**Job and Resumé Screener**

Arshia, Tibor, Ruchit, Winston, Laurance

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Abstract:**

**This project explores the use of Natural Language Processing(NLP) and machine learning techniques as a time-efficient approach to rank qualified candidates for specific job positions. Resumé screening can be a time consuming process and our aim is to create an unbiased model that can speed up this process and accurately filter candidates based on their qualifications. Through research, we narrowed down an NLP approach utilizing the Bert, FastText\*, GloVe, ELMo, and Gpt2 models. Our method returns a ranked set of resumés using a dataset for both job and resumé postings.**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Background:**

Traditional approaches to Resumé Screening are often time consuming and inefficient considering the increase in volume of job applications. Often, in the resumé screening process, unconscious human biases may arise due to various factors such as race, gender, age, and more. Because of these reasons, turning to resumé screening automation could improve efficiency and remove biases.

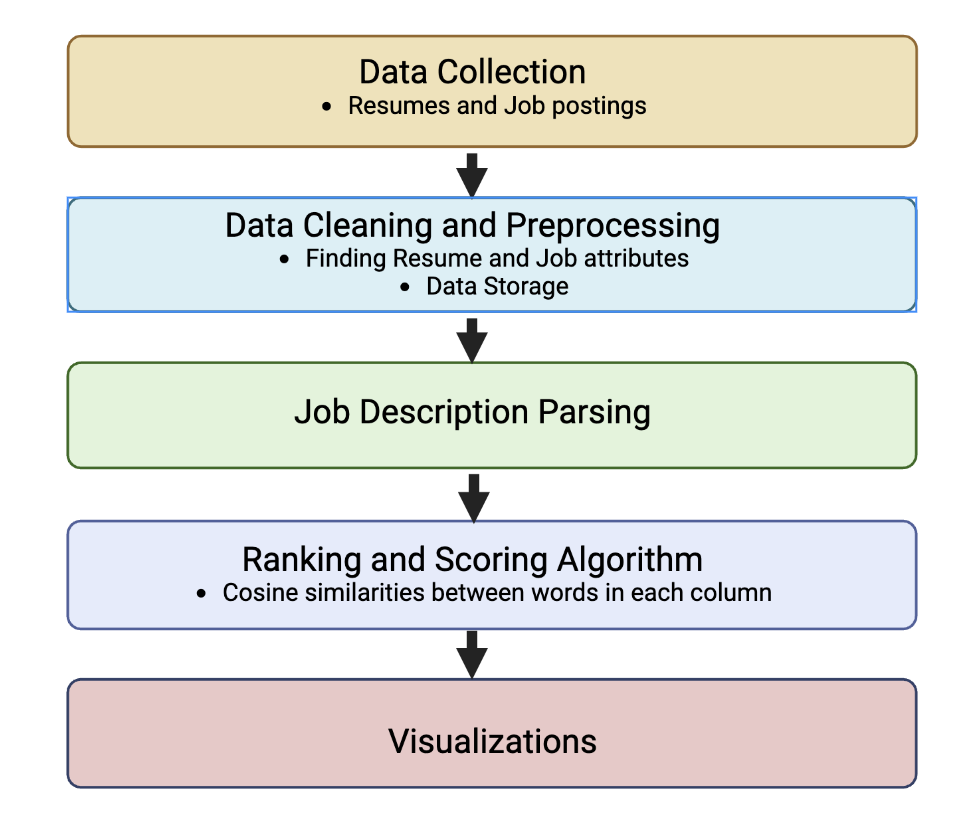
Through research, we identified several different approaches to screening resumés using both machine learning and Natural Language Processing techniques. For this project, it was essential that we could determine important job and resumé attributes to rank these candidates. For this reason, we decided to use a Natural Language Processing approach. Natural language processing is a branch of Artificial Intelligence that focuses on the interpretation of text and speech by computers. Word-embedding, cosine similarity, Euclidean Distance, and Pearson correlation coefficient are the main techniques we used to identify the best candidates. Our models of choice for this project are Bert, FastText\*, GloVe, ELMo, and GPT2.

