SwiftHire: Accelerating Recruitment with NLI Entity Recognition and Skills Matching

Project Report

Submitted by

Anila K R

ABSTRACT

This project explores the application of natural language processing (NLP) techniques, primarily using spaCy, for the analysis of resumes to aid recruiters in efficiently processing a large volume of job applications. The dataset comprises over 200 resumes sourced from livecareer.com, categorized into diverse job roles such as HR, IT, teaching, and more. Additionally, a Jobzilla skill dataset is utilized to enhance entity recognition within the resumes. Key steps involve data loading, entity recognition, skills extraction, text cleaning, and visualization of job category distributions and skill trends. Features such as custom entity recognition, resume analysis with match scores, and topic modeling using Latent Dirichlet Allocation (LDA) are implemented. The project concludes with reflections on learnings and potential future enhancements, including the integration of classification models and advanced visualization techniques. Overall, this project offers valuable insights and tools for streamlining the recruitment process through automated resume analysis.

INTRODUCTION

In today's competitive job market, recruiters often face the daunting task of sifting through a vast number of resumes to identify suitable candidates for various job roles. This process can be time-consuming and labor-intensive, leading to inefficiencies in the hiring process. To address this challenge, the application of natural language processing (NLP) techniques has emerged as a promising solution. By leveraging NLP tools such as spaCy, recruiters can automate the analysis of resumes, extract relevant information, and identify key skills and qualifications.

In this project, we embark on a journey to explore the capabilities of spaCy and other NLP tools for resume analysis. Our primary objective is to develop a system that can assist recruiters in quickly and accurately assessing resumes, thereby streamlining the recruitment process. We utilize a dataset consisting of over 200 resumes sourced from livecareer.com, covering a diverse range of job categories including HR, IT, teaching, and more. Additionally, we incorporate a Jobzilla skill dataset to enhance the entity recognition capabilities of our system.

Throughout the project, we follow a systematic approach, beginning with data loading and preprocessing, followed by entity recognition, skills extraction, and text cleaning. We then visualize job category distributions and skill trends to gain insights into the dataset. Furthermore, we implement advanced features such as custom entity recognition, resume analysis with match scores, and topic modeling using Latent Dirichlet Allocation (LDA). These features not only enhance the efficiency of the recruitment process but also provide valuable insights for recruiters and job seekers alike.

In conclusion, this project represents a significant step forward in leveraging NLP techniques for resume analysis. By automating tedious tasks and providing actionable insights, our system aims to revolutionize the way recruiters approach candidate evaluation. Through continuous refinement and future enhancements, we aspire to further enhance the capabilities of our system and contribute to the advancement of recruitment practices in the digital age.

GENERAL BACKGROUND

The process of recruitment and candidate evaluation is a critical aspect of human resource management for organizations across various industries. Traditionally, recruiters manually review resumes to identify suitable candidates based on their skills, qualifications, and experience. However, as the volume of job applications continues to rise, recruiters are faced with the challenge of efficiently processing large amounts of textual data within limited timeframes.

Natural Language Processing (NLP) has emerged as a powerful tool for automating text analysis tasks, including resume parsing and entity recognition. NLP techniques enable computers to understand, interpret, and generate human language, making it possible to extract valuable insights from unstructured text data such as resumes.

SpaCy is a popular open-source NLP library that provides robust support for various NLP tasks, including tokenization, part-of-speech tagging, named entity recognition (NER), and dependency parsing. By leveraging spaCy's advanced capabilities, developers can build sophisticated NLP applications for tasks ranging from information extraction to text classification.

In recent years, the application of NLP in recruitment and human resource management has gained traction, with organizations increasingly adopting automated tools and systems to streamline their hiring processes. These tools help recruiters identify top candidates more efficiently, reduce bias in candidate selection, and improve overall hiring outcomes.

By harnessing the power of NLP, organizations can gain valuable insights into candidate profiles, identify key skills and qualifications, and make data-driven decisions in the recruitment process. Additionally, NLP-powered systems can enhance the candidate experience by providing personalized feedback and recommendations, ultimately contributing to a more efficient and effective recruitment process.

