Disclaimer : This document talks about the solution of the Parspec Assignment for position Data Scientist [ MLE 2 ]

Author : Manish Sharma : [Linkedin](https://www.linkedin.com/in/manish-sharma-%F0%9F%93%8A-355ba3189/)

Hosted Web App Link : <https://parspec-bert-finetuned-lighting.streamlit.app/>

**---------------------------------------------------------------------------------------------------------**

**Problem ?**

* **Given an input pdf [ ranging from 1 - 30 / 40 pages ], classify it into Lighting / Non-Lighting category**

**---------------------------------------------------------------------------------------------------------**

**Solution –**

**1 - Code for the problem statement ?**

* Checkout my github repo : <https://github.com/MANISH007700/parspec_assignment>
* Please go through the readme.md for all information about the repo and where the code is

**2 - Inference Pipeline / Hosted link**

* Checkout my github repo [this also has a tutorial notebook if you wanna run locally ] - <https://github.com/MANISH007700/parspec_assignment>
* Hosted Web App Link -<https://parspec-bert-finetuned-lighting.streamlit.app/>

**3 A - How long did it take me to solve the problem ?**

* Working on freelance projects and side house-hold activities / attending functions. I could spare an **hour-to-two** at max per day
* Couldn’t work on Saturday, personal reasons.
* Started the project on -> 30/11/2023 [ 8:30 pm ]
* Completed on -> 03/11/2023 [ 11 am ]

**In total, ~4-5 hrs were spent on**

* **Downloading, Extracting, Cleaning, Downloading Models, Coding, Finetuning Model, Benchmarking the model, Pushing it to HUB, Deploying + Readme and Documentations**

**3 B - Explain my solution**

* **DOWNLOAD STAGE :** 
  + Firstly, all the PDF URL were downloaded in my local laptop for both training and testing data
  + Some of the PDF URL were invalid link, tried breaking it up w/ Selenium too, but since the URL were not proper, couldn’t bypass it
* **EXTRACTION STAGE :** 
  + Used **PyMuPDF and Fitz** to extract the text from the PDF. Since all of them were Digital in nature, it was easy to do so
  + If the Docs are in Non-Digital, we could have used Google-OCR for this purpose.
* **CLEANING STAGE :** 
  + Multiple thoughts went into this phase : Since there were lot of information in the PDF, but only few info were important to get it classified into lights and no-lights : Removed Stopwords, Removed Punctuations, Digits, Web URL, Emoji’s if any.
  + Idea was to take the summary of every entry, as model couldn’t take ~20k tokens while training. But Summarization model was taking a huge amount of time for processing, and due to time constraint had to drop it off.
  + Saved all the cleaned text into a CSV file for later use
* **MODELLING PHASE :** 
  + Went with a simpler approach rather than complicating the tasks
  + FineTuned **BERT BASE MODEL** on our custom datasets
    - Max length : 512
    - Truncation and Padding was used
  + Since BERT is pre-trained on large corpus and is built on top of transformers, it was a no-brainer to use this model as the start. It has good context window too, built on top of Attention
  + FineTuned this bert model on all the training data entry points [ ~830 data ] for 4 epochs
  + Configurations : [ PyTorch ]
    - Max Length : 512 tokens
    - Padding and Truncation : True
    - Epoch : 4
    - Used Learning Rate : 10e-4
    - Scheduler were used and Early Stopping
    - Training Acc - ~95 %
* Benchmarked on the holdout set
  + Accuracy : 91.25 %
* **ANY IMPROVEMENTS ? :** 
  + Yes, there can be a good amount of improvements can be made
  + Hyperparam optimization could have been done
  + We can use LayoutLM kind of Model which takes both structure and text into picture
  + We can extract the images from the PDF and take embeddings of those and merge with the text embeddings and finetune a model on it. There by utilizing both Images and Texts
* **ACCURACY on HOLDOUT :** 
  + 91.25 %

—----------------------------------------------------------------------------------------------------------------

Thank you

* Manish Sharma