**Methods:**

Our main goal of this resumé screener was to rank a set of given resumés that are most relevant to a given job description. To do this, we decided that the best approach was to compare a variety of similarity scores of word embeddings generated by several models. We decided to compare the following 5 models: GloVe, FastText, ELMo, BERT, and GPT2. We compared the word embeddings by combining the following similarity scores as a final weighted similarity score: Cosine Similarity, Euclidean Distance, Pearson Correlation Coefficient, and Manhattan Distance. We combined these similarity scores by performing the following calculation: Cosine Score - Euclidean Distance + Pearson Coefficient - Manhattan Distance. It is important to note that although FastText has been tested and works as expected, our final approach does not use FastText because it uses too much RAM.

In the end, there was also an attempt to build a CNN model that we could compare our model to, but because of time constraints, we were unsuccessful to fully complete it, but we were close though.

The following is a diagram of our process:

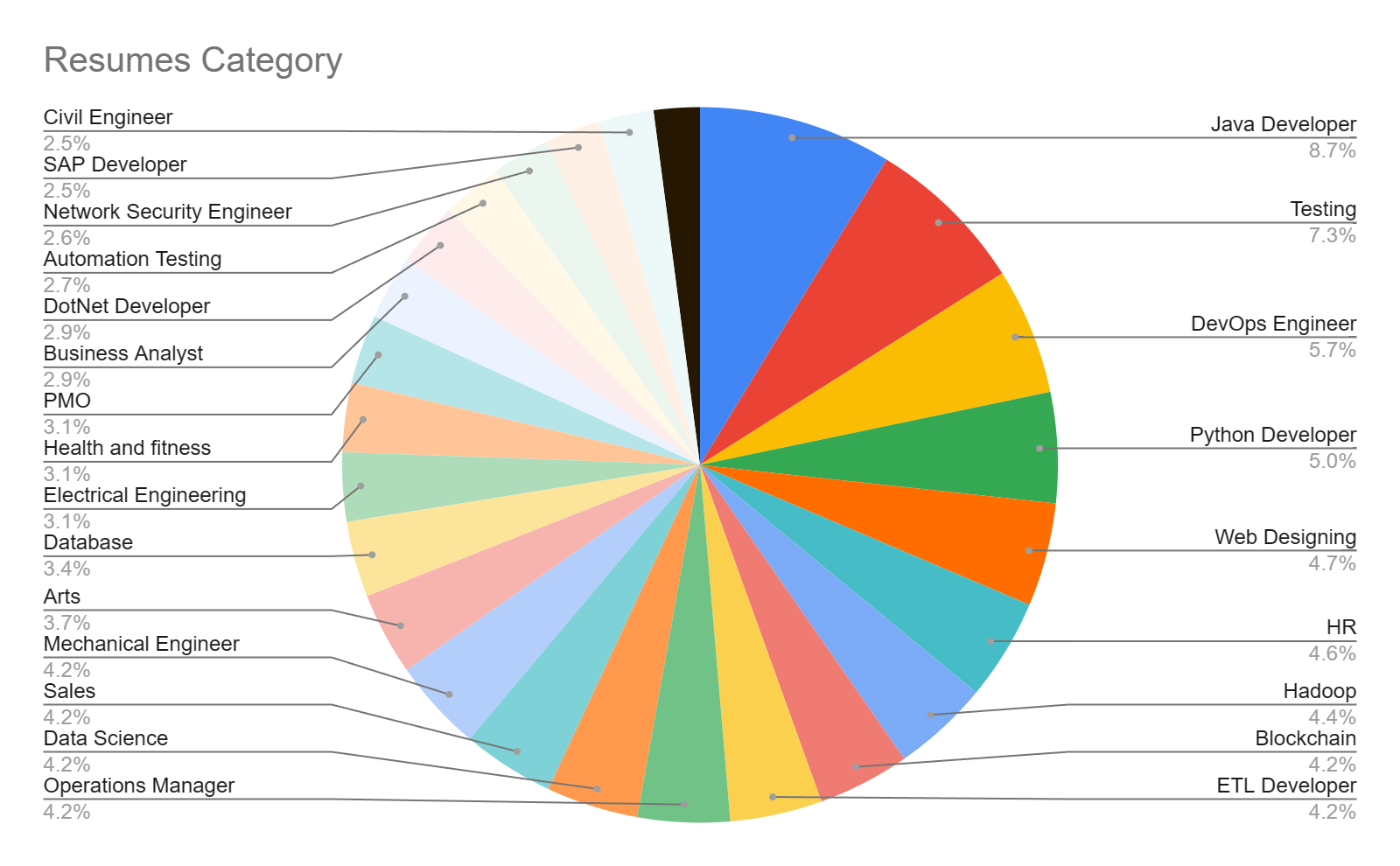


***Figure 1*. *Pipeline:*** This figure outlines our process for

Completing this project.

Our dataset contained 3446 resumés. We combined two resumé datasets we found on Kaggle. We used Regex to remove symbols, links, and punctuations. We then extracted features by tokenizing the resumés. This pool of resumés was then given to our 5 models for comparison.

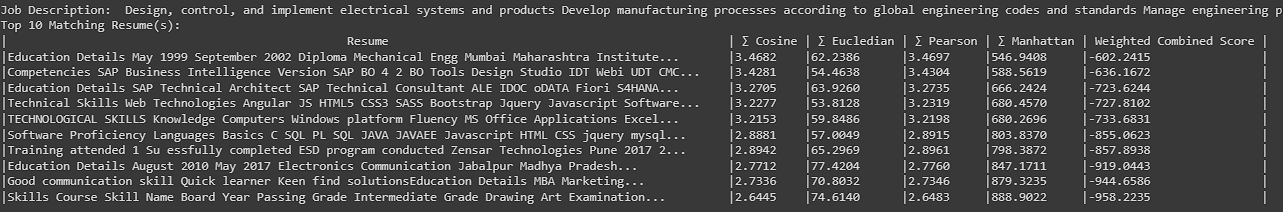
The following is a diagram showing the types of resumes contained within our dataset:



***Figure 2: Job categories:*** The diagram illustrates the split of the various job categories

from the data.

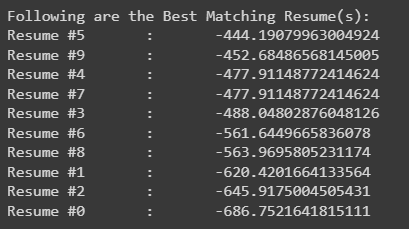
We chose this approach because we felt that only large language models were able to understand the complex relationships between words within resumés/job descriptions. Additionally, we chose to combine the 4 similarity scores to get the most accurate similarity score.

**Results:**

***Figure 3. Example best resumé candidate:*** The screenshot is an example output from our model that displays the best resumés for a specific electrical engineering position.

We were very happy with the results. The following is an image of our model finding the best resumés from our resumé dataset for an electrical engineering position:

What the above image shows is the top ranked resumés (in order of relevance), their individual similarity score, and their individual combined score. The following image is a summary representation of a similar output to the once shown above:



***Figure 4: Example Ranking:*** This figure illustrates the scores for a job

position of the relevant resumés

**Discussion/Limitations:**

Unfortunately there are a few limitations to our approach. The largest of which is that we ourselves determined the accuracy of ranked training resumés. This means that the model has bias that reflects our biases as students. If we were to ever market our model as a serious tool for professional recruiters, we would need to have recruiters help train our model. To combat this, we attempted to implement a cnn approach that could be used to compare the results of the model to.