Overall, the integration of NLP techniques such as entity recognition, text analysis, and topic modeling into recruitment workflows has the potential to revolutionize the way organizations identify and hire talent, driving greater efficiency, accuracy, and fairness in the hiring process.

SCOPE OF THE PROJECT

The scope of this project revolves around leveraging natural language processing (NLP) techniques, primarily utilizing the spaCy library, for the analysis of resumes with the objective of aiding recruiters in streamlining the candidate evaluation process. Initially, the project involves acquiring and preparing a dataset comprising over 200 resumes sourced from livecareer.com, covering a diverse array of job categories. Following data preparation, the focus shifts to entity recognition and skills extraction, employing spaCy to identify key entities such as skills, job titles, and qualifications within the resume text, and subsequently extracting skills using custom NLP patterns and rules.

Moving forward, the project encompasses visualization and analysis tasks, including the visualization of job category distributions and skill trends using histograms and word clouds to glean insights into the dataset. Custom visualization techniques are implemented to display entity recognition results and dependencies within the resume text. Additionally, features are developed to enable users to input their resumes for analysis, facilitating skills matching and calculating percentage match scores to assess a resume's suitability for a given job description based on required skills.

Furthermore, the project incorporates topic modeling with Latent Dirichlet Allocation (LDA) to identify latent topics within the resume dataset. Topics are visualized using pyLDAvis to provide insights into the thematic content of the resumes. In conclusion, the project summarizes its findings, reflecting on the effectiveness of NLP techniques for resume analysis, and outlines potential future enhancements, including the integration of classification models for predicting job categories and T-SNE visualization for exploring topic clusters. Overall, the project aims to provide a comprehensive exploration of NLP techniques for resume analysis, offering valuable insights and tools for recruiters, job seekers, and NLP practitioners.

LITERATURE SURVEY

1. "Named Entity Recognition with NLTK and SpaCy" by Susan Li: This article provides an overview of named entity recognition (NER) techniques using NLTK and spaCy libraries. It covers the basics of NER, compares NLTK and spaCy for NER tasks, and discusses practical applications in text analysis.

2. "How I used NLP (Spacy) to screen Data Science Resumes" by Venkat Raman:** In this article, the author shares their experience of using spaCy for resume screening in the context of data science roles. It highlights the challenges faced in resume analysis and demonstrates how NLP techniques can be applied effectively to extract relevant information from resumes.

3. "Resume and CV Summarization and Parsing with SpaCy in Python" by GitHub user "datasciencelearner": This GitHub repository provides code examples and tutorials for parsing and summarizing resumes using spaCy in Python. It offers practical insights into resume analysis techniques and demonstrates the implementation of NLP-based resume parsing.

4. "AI Models for Automatic Job Application Pipeline" by GitHub user "zake7749": This GitHub repository presents a comprehensive pipeline for automating job application processes using AI models. It covers various stages of the recruitment process, including resume analysis, job description analysis, and artificial cover letter generation, demonstrating the potential of NLP in recruitment workflows.

5. "SpaCy Tutorial | SpaCy For NLP | SpaCy NLP Tutorial" on Analytics Vidhya: This tutorial provides an in-depth exploration of spaCy for natural language processing tasks. It covers spaCy installation, tokenization, part-of-speech tagging, named entity recognition, and dependency parsing, offering a comprehensive understanding of spaCy's capabilities for NLP applications.

IMPLIMENTATION

Dataset

livecareer.com resume Dataset

A collection of 2400+ Resume Examples taken from livecareer.com for categorizing a given resume into any of the labels defined in the dataset: Resume Dataset.

Inside the CSV

- ID: Unique identifier and file name for the respective pdf.
- Resume_str : Contains the resume text only in string format.
- Resume_html : Contains the resume data in html format as present while web scrapping.
- Category : Category of the job the resume was used to apply.

