Automated Resume Screening

Anuja Prakash Kolse[1], Oum Parikh[1], and Alexander Wilcox[1]

[1]Khoury College of Computer Sciences, Northeastern University

Abstract

Our project leverages advanced unsupervised natural language processing (NLP) technologies to enhance the efficiency and fairness of the resume evaluation process by automating the comparison of resumes across various professions using similarity scores against job descriptions and exemplary resumes. We aim to refine candidate selection by optimizing time management and reducing unconscious biases. Through a comprehensive exploration of vectorization methods like Word2Vec, TF-IDF, and GloVe, we transform resumes into numerical embeddings to train a sophisticated classifier capable of accurately assigning professional categories to resumes, achieving effective computation of similarity scores and enhancing automated resume screening.

1 Introduction

In the rapidly evolving job market, resume screening has become a complex, labor-intensive task for HR departments. Traditional methods are slow and susceptible to bias, resulting in inefficiencies and unfairness. Our project seeks to transform resume evaluation using unsupervised NLP technologies. We employ advanced NLP techniques to automate resume comparison across different professions, calculating similarity scores between candidate resumes and job descriptions or model resumes. This system offers a more objective and efficient approach to initial candidate selection, improving time management for HR professionals and reducing bias in the evaluation process.

Our research involves various vectorization methods to transform textual resume data into numerical embeddings. These embeddings form the basis for our classification model, which can categorize resumes by professional field. We utilize cutting-edge NLP techniques like Word2Vec, TF-IDF, and GloVe, each providing distinct benefits in capturing the semantic depth of resume content. Through this project, we aim to showcase the effectiveness of our automated resume screening system in improving the accuracy and fairness of candidate selection. By leveraging unsupervised NLP, we strive to establish a new benchmark in recruitment, making it more efficient, objective, and inclusive.

2 Methodology

2.1 Dataset

For our project, we utilized two Kaggle datasets containing industry-labeled resumes from a wide range of professions and job roles [1, 2]. This dataset was carefully curated to represent a diverse range of job categories, ensuring a broad spectrum of vocabulary and terminologies commonly found in resumes. Each resume in the dataset was labeled with its corresponding professional field, providing a ground truth for our classification model. Each resume was preprocessed to remove personal identifiers and structured to highlight key information such as skills, education, work experience, etc.

2.2 Preprocessing

The preprocessing stage was crucial to prepare the resume data for analysis and modeling. We implemented the following steps:

i. **Text Cleaning:** We removed any irrelevant information, such as special characters, punctuation, and numbers, that do not contribute to the semantic understanding of the resume content. This step helped in reducing noise and improving the quality of the text data.

ii. **Tokenization:** The cleaned text was then tokenized, converting the continuous text into a list of words or tokens. This process is essential for further processing the text into manageable units.

iii. **Stopword Removal:** Common words that appear frequently in the text but do not carry significant meaning, known as stopwords, were removed. This included words like "the," "is," "in," etc. Eliminating these words helped in focusing on the more meaningful words that are relevant to the professional context of the resumes.

iv. **Lemmatization:** We applied lemmatization to reduce the words to their base or root form. This step is important for normalizing the text data and reducing the complexity of the vocabulary.

v. **Vectorization:** Finally, the preprocessed text was converted into numerical representations using various vectorization methods such as TF-IDF, Word2Vec, and GloVe. This transformation is critical for enabling the machine learning algorithms to process and analyze the text data.

Through these preprocessing steps, we ensured that the resume dataset was cleaned, normalized, and transformed into a suitable format for training our classification model. This meticulous preparation of the data was fundamental to the success of our automated resume screening system.

2.3 Model Architecture

Our automated resume screening system employs a classification model that is designed to categorize resumes into their respective professional fields. The architecture of our model is structured to effectively handle the numerical embeddings generated from the preprocessed resume text. We explored various machine learning algorithms to find the most suitable one for our task. The key components of our model architecture are as follows:

i. **Embedding Layer**

The first layer of our model is the embedding layer, which receives the numerical representations of the resumes. Depending on the vectorization method used (TF-IDF, Word2Vec, or GloVe), this layer is responsible for mapping the high-dimensional sparse vectors to a lower-dimensional dense vector space. This transformation is crucial for capturing the semantic relationships between words and reducing the computational complexity of the model.

ii. **Feature Extraction**

Following the embedding layer, we implement feature extraction techniques to identify the most relevant features from the resume embeddings. This step is essential for distilling the most informative aspects of the resumes that indicate the professional field. We experimented with different feature selection methods to optimize the performance of our model.

iii. **Classification Algorithms**

For the classification task, we explored various algorithms to determine the most effective approach for our dataset. Our investigation included:

- One-vs-Rest Classifier: A multi-class strategy that fits one classifier per class and predicts the class with the highest probability.
- Multilayer Perceptron (MLP): A type of neural network that consists of multiple layers of neurons, each connected to the next with weighted edges. MLPs are capable of learning complex non-linear relationships.
- Support Vector Machine (SVM): A powerful algorithm that finds the optimal hyperplane that separates different classes in the feature space.

Each of these classifiers was evaluated based on their ability to categorize resumes into the correct professional field accurately.

iv. **Optimization and Evaluation**

To ensure the optimal performance of our model, we employed various optimization techniques such as hyperparameter tuning and cross-validation. The model was trained on a training set and validated on a separate validation set to assess its generalization ability. Key metrics such as accuracy, precision, recall, and F1-score were used to evaluate the model's performance.

Through this carefully designed model architecture, our automated resume screening system aims to provide an efficient and accurate solution for categorizing resumes into their respective professional fields, thereby enhancing the recruitment process.