Another limitation of our model is its questionable ethics. This resumé is a black-box, and as a result, it is impossible to determine with certainty why a resumé was accepted or rejected. This can leave candidates discouraged. Furthermore, this model only ranks textual similarity, and doesn't consider factors such as culture fit, ability to learn more skills, and other soft skills. Also, our model favors candidates who are specialized in a certain field, rather than someone with a more wide range of skills. Someone who is equally capable in several domains is considered less favorable to a candidate capable in a singular domain. These problems mean that potentially good applicants could be ignored. Because of this we believe that the best use of our model should be to filter out the truly unqualified candidates. For a more nuanced analysis of resumés, we think a recruiter would do a better job.

**Conclusion:**

In summary, our team aimed to create a resumé screening tool that ranks the best candidates by relevance to a given job description. The purpose of our work, besides learning more about ML, is to help reduce bias and screening time when recruiters are searching for the most relevant candidates. Currently, some job markets are becoming so competitive that resumé screening tools have become a necessity. Understanding the significance of resumé screeners in the tech industry, we hoped to learn more about them to potentially gain insight on how our resumés can be written to better perform compared to the competition. In this regard, the key takeaway was to make sure we use lots of relevant keywords from the job posting in our resumés.

Although our model has potential, we believe that more work needs to be done. Notably, we trained our model based on what us students thought were the most relevant resumés. We are no experts in recruiting and as a result, we created bias in our model. In order to increase the accuracy of our model and reduce bias we would need to work with professional recruiters. In addition, we would also need to work on better data cleaning techniques and optimizing our model. In its current state, our model serves as a learning tool, and would not be able to be easily used by a recruiter in a professional setting. It has no user interface and no user-oriented features.

Although there is more potential work to be done, we are very happy with the current state of our model. We learned more about data processing techniques, transformer models, similarity scores, data visualizations, and how to bring all these ML techniques together to build a useful product. As a result of this class and project, some of us have decided to look further into ML as a potential career path for themselves. Additionally, some of us will be taking other ML classes offered at UCSC to further extend our knowledge.

**Contributions:**

Arshia - Attempting a CNN model that we could compare the results of our model to.

Winston - Implemented document cosine similarity, approach breakdown

Ruchit - Implementing Pre-trained model, scores, and TF-IDF visualizations.

Laurance - Visualization and Data Analysis

Tibor - Combining and preprocessing resume datasets.

**Bibliography:**

The following is a list of relevant resources we used during this project. They are organized in general categories:

**Resume Parser**

Resume Parser : <https://www.kaggle.com/datasets/prudhviram/resumeparsermp>

Extract features : <https://github.com/OmkarPathak/pyresparser>

Extractor : <https://github.com/bjherger/ResumeParser>

To JSON : <https://github.com/gogsbread/ResumeParser>

With Constraints : <https://github.com/OmkarPathak/ResumeParser>

Rank Resumes : <https://github.com/Satrat/Resume-Parser>

From PDF : <https://github.com/jineshdhruv8/ResumeParser>

All Formats : <https://github.com/likerRr/code4goal-resume-parser>

JS w lever API : <https://github.com/KnlnKS/resume-parser>

JS : <https://github.com/perminder-klair/resume-parser>

**Resume Job Matching**

Similarity Score Matching : <https://github.com/amiradridi/Job-Resume-Matching>

Rank Based (Similar appr) : <https://github.com/saurabhraidev/Naive-Resume-Matching>

CNN approach : <https://github.com/ruozhengu/job-resume-matching-algo>

Naive : <https://github.com/jianfeiZhao/Resume-Matching-System>

Naive (manual) scores : <https://github.com/faizan170/resume-job-match-nlp>

NLP Genism : <https://github.com/ahnineamine/Job-Matcher>

TF-IDF approach : <https://github.com/saurabhraidev/Naive-Resume-Matching>

KNN Classification : <https://thecleverprogrammer.com/2020/12/06/resume-screening-with-python/>

**Resume Datasets Used**

[https://www.kaggle.com/datasets/snehabhawal/resume-dataset](https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset)

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

**Python Library to Extract Data From Resumes:**

<https://pypi.org/project/pyresparser/>

<https://pypi.org/project/resume-parser/>