Present categories

HR, Designer, Information-Technology, Teacher, Advocate, Business-Development, Healthcare, Fitness, Agriculture, BPO, Sales, Consultant, Digital-Media, Automobile, Chef, Finance, Apparel, Engineering, Accountant, Construction, Public-Relations, Banking, Arts, Aviation

Acknowledgements

Data was obtained by scrapping individual resume examples from www.livecareer.com for categorizing a given resume into any of the labels defined in the dataset:

Loading

In this section, we are going to load the spaCy model, Resume Dataset, and Jobzilla skills dataset directly into the entity ruler.

Resume Dataset

- Using Pandas read_csv to read dataset containing text data about Resume.
- we are going to randomized Job categories so that 200 samples contain various job categories instead of one.
- we are going to limit our number of samples to 200 as processing 2400+ takes time.

Loading spaCy model

You can download spaCy model using python -m spacy en_core_web_lg Then load spacy model into nlp.

Loading spaCy model

- You can download spaCy model using python -m spacy en_core_web_lg
- Then load spacy model into nlp.

```
nlp = spacy.load("en_core_web_lg")
skill_pattern_path = "skill_patterns.jsonl"
```

Entity Ruler

To create an entity ruler we need to add a pipeline and then load the .jsonl file containing skills into ruler. As you can see we have successfully added a new pipeline entity_ruler. Entity ruler helps us add additional rules to highlight various categories within the text, such as skills and job description in our case.

```
ruler = nlp.add_pipe("entity_ruler")
ruler.from_disk(skill_pattern_path)
nlp.pipe_names
```

```
['tok2vec',
 'tagger',
 'parser',
 'attribute_ruler',
 'lemmatizer',
 'ner',
 'entity_ruler']
```

Skills

We will create two python functions to extract all the skills within a resume and create an array containing all the skills. Later we are going to apply this function to our dataset and create a new feature called skill. This will help us visualize trends and patterns within the dataset.

- get_skills is going to extract skills from a single text.
- unique_skills will remove duplicates.

Cleaning Resume Text

We are going to use nltk library to clean our dataset in a few steps:

We are going to use regex to remove hyperlinks, special characters, or punctuations.

- Lowering text
- Splitting text into array based on space
- Lemmatizing text to its base form for normalizations
- Removing English stopwords
- Appending the results into an array.

Applying functions

In this section, we are going to apply all the functions we have created previously

- creating Clean_Resume columns and adding cleaning Resume data.
- creating skills columns, lowering text, and applying the get_skills function.
- removing duplicates from skills columns.

Visualization

Now that we have everything we want, we are going to visualize Job distributions and skill distributions.

```
fig = px.histogram(
    data, x="Category", title="Distribution of Jobs Categories"
).update_xaxes(categoryorder="total descending")
fig.show()
```

Jobs Distribution

As we can see our random 200 samples contain a variety of job categories. Accountants, Business development, and Advocates are the top categories.

