**Automated Resume Evaluation with NLP, SpaCy, and Topic Modelling**

Course Name– ENMG606 – Capstone Project

Instructor – Professor Jester Ugalde

Date – May 8th, 2024

**Team Members:**

Jagadeesh Chandra Prasad Sunkara

Kowshik Tata

Rahul Bojja

Anvesh Kumbam

**Introduction**

In today's job market, the resume screening process poses a considerable challenge, both in terms of time consumption and ensuring fairness. Traditional methods are labor-intensive and prone to human biases, resulting in inconsistencies during candidate selection. Our data science project aims to tackle these issues by leveraging advanced technology to streamline and improve the accuracy of matching candidates with job requirements.

Our project introduces an automated system that utilizes NLP, spaCy library, and topic modeling techniques for comprehensive text analysis of resumes. The main goals are to expedite the hiring process and ensure fair evaluation for all applicants. By automating resume evaluation, our system aims to reduce manual effort in recruitment and enhance overall efficiency.



This project can significantly change the hiring process. Our technology can better match candidate profiles with job requirements by combining machine learning and natural language processing (NLP) to grasp complex resume information. This improves recruiting decisions overall by expediting the hiring process and guaranteeing that candidates are more appropriate for their roles.

Our methodology involves analyzing two distinct datasets: resumes and a collection of defined skills relevant to different job categories. We rely on the spaCy toolkit, renowned for its NLP capabilities, to extract key information from these datasets. Additionally, we employ topic modeling to further refine our understanding of the data, ensuring a better match between candidate skills and job requirements.

In this report, we will delve into the methodologies used, discuss the system architecture, and showcase how these technologies have been applied to achieve automated, efficient, and fair hiring processes. Our aim is to not only innovate but also create a scalable model adaptable to various recruitment scenarios, potentially revolutionizing human resource management practices. Through these efforts, we aspire to contribute to a more equitable and effective employment landscape, fostering recognition of talent and accessibility of opportunities for all.

**Hypothesis / Business Use**

Our hypothesis is that using machine learning to automate resume screening and shortlisting can make the candidate selection process faster and easier. This method aims to improve recruitment processes and make hiring decisions better.

**Dataset**

For this project, we obtained the dataset from <https://www.livecareer.com/> , which includes over 2400 resume examples. Each resume is labeled according to the job category it corresponds to, with categories such as HR, Designer, information technology, Teacher, and more.

The dataset is structured as follows:

• ID: Unique identifier and file name for each resume in PDF format.

• Resume\_str: The resume text in string format.

• Resume\_html: The resume data in HTML format as scraped from the website.

• Category: The job category associated with each resume.

Link: <https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset>

we planned to gather the dataset by scraping resumes from the livecareer.com website. However, we couldn't complete this because of some restrictions and changes in the website's content.

**Data Cleaning**

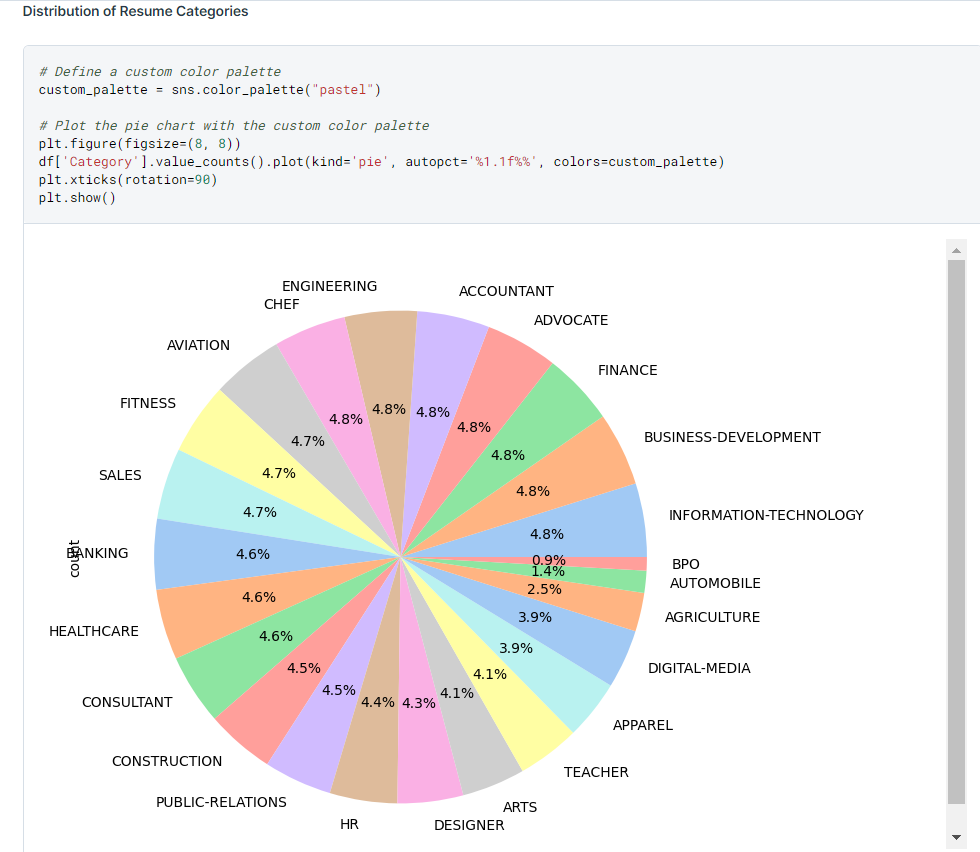
The dataset underwent several cleaning steps to ensure its suitability for analysis and modeling. The following activities were performed:

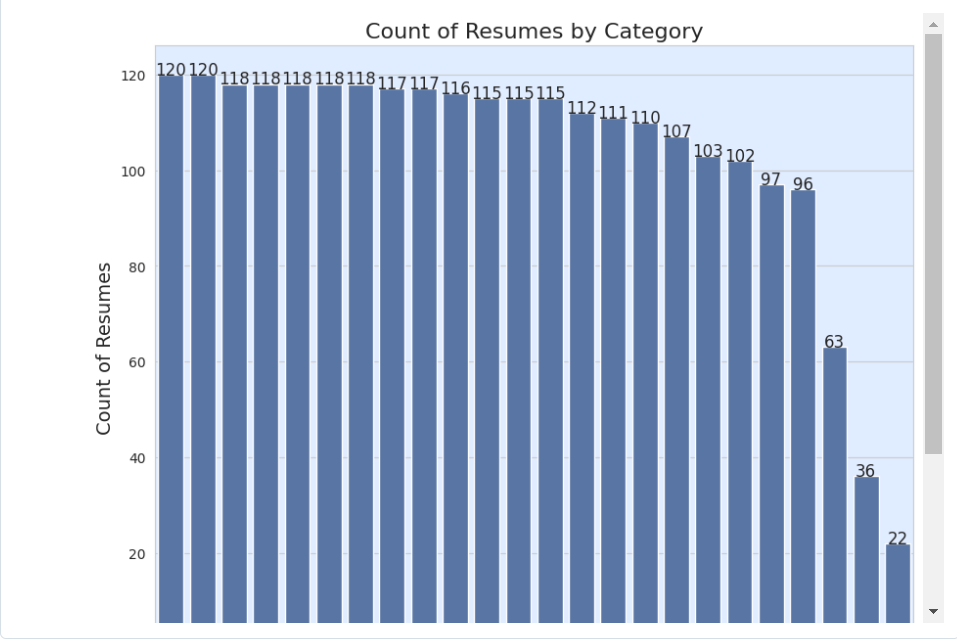
1. **Dropped Unused Columns:** Initially, we identified and dropped unused columns such as ID and Resume\_html, as they did not contribute to our analysis objectives.
2. **Preprocessing Text Data**: The text in the resumes had different things like symbols from other languages, punctuation marks, numbers, common words that do not add much meaning (like "the" or "and"), mentions of people or companies, hashtags, and web links. These things make it harder to analyze the text. So, we used tools from natural language processing (NLP) to help clean up the text. One technique we used is called stemming, which helps to simplify words by turning them into their basic form. This makes it easier to compare words and understand their meaning. Another technique we used is called word tokenization, which breaks down the text into individual words. This way, we can better analyze and work with the text data.
3. **Removing Commonly Used Words**: Certain words such as 'city', 'state', 'country', 'full name', and 'company' were identified as commonly occurring but not relevant to our modeling objectives. These words were removed to prevent them from causing noise and potentially affecting the accuracy of our models. These cleaning activities were crucial in preparing the dataset for subsequent analysis and modeling tasks. By eliminating unnecessary columns, standardizing text data, and removing irrelevant words, we ensured the integrity and quality of the dataset, laying a solid foundation for our analytical endeavors.

**Exploratory Data Analysis**

The exploratory data analysis conducted on the dataset aimed to uncover patterns and insights that could inform our machine-learning models. Key findings from the EDA include:

**Job Category Distribution:** The analysis revealed a varied distribution of resumes across different job sectors. This uneven distribution highlighted the popular and niche sectors, guiding us in tailoring our analysis tools effectively.



