|  |  |
| --- | --- |
| Resume Parser Project: Meeting minutes | |
| PROJECT: | Building a Resume Parser in Python using NLP and deploying the model on a cloud based platform using an application. |
| DATE: | February 1 – February 24, 2023 |
| MEMBERS: | SIDDHARTHA PATRA (0787887) |
|  | USAMA NORAT (0792225) |
|  | AMIR DAHYA (07912521) |

# Specific Activities from the prior 3 weeks

* **Data preparation**: We developed a code that reads a JSON file containing labeled data and converts it into a format that can be used by spaCy library. This step is essential for training and evaluating machine learning models.
* **Text extraction**: We also developed a code contains a function to convert PDF files to text. This step is essential when dealing with resumes that are in PDF format.
* **Model training**: Then we worked on a code that trains a spaCy model using the converted data from step 1. This step involves defining the model architecture, hyperparameter tuning, and training the model on the training data.
* **Model evaluation**: After that we evaluated the trained model's performance based on some new unknown resumes.
* **Debugging and testing**: We had to go through several iterations, including debugging, testing, and fixing issues encountered during development.

# Specific Outcome from prior 3 weeks

In the project code development phase, we defined and utilized multiple functions for an NLP pipeline. The output of those functions are discussed below:

* 1. The JSON file containing labeled data is successfully converted to spaCy format using the convert\_data\_to\_spacy() function.
  2. The check\_existing\_model() function is called to check if there is an existing spaCy model, and if there isn't, a new one is trained using the labeled data.
  3. The convert\_pdf\_to\_text() function is used to extract text contents from PDF files and store them in a list.
  4. The build\_spacy\_model() function trains a new spaCy model using the labeled data.
  5. The predict() function is called to make predictions on new data using the trained spaCy model.
  6. If there is an existing spaCy model, the build\_spacy\_model() function is called to update the existing model with the labeled data.
  7. The convert\_data\_to\_spacy() function logs an error message if there is an exception during the conversion process.
  8. The predict() function predicts named entities in the text using the trained spaCy model.
  9. The convert\_data\_to\_spacy() function returns None if there is an exception during the conversion process.
  10. The convert\_pdf\_to\_text() function writes the extracted text contents to text files if the appropriate code is uncommented.

# On Target

The project is on target and the status is green, as we have successfully developed our baseline model using the SpaCy library to perform entity recognition and identify key entities in a resume. We were also able to successfully test our model on a new unseen data (resumes) and check the accuracy of our entity identification model.

# Challenges/Disagreements

1. **Training Data Quality**: The training data used to train the spaCy model is not very representative of all the resumes types available in the market. This could lead to the model producing inaccurate or inconsistent results.
2. **Training Time**: Training the spaCy model is time-consuming, especially when we have to iterate over 100 iterations for better (or lower) loss. As a result, it can be challenging to optimize the training process to achieve the best possible results in a reasonable amount of time.
3. **Text Extraction**: The convert\_pdf\_to\_text() function used to convert PDF files to text may not work correctly for all PDFs. PDFs with non-standard formats or encryption can cause the function to fail.
4. **Error Handling**: Our code did encounter errors at various stages of the pipeline. Proper error handling is important for us to ensure that the pipeline continues to run smoothly even in the face of errors. This needed careful consideration of potential errors and the implementation of appropriate error-handling mechanisms (such as using try, catch and except block for coding).

# Planned Activities for coming weeks

The team has planned the following activities for the coming week:

* + - Research on google GCP (cloud platform) and understand how to utilize its engine APIs to deploy the code for the baseline model on the platform, using the Streamlit app.
    - Test on more real-world data and check for NER accuracy.

These planned activities will keep the project on track and bring us closer to delivering a functional Resume Parser in Python using NLP and deploying it on a cloud-based platform.