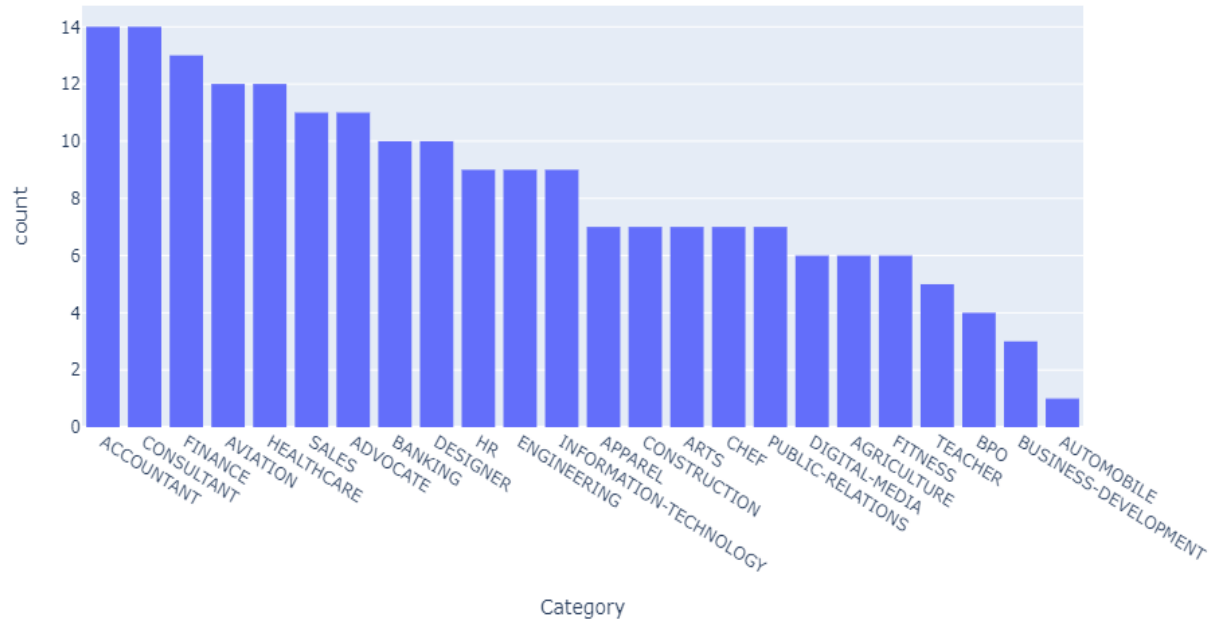

Fig 1: Distribution of Job Categories

Skills

First, we need to create variables from the dataset using unique() and then add the ALL category so that we can also visualize the overall skills trend in the dataset. lastly, we are going to use input cells and import categories from variables, as shown below.

```
Job_cat = data["Category"].unique()
Job_cat = np.append(Job_cat, "ALL")
```

```
import numpy as np
import plotly.express as px
Job_Category = "HR"
Job_cat = data["Category"].unique()
Job_cat = np.append(Job_cat, "ALL")

for category in Job_cat:
    Total_skills = []
    if category != "ALL":
        fltr = data[data["Category"] == category]["skills"]
        for x in fltr:
            for i in x:
                Total_skills.append(i)
    else:
        fltr = data["skills"]
        for x in fltr:
            for i in x:
                Total_skills.append(i)

    # Count occurrences of each skill
    skill_counts = {skill: Total_skills.count(skill) for skill in set(Total_skills)}

    # Plot histogram
    fig = px.bar(
        x=list(skill_counts.keys()),
        y=list(skill_counts.values()),
        labels={"x": "Skills", "y": "Count"},
        title=f"{category} Distribution of Skills",
    )
    fig.update_xaxes(categoryorder="total descending")
    fig.show()
```

INFORMATION-TECHNOLOGY job category's skills distributions.

Top Skills

- Software
- Support
- Business

If you are looking to improve your chance of getting hired by a software company try focusing on software engineering, Support, and Business skills.

