretrained model from, for example, transformers and hen finetune in my data.

4.4 Can you think of a way to measure whether the accuracy is higher for some of the project categories instead of others, e.g. whether sentiment classification accuracy is higher for the Public Spaces category than for Community and Culture?

I ave implemented this step in my notebook. Actually, I obtain better accuracy in the samples from category 'Community and Culture' than from 'Public -Spaces'. The possible reason for this is that the original Sentiment dataset must have samples more related/similar to 'Community and Culture' topic.

4.5 Can you think of any way to measure whether the metrics are higher for documents containing specific types of entities, e.g. Person vs. Location?

I have not implemented this analysis in my code but it would be a Name Entity Recognition (NER) problem to select the samples and then check in the same way I did before. For example, find the samples with person names with some NER model adn the compute accuracy in those samples.

4.6 Can you make a proposal to address any/all the issues you have encountered in the evaluation?

Data augmentation or pretrained model as I mentioned previously for this 3-classes problem. That would be using external tool trained in other datasets. The option I take in my code is much more easier, assigning the class Neutral to the samples with more uncertainty, that is probability between 0.45 and 0.55 for example.

4.7 Can you bring any idea of new feature to develop at Citibeats for Social Understanding after working on this home assignment ?

User data such as location (country), sex and age can be very useful. In addition, I mention in the notebook about using also the date, where you can extract week day, year or festive period that can alter the sentiment feelings of a person.

5 Deployment

When deploying machine learning models in the cloud, there are several technologies and procedures you can use. Here is an explanation of a recommended approach to deploying both models in the cloud:

5.1 Containerization

One of the best practices for deploying machine learning models is to use containerization technology such as **Docker**. Containerization allows you to package your models, dependencies, and runtime environment into lightweight and portable containers. This ensures consistency and reproducibility across different environments.

5.2 Model Serving

To serve the models in the cloud, you can use a dedicated model serving framework or library. One popular choice is **FastAPI**, which is a Python web framework that can be used for serving machine learning models. These frameworks allow you to expose restful APIs to receive requests and return predictions.

5.3 Cloud Provider

Choose a cloud provider that offers scalable and managed infrastructure services. Some popular options include **Amazon Web Services** (AWS), **Google Cloud Platform** (GCP), and **Microsoft Azure**. Selecting a cloud provider allows you to take advantage of their infrastructure capabilities and easily scale your deployment as needed.

5.4 Monitoring and security

To improve the user experience and security you can implement monitoring and logging to gain insights into the performance and behavior of your deployed models. In the same way you can implement appropriate security measures to protect your models and data. This includes securing network connections, implementing authentication and authorization mechanisms, and encrypting sensitive data.

These are some of the key considerations when deploying machine learning models in the cloud. The specific technologies and procedures may vary depending on your requirements, chosen cloud provider, and the nature of your models. It is important to thoroughly plan and test your deployment process to ensure smooth and reliable operation in the cloud.