IA Series - Sentiment Analysis NLP cook-book (using Pre-Built NLP Model)

Introduction

As part of mandatory reading for folks in RPA these days, machine learning is a topic which most of the time remains in the to-do list. But building large Machine learning models itself require machines having considerable processing power, which may not be possible for every RPA developer.
Sentiment analysis is a very old topic in NLP space, the guide gives a brief how to for implementation of a pre-built Sentiment Analysis model and its use in a basic solution without using high performance GPU based machines.

Actual use cases are varied in the fields of Social media monitoring, Brand monitoring, Voice of customer (VoC), Customer service, Market research etc.

Background

The guide tries to give a simplistic set up which can be set up and tested in a decently sized work laptop (for development and evaluation only). BERT is a language model first open sourced by Google in 2018 and rolled out in 2019. It is one of the best models available today for NLP and is based on the transformer model (not the movie).

The current model is created from fine tuning the existing BERT model using the IMDb movie reviews data set and should cater to most English content for sentiment analysis. It can except paragraphs with a maximum of 500 words. Ktrain is a python package which is used for simplification of the existing implementation. The actual model creation is not in the scope of this guide.

Software information

Software List and links

Term	Link
Blue Prism	Robotic Process Automation tool - https://portal.blueprism.com/products/current (v6.7)
GIT	Source code repository. You need to have GIT installed in your system. For this guide a public repo was created at: https://github.com/ashz30/Sentiment-Analysis-BERT-KTrain.git
Python	Python Version 3.6 is used, Download link here. Python IDE used is PyCharm. Download link here. (users can also run the code without the IDE as a command line too)
Python Libraries	Ktrain (installed with pip commands) Flask (installed with pip commands) – only used for exposing python code as Web API's. (please refer to links above for any license information)

Pre-requisites

Prior to starting this process the below steps need to be configured for GIT repository set up

- Clone git repository - (above 1GB) – open cmd in a blank folder (mine is C:\GIT) and type

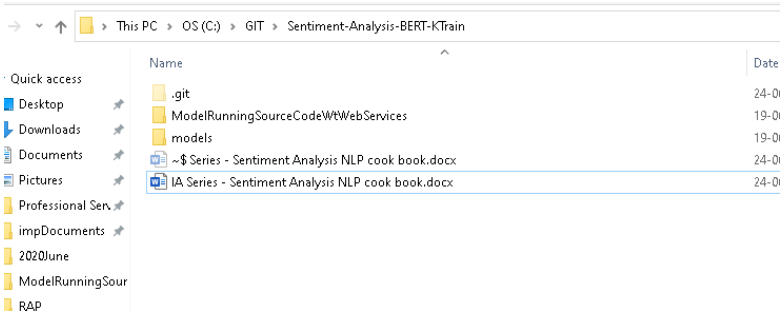
  “git clone https://github.com/ashz30/Sentiment-Analysis-BERT-KTrain.git”

- Install Python, open any cmd window and run the below commands:
  - Check python version – “python --version”

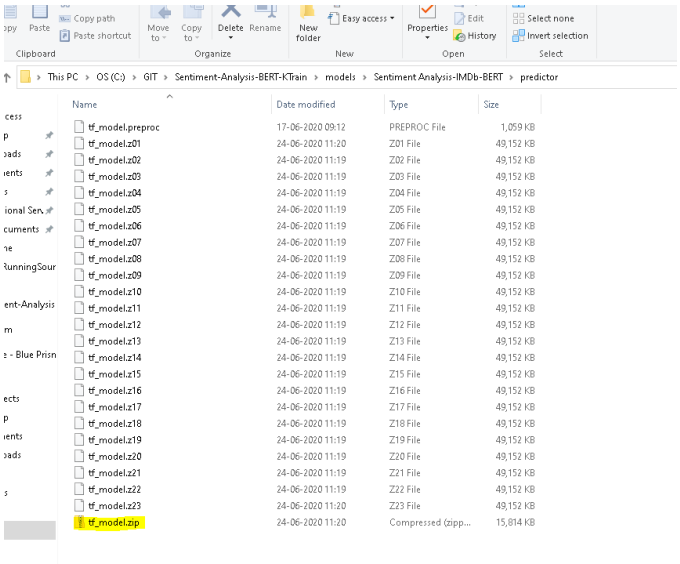
    (if this does not show the correct python version, you will need to edit windows path to point to python folderand open a new cmd window to try again)
  - “pip install ktrain” (to download and install Ktrain)
  - “pip install flask” (to download and install wrapper REST API framework)
- User will need to install Blue Prism in any machine.