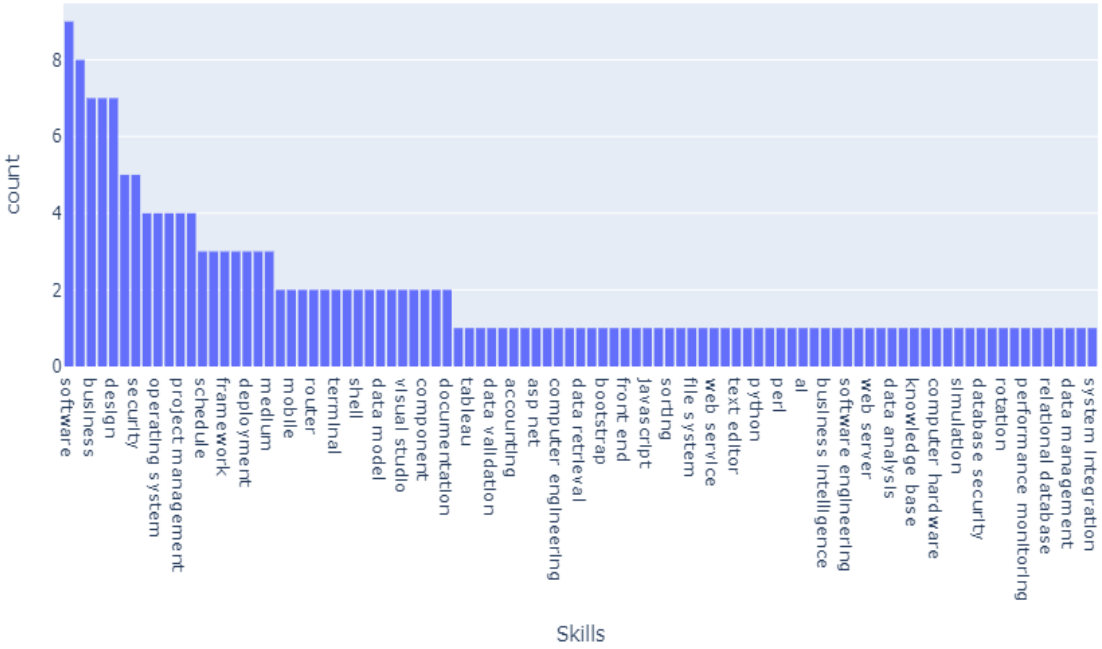

Fig 2: INFORMATION-TECHNOLOGIES Distribution Skills

Most used words

In this part, we are going to display the most used words in the Resume filter by job category. In Information technology, the most words used are system, network, and database. We can also discover more patterns by exploring the word cloud below.

Fig 3: Most used words in INFORMATION-TECHNOLOGY Resume

Entity Recognition

We can also display various entities within our raw text by using spaCy displacy.render. I am in love with this function as it is an amazing way to look at your entire document and discover SKILL or GEP within your Resume.

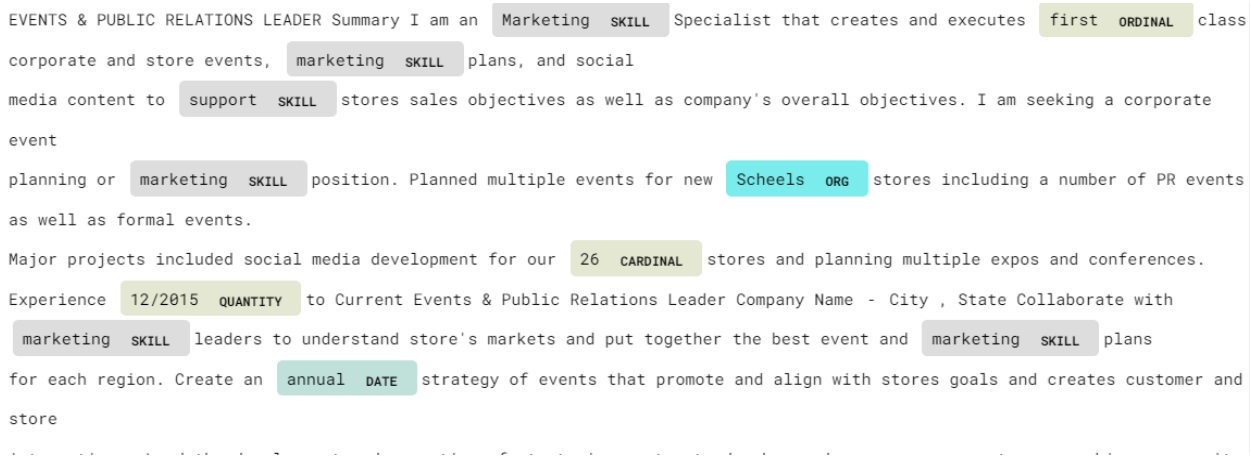

Fig 4: Entity Recognition of INFORMATION-TECHNOLOGY

Dependency Parsing

We can also visualize dependencies by just changing style to dep as shown below. We have also limited words to 10 which includes space too. Limiting the words will make it visualize the small chunk of data and if you want to see the dependency, you can remove the filter.

