Project

Final Report

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: 02.07.2024

Contents

Abstract

The surge of online job applications has overwhelmed Human Resources (HR) departments with the task of manually screening a vast number of resumes. This process is not only time-consuming and labor-intensive, but also prone to human error, potentially leading to qualified candidates being overlooked. This project investigates the application of Named Entity Recognition (NER) models to automate resume screening, with the goal of improving efficiency and accuracy.

We explore three different NER models: SpaCy, BERT, and Conditional Random Fields (CRF). These models are trained on a dataset of resumes to identify and classify essential information such as names, contact details, educational qualifications, work experience, and skills. The performance of each model is evaluated using metrics like F1-score to determine the most effective approach for resume entity extraction.

Our findings indicate that the SpaCy model outperforms the BERT and CRF models in extracting resume entities. This demonstrates the potential of SpaCy as a valuable tool for streamlining the resume screening process for HR departments. By automating the extraction and classification of key applicant information, SpaCy can significantly reduce processing time, minimize human error, and enable HR professionals to focus on the more strategic aspects of recruitment.

Introduction

The ever-growing volume of online job applications has created a significant burden for Human Resources (HR) departments. The traditional method of manually screening resumes is a highly time-consuming and labor-intensive process. This approach suffers from several key challenges. First, inefficiency: HR professionals spend a considerable amount of time sifting through numerous resumes, often at the expense of other crucial tasks like conducting candidate interviews and selection. Second, human error: Manual screening is susceptible to human bias and fatigue, potentially leading to qualified candidates being overlooked due to inconsistencies in resume formatting or keyword matching. Finally, scalability: As the number of applicants increases, the manual approach becomes increasingly difficult to manage, hindering the overall recruitment process.

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.

Model Development

SpaCy Model

- **Implementation**: Used SpaCy for NER model training and entity extraction.
- **Preprocessing**: Cleaned data to retain punctuation marks that delineate entities.

BERT Model

- **Implementation**: Fine-tuned a pre-trained BERT model for entity recognition.
- **Preprocessing**: Similar text cleaning as SpaCy to maintain entity boundaries. The input format to the BERT model differs from Spacy and CRF model

CRF Model

- **Implementation**: Developed a Conditional Random Fields (CRF) model for NER.
- **Preprocessing**: Applied appropriate text cleaning techniques to ensure data quality.

This process of model development and pre-processing highlights the importance of tailoring both the NER model and data preparation to the specific domain and task at hand. Evaluating and comparing the performance of these three models would allow us to determine which approach is most effective for resume entity extraction.

Results and Discussion

CRF Model Performance

Entity Type	Precision	Recall	F1-score	Support
B-College Name	0.90	0.45	0.60	20
I-College Name	0.86	0.59	0.70	61
B-Companies worked at	0.65	0.26	0.37	42
I-Companies worked at	0.43	0.12	0.19	25
B-Degree	1.00	0.62	0.77	16
I-Degree	0.95	0.95	0.95	37
B-Designation	0.83	0.60	0.70	25
I-Designation	0.96	0.57	0.72	40
B-Email Address	0.89	0.89	0.89	9
I-Email Address	1.00	0.78	0.88	9
B-Graduation Year	1.00	0.33	0.50	12
B-Location	0.33	0.36	0.35	11
I-Location	0.00	0.00	0.00	1
B-Name	1.00	0.83	0.91	12
I-Name	0.91	0.83	0.87	12
O	0.94	0.98	0.96	4568
B-Skills	0.90	0.50	0.64	18
I-Skills	0.77	0.67	0.72	491
B-Years of Experience	0.00	0.00	0.00	1
I-Years of Experience	0.00	0.00	0.00	1
Micro avg	0.92	0.92	0.92	5411
Macro avg	0.72	0.52	0.59	5411
Weighted avg	0.92	0.92	0.91	5411

SpaCy Model Performance

Entity Type	Precision	Recall	F1-score	Support
College Name	0.88	0.96	0.92	106
Companies worked at	0.58	0.89	0.71	110
Degree	0.58	1.00	0.74	57
Designation	0.46	0.80	0.58	140
Email Address	1.00	0.87	0.93	23
Graduation Year	0.56	0.50	0.53	20
Location	0.93	0.68	0.78	37
Name	1.00	1.00	1.00	40
Skills	0.94	0.56	0.70	1260
UNKNOWN	0.00	0.00	0.00	0
Years of Experience	1.00	0.55	0.71	11
O	0.94	0.97	0.96	10312
Micro avg	0.92	0.92	0.92	12116
Macro avg	0.74	0.73	0.71	12116
Weighted avg	0.93	0.92	0.92	12116