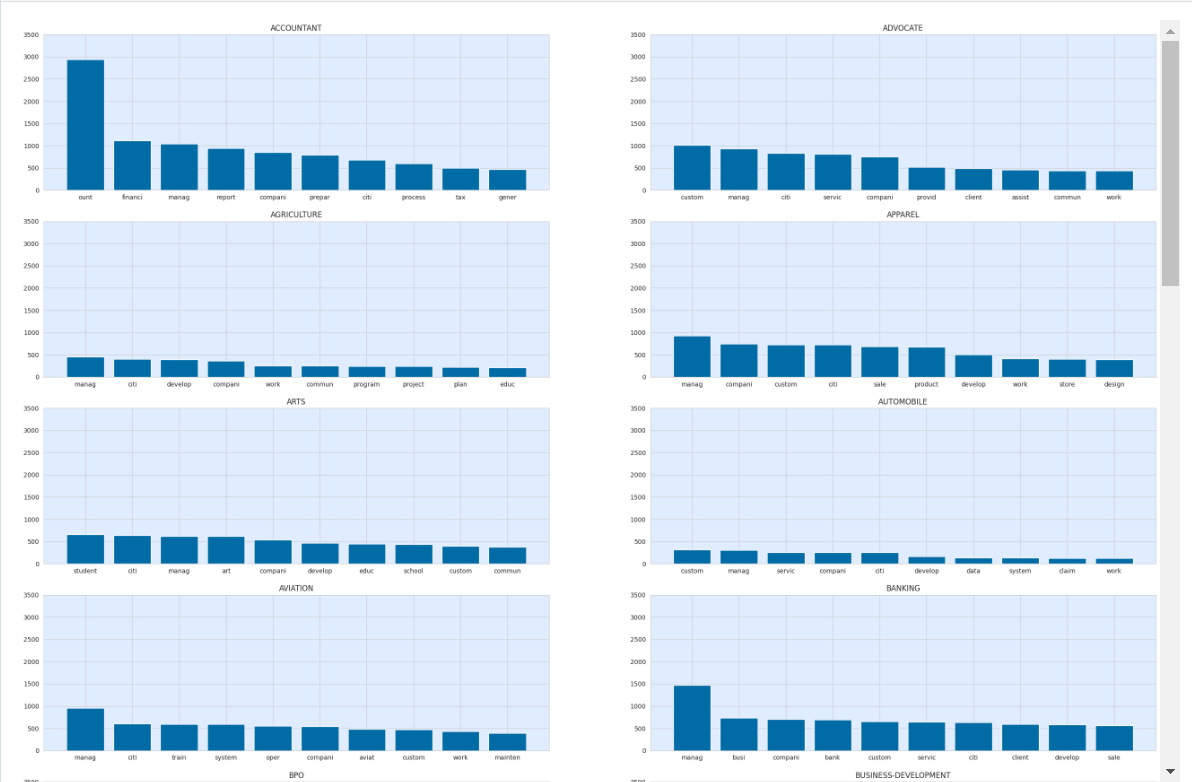


**Common Keywords Identification**: By identifying frequent keywords and phrases across resumes, we gained insights into prevalent skills and qualifications valued in different fields. This keyword analysis is crucial for aligning candidate profiles with job descriptions.





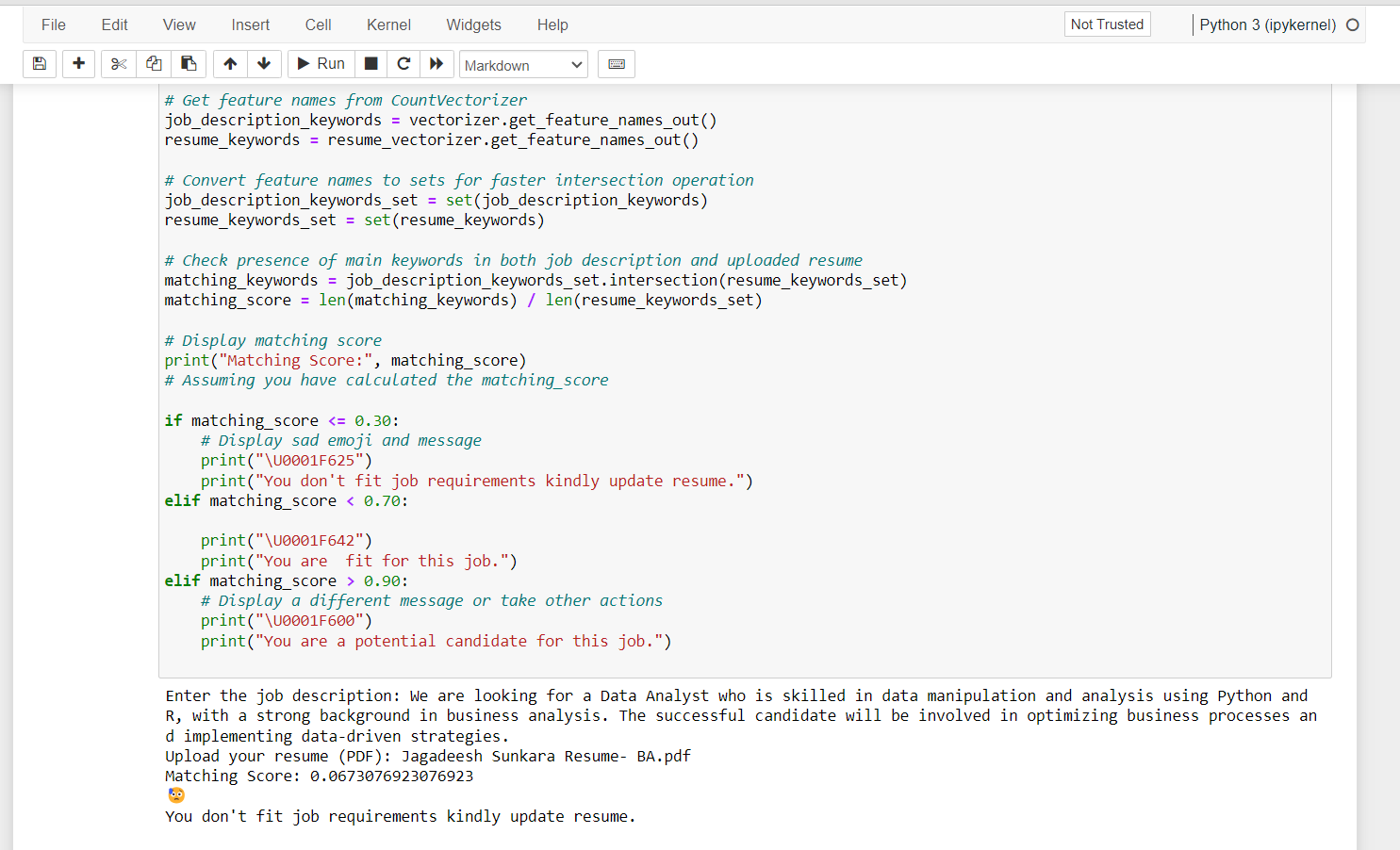
**Visualization Techniques**: Utilizing various visualization tools such as histograms, Pie charts, word clouds, and scatter plots, we were able to visually represent the distribution of job categories and the prevalence of certain keywords.



**Model Building**

The model is designed to compare a job description with an uploaded resume to see how well they match. It uses natural language processing (NLP) techniques to extract and analyze text from both the job description and the resume. Below is the mechanism:

1. **Extracting Text from PDF:** The model includes a feature called extract\_text\_from\_pdf that grabs text from a PDF resume file.
2. **Preprocessing:** The job description and resume text go through preprocessing to clean and standardize the data, ensuring the analysis is consistent and precise.
3. **Vectorization**: The job description text and the resume text are tokenized and vectorized using the CountVectorizer from scikit-learn. This process converts the text data into numerical vectors, representing the frequency of words in each document.
4. **Keyword Matching**: After vectorization, the model identifies common keywords between the job description and the uploaded resume. It calculates a matching score based on the presence of these keywords, indicating the extent of compatibility between the two documents.