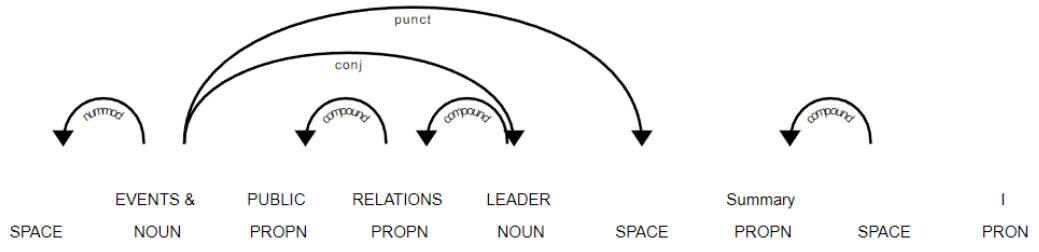

Fig 5:Dependency Parsing

Custom Entity Recognition

In our case, we have added a new entity called SKILL and is displayed in gray color. I was not impressed by colors and I also wanted to add another entity called Job Description so I started experimenting with various parameters within `displace.

- Adding Job-Category into entity ruler.
- Adding custom colors to all categories.
- Adding gradient colors to SKILL and Job-Category

```
patterns = df.Category.unique()
for a in patterns:
    ruler.add_patterns([{"label": "Job-Category", "pattern": a}])
```

```
# options=[{"ents": "Job-Category", "colors": "#ff3232"},{"ents": "SKILL", "colors": "#56c426"}]
colors = {
    "Job-Category": "linear-gradient(90deg, #aa9cfc, #fc9ce7)",
    "SKILL": "linear-gradient(90deg, #9BE15D, #00E3AE)",
    "ORG": "#ffd966",
    "PERSON": "#e06666",
    "GPE": "#9fc5e8",
    "DATE": "#c27ba0",
    "ORDINAL": "#674ea7",
    "PRODUCT": "#f9cb9c",
}
options = {
    "ents": [
        "Job-Category",
        "SKILL",
        "ORG",
        "PERSON",
        "GPE",
        "DATE",
        "ORDINAL",
        "PRODUCT",
    ],
    "colors": colors,
}
sent = nlp(data["Resume_str"].iloc[5])
displacy.render(sent, style="ent", jupyter=True, options=options)
```