BERT Model Performance

Entity Type	Precision	Recall	F1-score	Support
College Name	0.03	0.02	0.03	45
Companies worked at	0.03	0.02	0.02	53
Degree	0.07	0.02	0.03	52
Designation	0.54	0.38	0.45	84
Email Address	0.12	0.09	0.11	11
Empty	0.95	0.98	0.97	9652
Graduation Year	0.00	0.00	0.00	5
Location	0.25	0.11	0.15	9
Name	0.38	0.89	0.53	45
Skills	0.40	0.04	0.07	273

Entity Type	Precision	Recall	F1-score	Support
Years of Experience	0.00	0.00	0.00	11
Micro avg	0.93	0.93	0.93	10240
Macro avg	0.25	0.23	0.21	10240
Weighted avg	0.92	0.93	0.92	10240
Samples avg	0.93	0.93	0.93	10240

Model Performance Comparison

We evaluated three different models for Named Entity Recognition (NER) on our dataset of resumes: SpaCy, BERT, and CRF. Here's a detailed comparison of their performance metrics:

CRF Model:

- The CRF model achieved varying levels of precision, recall, and F1-score across different entity types.
- It showed strong performance for B-Degree (precision=1.00, recall=0.62, f1-score=0.77) and I-Degree (precision=0.95, recall=0.95, f1-score=0.95).
- However, it struggled with entities like B-Companies worked at (precision=0.65, recall=0.26, f1-score=0.37) and I-Companies worked at (precision=0.43, recall=0.12, f1-score=0.19), where precision and recall were lower.

SpaCy Model:

- The SpaCy model demonstrated competitive performance across most entity types.
- It excelled particularly in Name (precision=1.00, recall=0.83, f1-score=0.91), Email Address (precision=0.89, recall=0.89, f1-score=0.89), and Skills (precision=0.90, recall=0.50, f1-score=0.64).
- Notably, it achieved a high F1-score of 0.96 for the O (non-entity) category, indicating robust generalization.

BERT Model:

- The BERT model, while performing well on UNKNOWN and O categories with high precision and recall, struggled with lower support entity types.
- It achieved higher performance in Name (precision=0.38, recall=0.89, f1-score=0.53) and Designation (precision=0.54, recall=0.38, f1-score=0.45) compared to entities like College Name and Companies worked at.

Model Selection: SpaCy Model

Based on the comprehensive evaluation of precision, recall, and F1-scores across all entity types, we select the SpaCy model as the preferred model for automating the extraction and classification of entities from resumes. The SpaCy model not only demonstrated competitive performance metrics but also maintained consistent accuracy across a wide range of entities critical for resume screening.

Discussion

The choice of the SpaCy model aligns with its strengths in handling various entity types with high precision and recall. Its performance in extracting crucial information such as Name, Email Address, and Skills underscores its suitability for practical applications in HR automation. Future enhancements could involve fine-tuning the SpaCy model further on domain-specific data or exploring ensemble techniques to improve performance on specific entity types with lower scores.

By adopting the SpaCy model, HR departments can streamline the resume screening process, reducing manual effort and enhancing decision-making based on extracted information. This approach not only improves efficiency but also ensures consistency and accuracy in candidate evaluation.

Conclusion

After evaluating the three models, the SpaCy model was chosen for its superior performance in precision, recall, and F1-score across most entity categories. While the CRF model performed well, particularly for certain entities, and the BERT model showed potential, especially for entities like 'Designation' and 'Name,' the SpaCy model consistently provided the most balanced and reliable results overall. This model will significantly enhance the efficiency and accuracy of the resume screening process, making it faster and less prone to human error.

References

Devashishbhake. (2023, September 28). *Named entity recognition with bert*. Kaggle. https://www.kaggle.com/code/devashishbhake01/named-entity-recognition-with-bert

Gupta, M. (2024, February 1). *Named entity recognition(ner) using conditional random fields in NLP*. Medium. https://medium.com/data-science-in-your-pocket/named-entity-recognition-ner-using-conditional-random-fields-in-nlp-3660df22e95c

Kocaman, A. M. (2023, October 6). *Mastering named entity recognition with Bert: A comprehensive guide*. Medium. https://medium.com/@ahmetmnirkocaman/mastering-named-entity-recognition-with-bert-a-comprehensive-guide-b49f620e50b0

Majumder, P. (2023, September 13). *Named entity recognition (NER) in Python with spacy*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2021/06/nlp-application-named-entity-recognition-ner-in-python-with-spacy/

Mohamedtaha. (2021, June 27). *NER on resumes using CRF*. Kaggle. https://www.kaggle.com/code/mohamedtaha7/ner-on-resumes-using-crf

Mohamedtaha7. (2021, June 27). *NER on resumes using spacy*. Kaggle. https://www.kaggle.com/code/mohamedtaha7/ner-on-resumes-using-spacy

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