**Result Interpretation**

Depending on how well the resume matches the job description, the model gives feedback to the user. Below are the details -

- Low Compatibility (Matching Score Below 0.30):

* A sad emoji appears.
* The user receives a message suggesting they update their resume to better match the job.

- Moderate Compatibility (Matching Score Between 0.30 and 0.70):

* A neutral emoji is displayed.
* The user is informed that their resume is a good fit for the job.

- High Compatibility (Matching Score Above 0.90):

* A happy emoji appears.
* The user is recognized as a strong candidate for the job.

**Spacy Modelling**

Outlined the detail the process of constructing a Resume Analyzer utilizing spaCy, a prominent natural language processing (NLP) library. The Resume Analyzer serves the purpose of analyzing raw resume texts, detecting pertinent skills mentioned within them, and computing a matching score against predetermined skillsets.

1. **Loading spaCy Model:**

The initial stage of constructing the Resume Analyzer involves loading the spaCy model. We opted for the en\_core\_web\_sm model in our method, which is a concise English pipeline tailored for web text.This model offers important NLP features like breaking text into tokens, identifying parts of speech, and analyzing sentence structure.

1. **Entity Ruler:**

We employed the Entity Ruler component in spaCy to recognize entities related to skills mentioned in resumes. The Entity Ruler allows us to define custom patterns for identifying entities in text. We created a pipeline that loads a list of skills from a .jsonl file and adds them to the Entity Ruler. This enables the model to identify specific skills mentioned in resumes.

1. **Entity Recognition:**

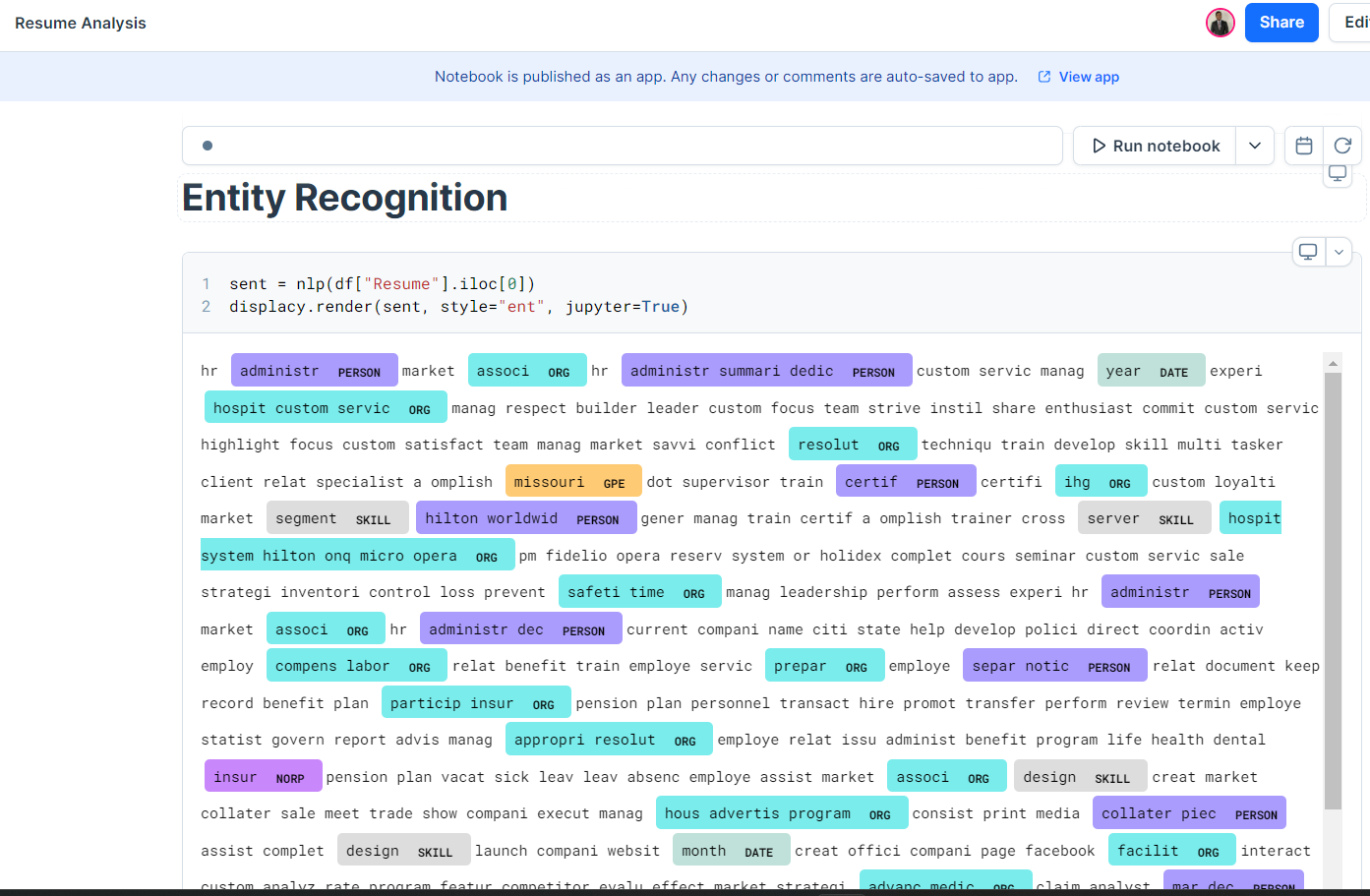
Once the Entity Ruler is added to the pipeline, the model can process raw resume texts and identify entities, including skills. We used spaCy's displacy.render function to visualize the recognized entities within the raw text. This step provides a clear understanding of how the model identifies skills in resumes.