2.4 Training

In the initial stages, we created resume embeddings using TF-IDF, Word2Vec, and GloVe to compute similarity scores against a job description, helping identify the most effective techniques for capturing relevant features. We then trained One-vs-Rest classifiers, SVMs, and MLPs with these embeddings, using 5-fold corss-validation to ensure a thorough evaluation. Hyperparameters were fine-tuned throughout the process. We also monitored precision, recall, F1-score, and accuracy, using confusion matrices to identify errors. This systematic approach enabled us to refine our models, leading to a robust resume screening system optimized for accurate candidate categorization.

3 Results

Please find the model performance metric tables below in Appendix A.1.

From our extensive analysis of two diverse datasets, we discovered significant variances in model and embedding performances, underpinning the intricate nature of resume screening.

For Dataset #1 [1], the models showed high efficacy on both training and testing data. Here the MLP and SVM models using GloVe embeddings achieve the highest validation accuracies, indicating superior generalization capabilities. These models reached validation accuracies close to 100%, illustrating the effectiveness of GloVe in capturing contextual nuances suitable for advanced model architectures.

Conversely, Dataset #2 [2] presented a more challenging scenario where even the best performing SVM-GloVe combination reached only 70.5% in validation accuracy, highlighting the impact of dataset idiosyncrasies on model performance.

4 Discussion

The disparity in performance across the datasets underscores the significance of tailoring embedding techniques to align with specific model capabilities and the unique characteristics of the dataset.

In Dataset #1 [1], higher accuracies suggest that the labels were well-defined and possibly more consistently applied, thus more amenable to the sophisticated pattern recognition capabilities of more advanced models like MLP and SVM when combined with GloVe embeddings. This environment likely presented a more homogeneous or cleaner dataset.

On the other hand, Dataset #2 [2] displayed lower performance metrics, possibly due to more anomalies or a broader diversity of expressions, which could have strained the ability of the embeddings to represent semantic meanings effectively. The lower performance of Dataset #2 may also hint at less precisely defined or inconsistently defined labels, complicating the training process. These insights highlight the need for not only refining prepossessing methods and exploring more nuanced embedding techniques but also ensuring the quality and consistency of labeling in the data. Addressing these issues could help in distinguishing closely related job roles more effectively, such as the frequent misclassification of ETL Developer resumes labeled as "Web Designing," and enhance overall model performance.

5 Conclusion

This project underscores the utility of NLP tools in automating the resume screening process, offering a scalable solution to enhance the recruitment workflow. Future work could focus on refining the models used, as well as exploring deep learning techniques for improved semantic analysis of resumes and job descriptions alike. Additionally, integrating BERT models could further enhance the ability to capture contextual nuances in the text, potentially leading to more accurate and effective categorization of candidates.

6 Future Work

For future work, we can employ the following:

- Model Optimization: Investigate hyperparameter tuning and ensemble methods to optimize the performance of existing models (SVM, MLP, and one-vs-rest classifiers).
- Advanced Vectorization: Experiment with more advanced word embedding techniques, such as fastText or ELMo, which can capture subword information and more complex syntactic nuances.
- Deep Learning Approaches: Delve into other deep learning architectures like Convolutional Neural Networks (CNNs) for sentence classification or Recurrent Neural Networks (RNNs) for handling sequential data within resumes.
- Contextual Embeddings: Explore the use of state-of-the-art contextual embedding models like BERT, GPT-3, or RoBERTa that offer dynamic word representations, improving the understanding of context within resumes.
- Dataset Expansion and Diversification: Enrich the dataset with more varied resumes and job descriptions to improve the model's generalization capabilities.

7 Acknowledgements

This project was completed for CS 6140 (Natural Language Processing) under Professor Uzair Ahmad during the Spring 2024 semester. We would also like to thank our TA Tarun

Reddy who helped us understand the requirements of the project and helped us throughout to achieve our final goals.

References

[1] Dutta, Gaurav. 2021. "Resume Dataset." Kaggle. Accessed April, 2024. https://www.kaggle.com/datasets/gauravduttakiit/resume-dataset/data.

[2] Palan, Maitri. 2021. "Resumes." Kaggle. Accessed April, 2024. https://www.kaggle.com/datasets/maitrip/resumes.

A Appendix A

A.1 Model-Embedding Pair Performance Tables

Model	Embedding	Training Accuracy	Validation Accuracy
One-vs-Rest	TF-IDF	97.9%	97.5%
One-vs-Rest	Word2Vec	96.9%	94.0%
One-vs-Rest	GloVe	97.4%	94.5%
MLP	TF-IDF	100%	99.7%
MLP	Word2Vec	80.5%	84.1%
MLP	GloVe	98.7%	99.4%
SVM	TF-IDF	99.9%	99.5%
SVM	Word2Vec	95.5%	92.7%
SVM	GloVe	99.7%	98.8%

Table 1: Dataset #1 [1], Comparison of Model-Embedding Pair Performances

Model	Embedding	Training Accuracy	Validation Accuracy
One-vs-Rest	TF-IDF	59.0%	37.4%
One-vs-Rest	Word2Vec	62.6%	45.0%
One-vs-Rest	GloVe	66.9%	54.8%
MLP	TF-IDF	94.0%	64.4%
MLP	Word2Vec	51.0%	49.2%
MLP	GloVe	79.4%	66.0%
SVM	TF-IDF	73.1%	55.2%
SVM	Word2Vec	43.8%	40.7%
SVM	GloVe	77.1%	63.9%

Table 2: Dataset #2 [2], Comparison of Model-Embedding Pair Performances