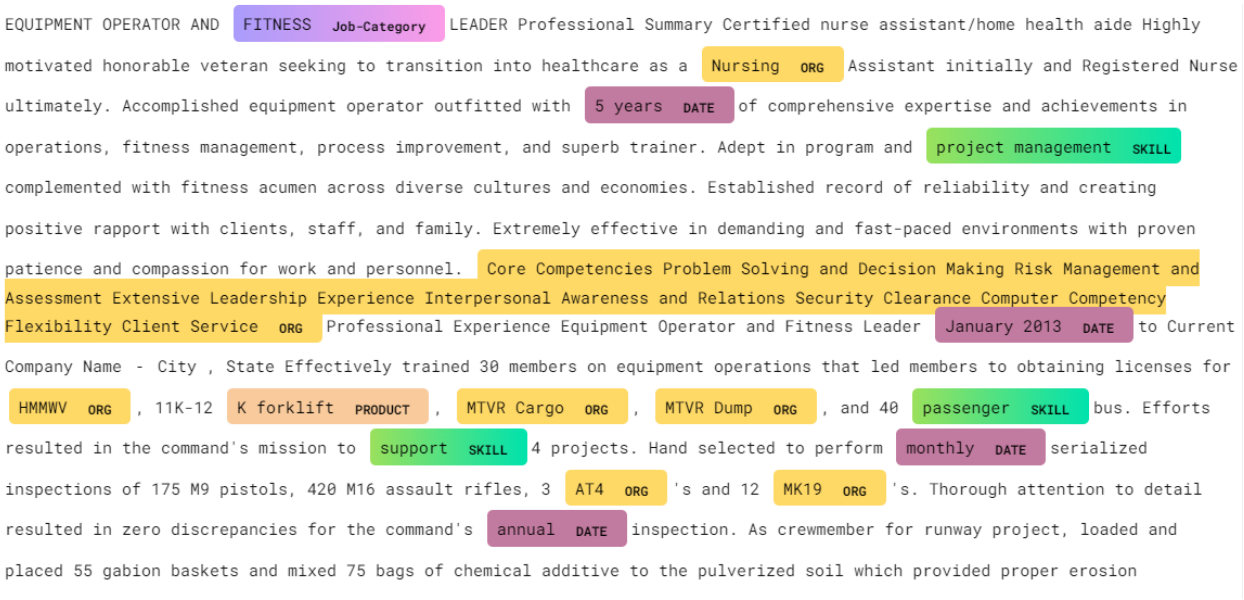

Fig 6: Custom Entity Recognition

Your Resume Anlaysis

In this part, I am allowing users to copy&paste their resumes and see the results.

As we can see my I have added my Resume and the results are amazing. The model has successfully highlighted all the skills.

Match Score

In this section, I am allowing recruiters to add skills and get a percentage of match skills. This can help them filter out hundreds of Resumes with just one button.

Please add the skills that are required by the job description without space in between commas and it will print out the percentage of match skills within the resume.

```
The current Resume is 66.7% matched to your requirements
```

We can also see the skills mentioned in your resume.

```
['testing', 'time series', 'speech recognition', 'simulation', 'text processing', 'ai', 'pytorch', 'communications', 'ml', 'enginee
```

Topic Modeling - LDA

LDA, or Latent Dirchlet Allocation is arguably the most famous topic modeling algorithm out there. Out here we create a simple topic model with 4 topics. The code was inspired by Allan's project: Topic Modeling of NLP GitHub repositories

```
docs = data["Clean_Resume"].values
dictionary = corpora.Dictionary(d.split() for d in docs)
bow = [dictionary.doc2bow(d.split()) for d in docs]
lda = gensim.models.ldamodel.LdaModel
num_topics = 4
ldamodel = lda(
    bow,
    num_topics=num_topics,
    id2word=dictionary,
    passes=50,
    minimum_probability=0
)
ldamodel.print_topics(num_topics=num_topics)
```

```
[(0,
  '0.015*"project" + 0.013*"system" + 0.008*"management" + 0.008*"company" + 0.008*"state" + 0.008*"city" + 0.007*"construction" + 0.007*"information" + 0.007*"technology" +
0.006*"name"'),
 (1,
  '0.015*"sale" + 0.014*"customer" + 0.012*"company" + 0.011*"state" + 0.011*"management" + 0.010*"city" + 0.010*"business" + 0.008*"name" + 0.008*"financial" + 0.007*"servi
ce"'),
 (2,
  '0.010*"state" + 0.008*"city" + 0.007*"company" + 0.007*"patient" + 0.007*"food" + 0.006*"team" + 0.006*"name" + 0.006*"service" + 0.006*"management" + 0.006*"staff"'),
 (3,
  '0.010*"state" + 0.009*"city" + 0.009*"company" + 0.007*"management" + 0.007*"name" + 0.007*"employee" + 0.005*"training" + 0.005*"program" + 0.005*"student" + 0.005*"offi
ce"')]
```

pyLDAvis

The best way to visualize Topics is to use pyLDAvis from GENSIM.

- topic #1 appears to relate to the customer, state, and city.
- topic #2 relates to management and marketing.
- topic #3 relates to systems and projects.
- topic #4 relates to financial and company.

```
pyLDAvis.enable_notebook()
pyLDAvis.gensim_models.prepare(ldamodel, bow, dictionary)
```

