Data Intake Report

Group Name: Teddy Waweru
Email: teddywaweru@gmail.com
Country: Kenya
Specialization: NLP
Name: Resume Extraction
Report date: November 29, 2021
Internship Batch:
Version: 1.0
Data intake by: Teddy Waweru
Data intake reviewer:
Data storage location: https://gist.github.com/Rahulrky/b57ad459545c896231c4b770bd8d22ef

Tabular data details:

Total number of observations	160
Total number of files	1
Total number of features	Resume content, annotation start values, annotation end values, annotation labels, annotation text
Base format of the file	.json
Size of the data	1.04MB

Note: Replicate same table with file name if you have more than one file.

Proposed Approach:

- Mention approach of dedup validation (identification)
  The data had been preprocessed into the dataTurk format & contained the following attributes:
  {

  'content' : resume content or free text

  'label' : Annotated Tagged Entities

  }

- Mention your assumptions (if you assume any other thing for data quality analysis)
  It was assumed that the annotations included in the label were valid.

Problem Understanding

The aim of the project was to develop an **NLP extraction model** against the data provided. By implementing this solution, the resume parser would assist in structuring resumes that are collected, thereby reducing the efforts of classifying manually.

Data Annotation

The data provided adhered to the dataturk layout for pre-training data:

```
[{
content: resume content in string format,
annotation:     {
                label: Annotation label for the entity,
                points: {
                        start: start character position for the token in the content string
                        end: end character position for the token in the content string
                        text: the token string from the content string
                        }
                }
}]
```

This data structure was converted to the SpaCy model format:

```
{
        content: resume content in string format,
        annotation:     {
                        start character position for the token in the content string,
                        end character position for the token in the content string,
                        annotation label for the entity
        }
}
```

Named Entity Recognition (NER)

The spacy v3.0 was utilized to carry out the model training.
The notebook:
https://github.com/teddywaweru/DataGlacier/blob/Week7_NLP_Project_ResumeExtraction/Week7_NLP_Resume_Extraction/spacy_resume_extraction.ipynb
The Manipulation script file:
https://github.com/teddywaweru/DataGlacier/blob/Week7_NLP_Project_ResumeExtraction/Week7_NLP_Resume_Extraction/data_manipulation.py

Overlapping Annotations

During training, it was noted that some of the annotations' **start end values were overlapping**, which the spacy module called out as an error. The optional solution for the issue was to disregard these tokens, however this would mean loss in data ie. 'Companies worked at' entity overlapped with the 'Skill' entity in numerous occasions. I chose to segment these overlapping

entities into separate training information. There was no concrete method in the modules documentation that declared how to deal with overlapping annotations.

Model Building & Training

The model was built based on the spaCy package. There are significant changes in the current version in terms of how the learning data is parsed to the trained model, whose documentation is not quite clear.
I have reached out to experts in the community to offer their feedback on the assignment.

Performance Evaluation & Reporting

Documentation on the spaCy module is not descriptive enough on the variables collected during the training activity.

Model Deployment

Model was not deployed due to prevailing errors in the current build. Some entities in the test data

Model Inference

Inconclusive

Project Lifecycle

It is necessary to clean the initial data that was acquired in order to remove the overlapping annotations & essentially clean the data, after which the model should run appropriately, with relevant tweaking to match the anticipated goal of the project.
Project duration: 1 week for data wrangling & cleaning, 1 week for model improvement & adjustment.