Python Web service set up steps

- Once cloned this is the structure of the GIT folder –



- Goto folder - C:\GIT\Sentiment-Analysis-BERT-KTrain\models\Sentiment Analysis-IMDb-BERT\predictor (or your corresponding folder path).
- You should be able to view the model divided into multiple zip files. Extract tf_model.zip file highlighted.

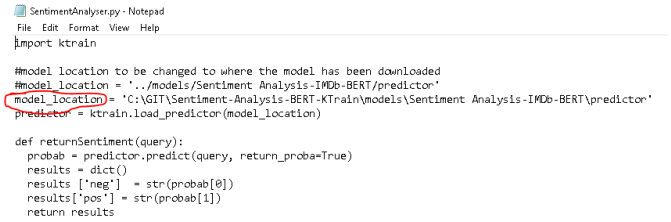


  This step should extract the model file - tf_model.h5 ~ 1.22 GB on windows.

  Keep only files tf_model.h5 and tf_model.preproc in the predictor folder, you can move the other zip files away to a different folder.

- Goto folder **ModelRunningSourceCodeWtWebServices**, open file SentimentAnalyser.py and update

model_location = <path of the predictor folder in **models** folder>

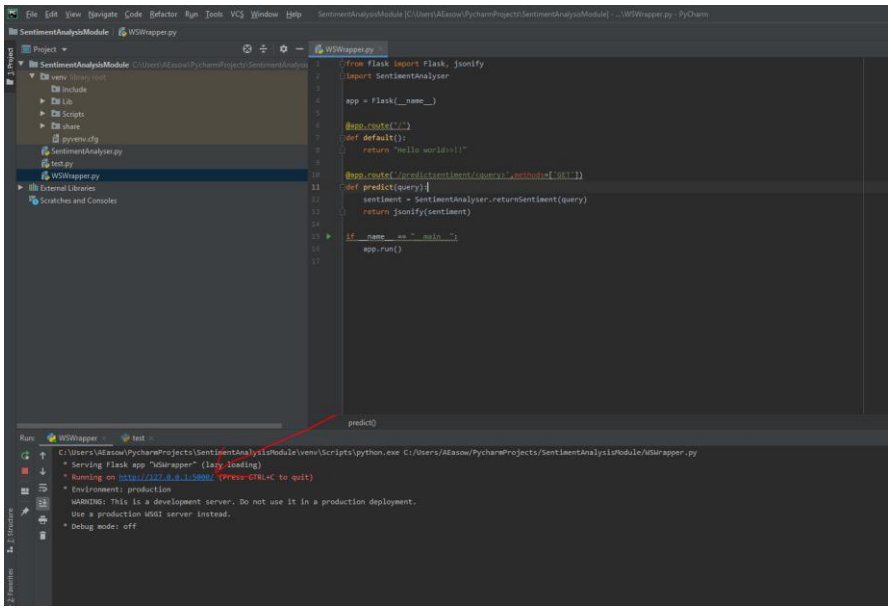


Save the file once done.

- Open command line in folder '**ModelRunningSourceCodeWtWebServices'** and run command

  'python WSWrapper.py'

  alternately run WSWrapper.py through the IDE.
- Provided everything is set up correctly, it will load the model (takes a min) and start a web server in your local system.



- Open your browser to goto link –

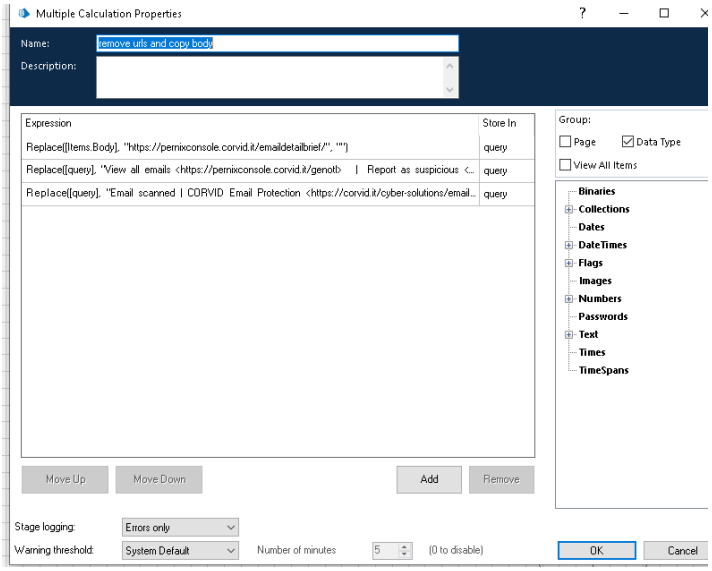
  127.0.0.1:5000/predictsentiment/"thats great..!! lets sign a deal for 100 more licenses"

  Hit enter
- You should see the result after a short pause the first time.