1. **Resume Analysis and Matching Score:**

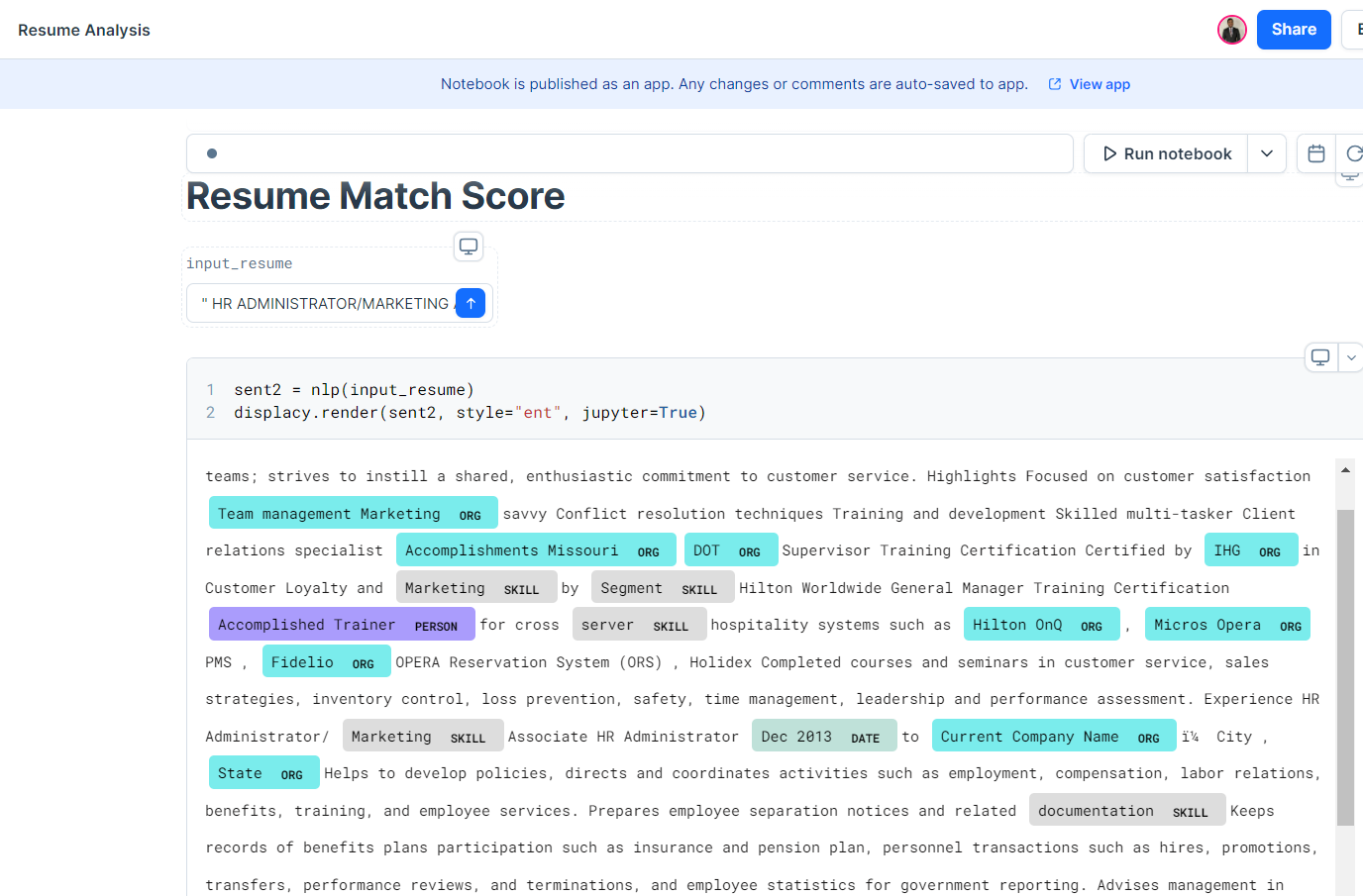
In the final step, we developed a function to analyze resumes and calculate a matching score based on the identified skills. The function processes a given resume text, matches the recognized skills against a predefined list of skills, and computes the matching score as a percentage. This score indicates the degree of similarity between the skills mentioned in the resume and the predefined skillset.

