Project

Resume Parsing and Classification Using Named Entity Recognition (NER)

Internship: Data Science Intern
Specialization: NLP
Name: Monisha Shree Senthil Nathan
University: IU International University of Applied Sciences, Germany
Batch: LISUM32, Data Glacier
Email: monishashree.career@gmail.com
Date: 23.06.2024

Contents

Problem description

HR departments face the challenge of manually processing a large number of resumes, which is both time-consuming and labour-intensive. Each resume contains various sections such as personal details, education, work experience, and skills. By using Named Entity Recognition (NER) models in Natural Language Processing (NLP), we can automate the extraction and classification of these entities, streamlining the resume screening process and making it more efficient and accurate.

Data understanding

The dataset contains text data from resumes, which includes both unstructured and semi-structured information. The key attributes in the dataset are:

content: This attribute contains the raw text of the resume. It includes various sections such as personal details, education, work experience, and skills.

label: This attribute contains the annotated tagged entities, which identify and classify specific information within the resume content.

Each annotation is a dictionary that includes, **label,** the category of the entity (e.g., Skills, Graduation Year, College Name, Degree, Companies worked at, Designation, Email Address, Location, Name) and **points,** the position of the text in the content, including the start and end positions and the actual text.

Problems in the data

1. **NA Values**:

- The presence of missing values (NA) can be problematic, especially in the label attribute if certain important entities are not tagged.
- It's essential to check for any missing values in the dataset and understand their impact on the NER model's performance.

2. **Outliers**:

- Outliers in text data are unusual or rare entities that do not conform to the general patterns observed in the data.
- For instance, a resume might have an exceptionally long or short content attribute, or it might contain entities that are not common in other resumes.
- Techniques such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) can be used to detect these outliers by clustering similar resumes and identifying those that do not fit well into any cluster.

3. **Skewed Data**:

- Skewness in text data can occur if certain entities or categories are overrepresented or underrepresented in the dataset.
- For example, if most resumes are from a specific industry or education background, the dataset becomes skewed towards those categories.

- This skewness can bias the NER model, making it less effective in recognizing entities from underrepresented categories.

Steps to Analyze and Mitigate Data Issues

1. **Check for Missing Values**:

- Identify any missing values in the content or label attributes.
- Analyze the impact of these missing values and decide on appropriate handling techniques, such as imputation or exclusion.

2. **Detect Outliers**:

- Use clustering techniques like DBSCAN to identify resumes that do not fit well into any cluster, indicating potential outliers (Sereshki et al., 2023). But this can be done after preprocessing.
- Analyze these outliers to understand their nature and decide whether to exclude them from the dataset or handle them differently.

3. **Analyze Skewness**:

- Skewness cannot be directly applied to raw text data. Instead, we can apply it to the numerical data present in the text data (entity types (tags) in the "label" attribute). For example, we can find the distribution of number of companies worked at, age, experience.
- Perform an exploratory data analysis (EDA) to identify any skewness in the data distribution.
- Visualize the frequency of different entity labels to understand the representation of each category.
- Consider techniques such as resampling or weighting to address any identified skewness.

Data Cleaning

Data cleaning is a crucial step in the data preprocessing pipeline that ensures data quality, consistency, and reliability. It involves identifying and correcting errors or inconsistencies in the dataset before analysis. Regular expressions (regex) are powerful tools for text processing and pattern matching, making them essential for data cleaning tasks involving textual data.

As stated in above section, the data does have some problems. So we have to deal with missing values and duplicate records. There are two records with missing labels

```
{'label': [], 'points': [{'start': 2585, 'end': 2590, 'text':
'Oracle'}]}
```

```
{'label': [], 'points': [{'start': 7878, 'end': 7882, 'text': 'B.B.M'}]}
```

We choose to ignore this and out of 200 records, only 1 record is duplicated. So, we can remove that duplicated record.

Handling Missing Values:

Function: `identify_records_with_missing_labels`

- **Objective:** Identify records with missing or empty labels in the 'annotation' column.
- **Implementation:** Iterate through each record and check for empty labels in annotations using nested loops.
- **Result:** Records with missing labels:

  Record 61
  Record 147

Removing Duplicate Records

Action: Identify and Remove Duplicates

- **Objective:** Ensure dataset integrity by identifying and removing duplicate records based on the 'content' column.
- **Implementation:** Utilize Pandas `duplicated()` method to identify and `drop_duplicates()` to remove duplicate records.
- **Result:** Ensure unique records are retained for accurate analysis.

Cleaning Text Data:

Functions: `remove_newlines_from_column`, `remove_punctuation`

- **Objective:** Standardize text format by removing newline characters and punctuation from specific columns.
- **Implementation:** Utilize regex (`re` library) to replace newline characters and remove non-alphanumeric characters.
- **Result:** Cleaned data for improved readability and analysis.

Skill Extraction and Cleaning:

Function: `extract_skills`

- **Objective:** Extract and clean skills data from resume annotations.
- **Implementation:** Use regex to split text by delimiters (e.g., comma, bullet points) and remove stopwords and irrelevant characters.
- **Action:** Ensure extracted skills are accurate and standardized for further analysis.

Custom Stopwords Removal:

Function: `remove_custom_stopwords`

- **Objective:** Remove predefined stopwords (e.g., 'having', 'experience', 'knowledge') from text data.

- **Implementation:** Compare each word against a custom set of stopwords and filter out irrelevant terms.
- **Result:** Improve the quality of skill data by eliminating non-informative words.

Data Transformation and Standardization

Mapping and Standardization:

- **Objective:** Standardize degree names (`qualifications_map`) and job titles (`designation_mapping`) for consistency.
- **Implementation:** Define mappings using dictionaries to map variations to standard names.
- **Action:** Apply mappings to respective columns (e.g., 'Degree', 'Designation') to ensure uniformity in analysis.

In the context of natural language processing (NLP), the exploration of various featurization techniques is pivotal for optimizing model performance. As part of this exploration, it has been observed that cleaning the data using regex to remove only '\n' characters and trailing spaces from entities sufficiently prepares the text for analysis without compromising entity boundaries. Specifically, attempts to remove additional punctuation marks have been found to introduce overlapping issues in entities detected by spaCy models. By retaining punctuation marks that delineate entities such as skills, locations, and job titles, the integrity of entity boundaries is preserved, ensuring accurate recognition and classification. This approach not only aligns with best practices in NLP preprocessing but also enhances the robustness and accuracy of subsequent model training and evaluation phases.

References

Taleb Sereshki, M., Mohammadi Zanjireh, M., & Bahaghi Ghat, M. (2023). Textual outlier detection with an unsupervised method using text similarity and density peak. *Acta Univ. Sapientiae Informatica*, 15(1), 91–110. DOI:10.2478/ausi-2023-0008