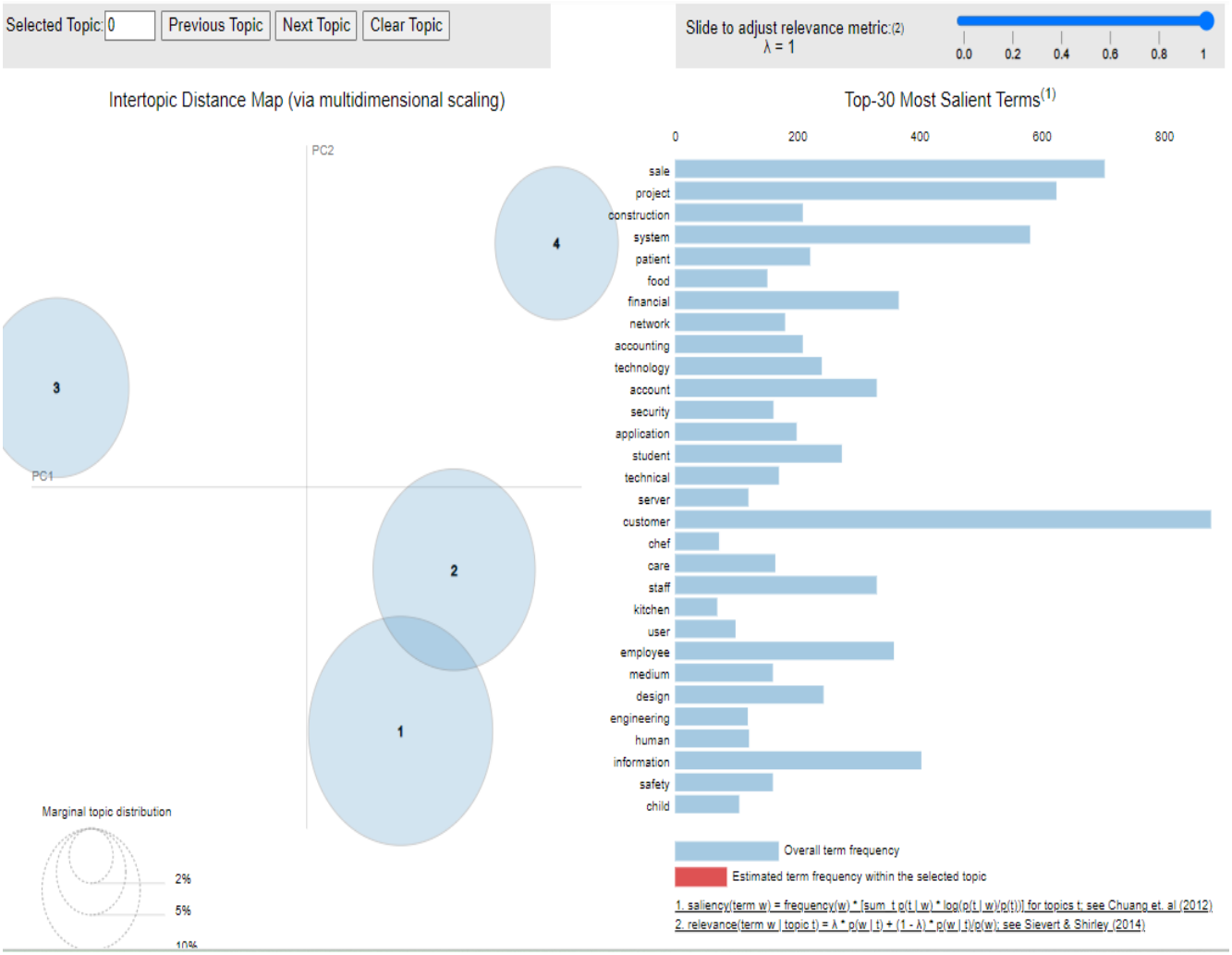

Fig 7: pyLDAvis

CONCLUSION

In this project, we have demonstrated the effectiveness of employing spaCy and other NLP tools for resume analysis, aiming to expedite the recruitment process for hiring managers. By leveraging entity recognition, skills extraction, and visualization techniques, we have provided valuable insights into job category distributions, skill trends, and topic modeling within the dataset of resumes. The implementation of custom entity recognition and match score calculation offers practical utility for recruiters in assessing candidate suitability based on required skills. Furthermore, the exploration of topic modeling using LDA enhances our understanding of underlying themes present in the resumes. Looking ahead, there is potential for further refinement and expansion of the project, including the integration of classification models and advanced visualization methods. Overall, this project serves as a valuable resource for recruiters, providing tools to efficiently navigate the resume screening process and identify top candidates for interview selection.