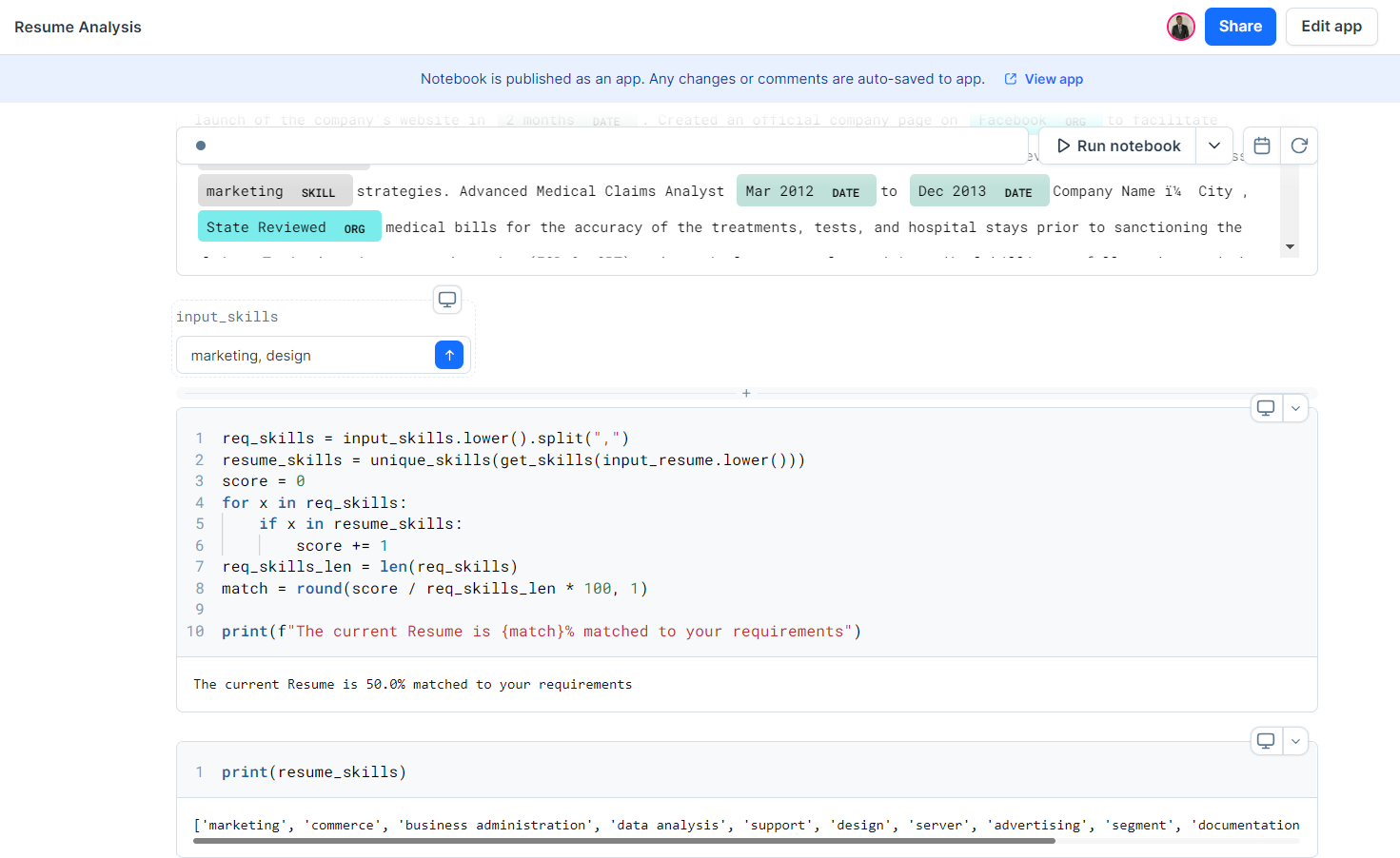
**Entity Recognition**

We used SpaCy's Named Entity Recognition (NER) to pull important information from resumes, like names, skills, job titles, and education details. After cleaning and standardizing the text, we applied and customized the NER model to pinpoint job-specific entities. This process enhanced our ability to analyze candidate qualifications and improved the accuracy of matching resumes with job descriptions.