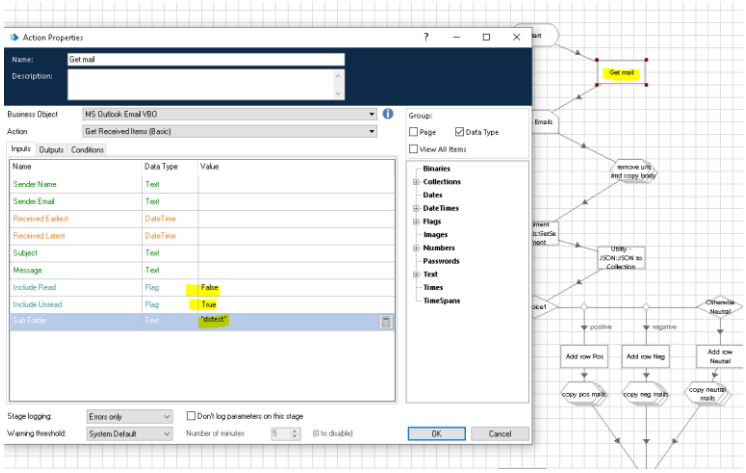
- There will never be a value for 100 percent, but rather a split between positive and negative.
- Also keep in mind, sentiment is a measure of feeling, so its never an exact science. To figure out if a statement is neutral it is best to take a call based on the confidence score. For eg. If both 'pos' and 'neg' are approaching 50%, then that's an indicator of the statement being a neutral one.

Blue Prism Steps

- Open Blue Prism.
- Import release file - Sentiment Analysis.bprelease, present in folder - BP release.
- This is a demo BP process, feel free to modify this as per your requirement. My emails have additional text from scans, so the demo process has replace functions used to remove them.

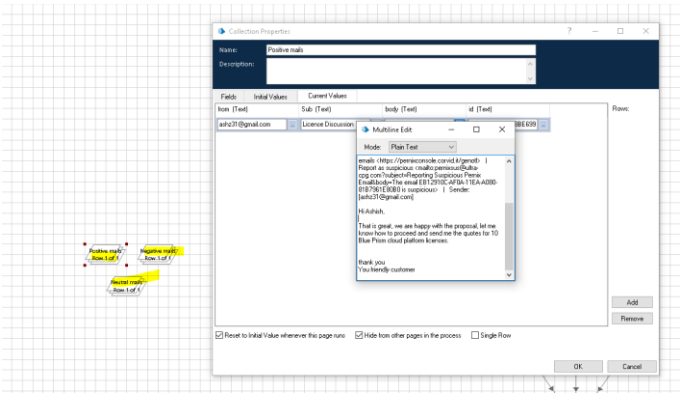


- Open system -> Web Api Services -> Sentiment Analysis and check if base url is as per python webservices which was hosted in Python Web Services set up step.
- Open Get Sentiment process -> Get Mail Action -> Verify conditions for email fetch are valid. In this case it checks for unread mails in ‘dxtest’ folder.



Demo Run

- If everything is set up correctly, run the BP process. It should get the emails and as per the sentiment divide the mails into 3 collections – Positive, Negative and Neutral.



Conclusion and Additional thoughts

The Sentiment Analysis module is trained from a pre existing data set, for cases like sentiment analysis on general English this works fine in most cases, as the data set is imdb, but in case of other languages and regional cases where the dialect and grammar spoken may be different, it may not be accurate in cases for other languages and results may not be accurate for the rest. In those case a new model needs to be built. It's best to evaluate using your own data set to get proper values of accuracy.

Pre-requisite for any Machine learning activity should be process improvement first and proper well labelled and categorised data next. Without the first one, the effort will be put in vain for cases which can be easily solved by process improvements, and without the second one, it will not be possible to even check if a machine learning model will work in solving the business case in a POC.