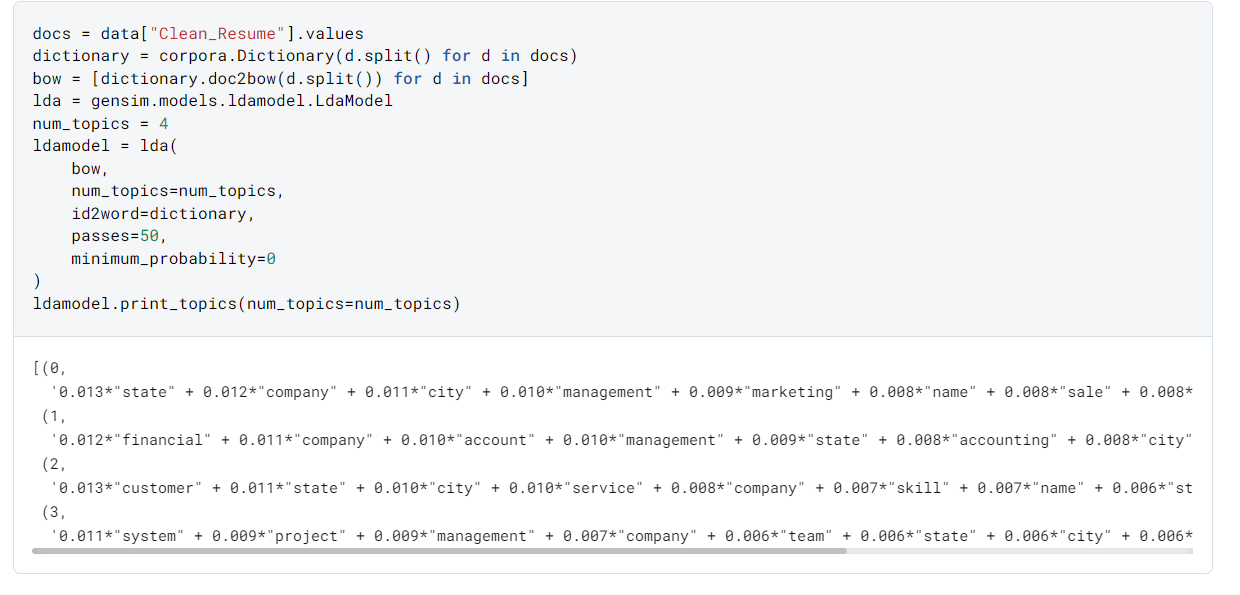
**Resume matching**



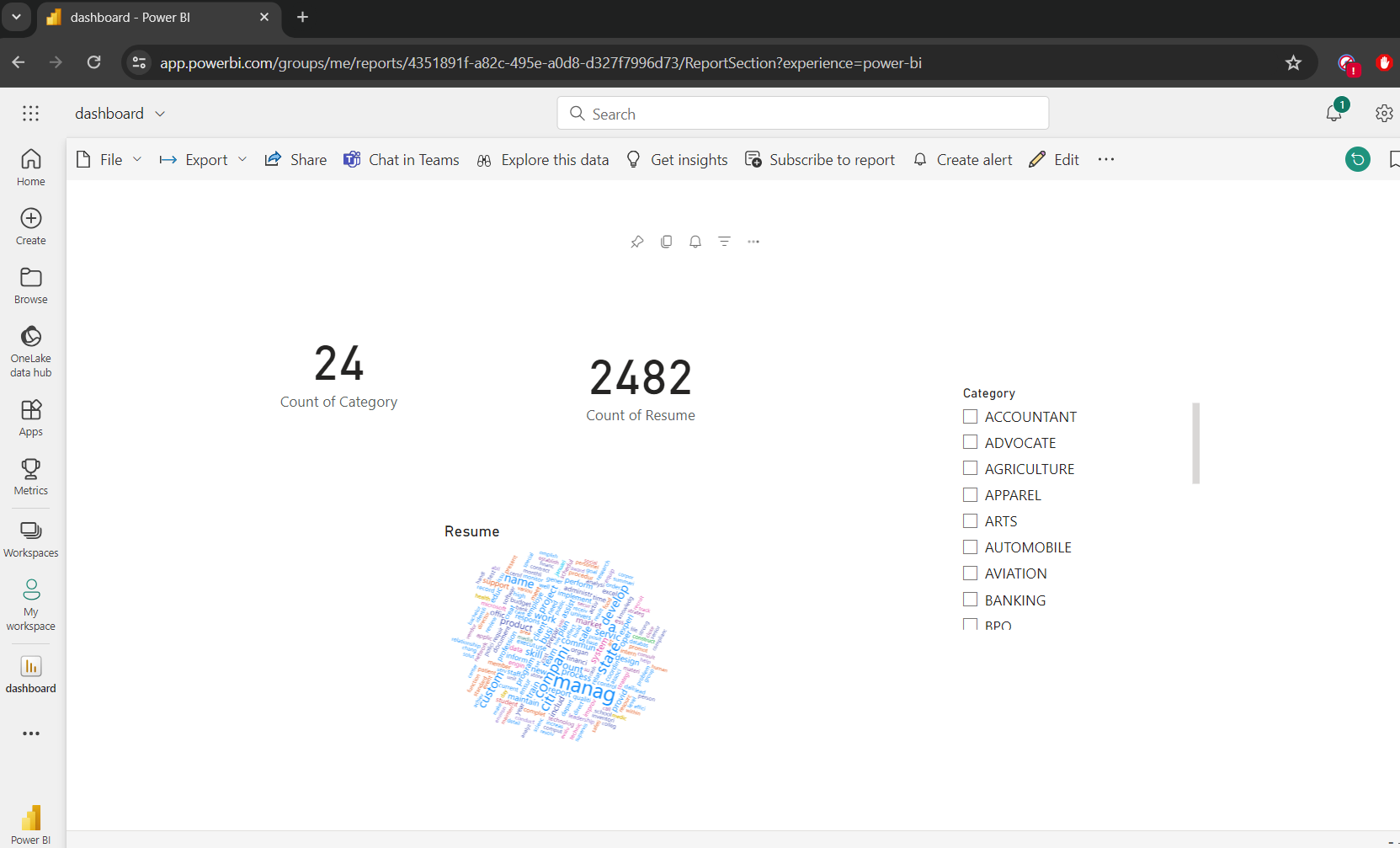


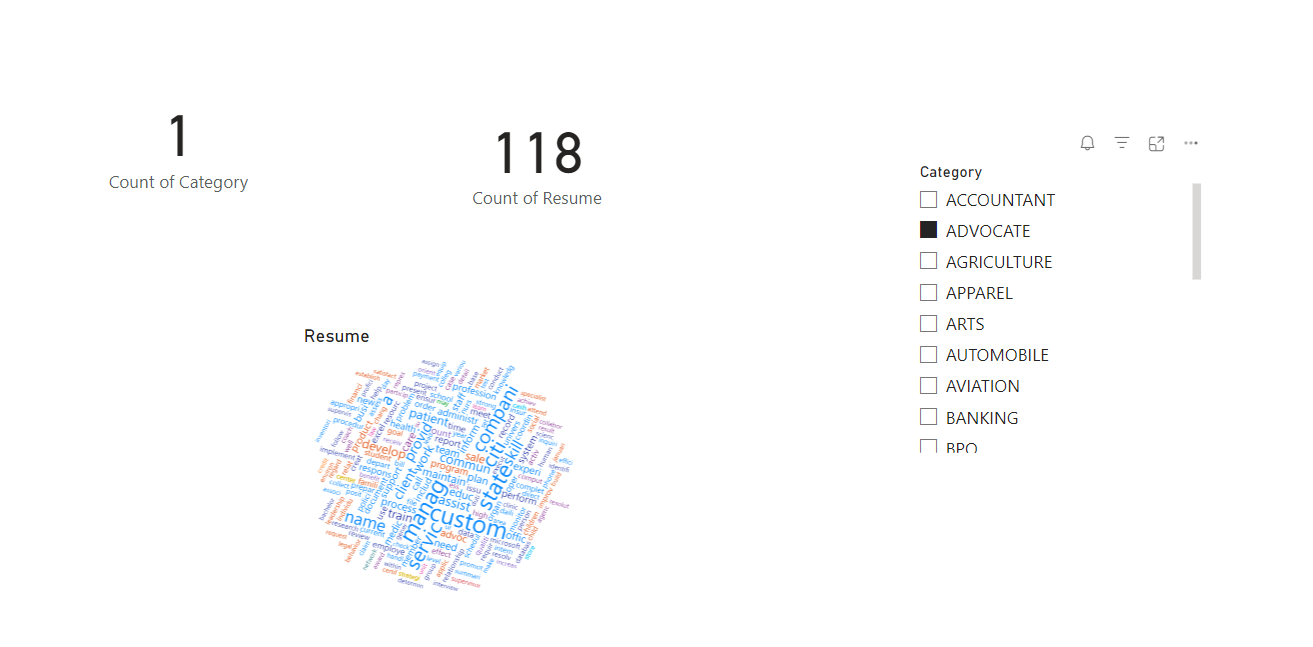
**Topic Modeling – LDA**

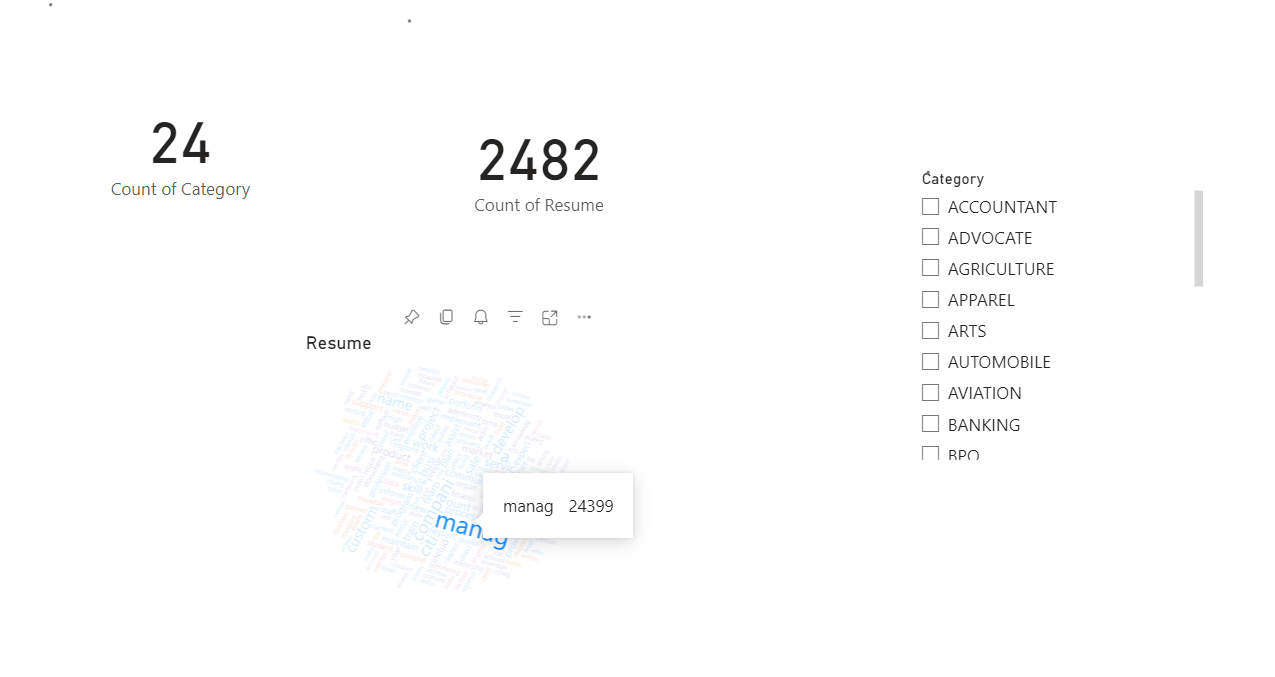
Topic modeling, particularly Latent Dirichlet Allocation (LDA), is a technique in natural language processing that helps find the hidden themes or subjects in a group of documents.



**Dashboard**

****

****

****

**Summary**

Through this project, we aimed to streamline the hiring process by developing an automated system for resume evaluation using SpaCy and topic modeling techniques. Despite initial plans to extract data directly from the LiveCareer website, we encountered challenges due to dynamic content and changes in website structure, which led us to use pre-existing scraped data instead.

We implemented an entity ruler to generate additional entities and customize their presentation with unique colors, enhancing the visualization of key resume information. By analyzing the distributions of job categories and skills, we enabled users to upload resumes and receive a skill match percentage, providing immediate feedback on their suitability for specific job roles. Additionally, we utilized Latent Dirichlet Allocation (LDA) for topic modeling to identify common themes and skills within the resumes.

This project provided a valuable learning opportunity to explore the capabilities of SpaCy and gain a comprehensive understanding of the data science pipeline. We covered essential aspects such as data cleaning, feature engineering, model evaluation, and automation. Through this experience, we recognized the significance of thorough data preprocessing, the effective application of natural language processing (NLP) techniques, and the potential of machine learning to improve efficiency and fairness in the recruitment process.

Overall, we successfully applied these advanced techniques to streamline recruitment and enhance candidate selection efficiency, showcasing the potential for technology to revolutionize traditional hiring practices.

**Source Code**

**GitHub:**

<https://github.com/Jagadeesh-Sunkara/Resume_Analysis_Capstone>

**Deepnote:**

<https://deepnote.com/app/capstone-resume-analysis/Resume-Analysis-b8d3f3f0-fa5d-48c0-8f8e-c97ad3726e67>

**References**

1. Deepnote. (n.d.). spaCy Resume Analysis. Retrieved from <https://deepnote.com/app/abid/spaCy-Resume-Analysis-81ba1e4b-7fa8-45fe-ac7a-0b7bf3da7826>
2. Dutta, G. (n.d.). Resume screening using machine learning [Notebook]. Kaggle. Retrieved May 14, 2024, from <https://www.kaggle.com/code/gauravduttakiit/resume-screening-using-machine-learning/notebook>
3. Alpaydin, E. (2020). Introduction to Machine Learning (4th Ed.). MIT Press. (Helpful for foundational concepts in machine learning and its applications)
4. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research, 3*, 993-1022.
5. Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.
6. Jurafsky, D., & Martin, J. H. (2020). Speech and Language Processing (3rd edition draft). (A comprehensive textbook on NLP and speech processing.