Class: Natural Language Processing

Date: 12/10/2023

Team 2: Amjad Altuwayjiri

Empowering Job Seekers with Advanced NLP: A Revolutionary Approach to Career Navigation in the Modern Job Market

1.Introduction

This project addresses the gap in current job search tools, such as the ONET® Database shown in Fig.1, which primarily focuses on manual navigation and job title-based searches, often not adequately aligning with a job seeker's unique skills and career level. The advanced Natural Language Processing (NLP) program is designed to overcome these limitations by offering a more dynamic, user-centric approach. It uses modern NLP techniques to streamline job searching, helping individuals find positions that precisely match their skills and career aspirations. The project comprises two main components: a Text Classification Model for accurately predicting job levels in resumes, and a Semantic Similarity Search using Siamese-BERT Networks. By deploying these innovative tools, we aim to transform the job-hunting experience into one that is more efficient, targeted, and aligned with the individual career development goals of job seekers.

2. Personal Contribution to the Project

In this project, my contributions were focused on the development, testing, and optimization of three distinct models: a Transformer-based Model using BERT with a customize liner layer, a Logistic Regression model, and a MLP model to classify job levels.

I was also responsible for developing and fine-tuning the Logistic Regression and MLP models and BERT transformer with linear layer. These models served as comparative benchmarks, allowing us to evaluate the effectiveness of our advanced NLP techniques against traditional classification methods. My role involved not just model training but

also extensive hyperparameter tuning using GridSearch and plotting confusion matrix for the three models and leaning curve for MLP and Logistic regression.

Finally, my contributions extended to the development of the Streamlit application, where I integrated all the models and their functionalities into a cohesive, user-friendly interface. This platform allowed users to interact directly with our models, upload resumes, and receive tailored job recommendations and classifications.

3. Methodology

This section describes the development and functionality of the three primary components of the project that was contributed by the writer:

3.1 BERT transformer with a Linear layer

Job level classification component was designed to categorize job roles into various levels. This task was accomplished using a classifier based on the BERT model, which was further enhanced with a linear layer. This BERT-based classifier effectively leveraged natural language processing (NLP) techniques to interpret and categorize the complexities of different job roles.

A vital part of this methodology involved the BERT tokenizer, which transformed textual job descriptions into a numerically tokenized format. Following this tokenization, the classifier, integrated with the pre-trained BERT model, uses a linear layer for the classification task. This layer performed a mathematical transformation of the input features from the BERT model, aligning them with the predefined job-level classes. The linear transformation was represented by the equation:

$$n = w \times p + b$$

Furthermore, the softmax function was applied to the output of the classifier's linear layer. This function mathematically transformed the output logits into a probability distribution over the job levels. The softmax equations for the output layer were given as

$$a = \text{softmax}(W \times a + b)$$

to ensure that the output probabilities summed up to one and were interpretable as class probabilities.

The training of this model focused on accurately classifying job descriptions by minimizing a loss function, typically Cross-Entropy Loss in multi-class classification tasks. This loss function quantified the difference between the model's predicted probabilities and the actual class labels.

To fine-tune this BERT model with a linear layer, a hyperparameter tuning process using grid search was used. The parameters adjusted included learning rates, batch sizes, and the number of epochs. Specifically, the learning rates considered were (1e-5, 3e-5, 5e-5); batch sizes were set at (8, 16, 32); and the number of epochs options were (3, 4).

Upon completion of the grid search, the best-performing model configuration was identified as having a learning rate of (5e-05_), a batch size of (16), and a duration of (3) epochs. This configuration was selected for its superior performance to be used.

3.2 MLP

The Multilayer Perceptron (MLP) classifier in the project was optimized using Grid Search, a method chosen for its efficiency in finding the best combination of hyperparameters. Key hyperparameters tuned included the number of hidden layers, activation function, solver algorithm, L2 regularization strength, and initial learning rate. These parameters were varied within predefined ranges: hidden layers (50 or 100), activation function (tanh or relu), solver (sgd or adam), L2 regularization strength (0.0001 or 0.05), and learning rate (constant or adaptive).

A Grid Search was done, and the best model obtained from the search is activation: relu, alpha: (0.05), hidden layer sizes: (100,), learning rate: constant, solver: adam which was evaluated on the test set. Its performance was measured in terms of accuracy.

To complement the analysis, visual representations were also created. A learning curve graph displayed the relationship between training size and accuracy, and a confusion matrix was plotted to provide a visual insight into the model’s classification accuracy across different classes. These visuals offered a comprehensive view of the MLP model's performance and its effectiveness in the classification task.

3.3 Logistic Regression

In the project, the Logistic Regression model was employed as a comparative classifier to evaluate against the advanced machine learning models. This model's development began with feature extraction using TF-IDF and setting a maximum feature limit of 5000, to

transform the job descriptions into a TF-IDF matrix, thus preparing the data for the Logistic Regression model. Mathematically, TF-IDF is computed as:

$$\text{tf idf}(t,d,D) = \text{tf}(t,d) \,.\text{idf}(t,D)$$

It is calculated by multiplying two components: Term Frequency (TF), which counts the number of times a word appears in a document, and Inverse Document Frequency (IDF), which gauges how common or rare a word is across all documents.

Logistic Regression, a staple in classification tasks, was selected for its simplicity and effectiveness. The logistic regression model estimates probabilities using a logistic function, which is an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits. The logistic function is defined as:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}}$$

Then it was optimized using Grid Search, a systematic approach to hyperparameter tuning. This process involved adjusting parameters which are the regularization strength values of (0.1, 1, 10) and the maximum number of iterations values of (100 and 200). The best model configuration was regularization strength: (10), maximum number of iterations: (100).

3.4 Streamlit Application

The Streamlit application serves as the interactive front-end of our system, designed to seamlessly integrate the sophisticated backend models with a user-friendly interface. This

application is essential for processing resumes uploaded in PDF format, where it extracts and structures text data for analysis, transforming unstructured resume content into a format amenable to computational processing.

Key to the application's functionality is the employment of a BERT-based classifier for predicting job levels. This classifier analyzes resumes against the backdrop of job descriptions, determining the most suitable job level for each user. The results are presented in a clear, interpretable format, aiding users in understanding their career prospects.

Additionally, the application offers job recommendations based on semantic similarity analysis. It compares the content of the user's resume with pre-encoded job descriptions, suggesting jobs that align with the user's skills and experience. This feature not only enhances job relevance but also personalizes the job-seeking experience.

Overall, the Streamlit application is a vital component of our system, combining advanced analytical capabilities with ease of use. It provides job seekers with an efficient and insightful tool to explore job opportunities and gain clarity on their career paths.

4. Result

In this study, the performance of two neural network models and two machine learning algorithms: BERT with a Linear Layer, MLP, Logistic Regression, and Naive Bayes were assessed. Their effectiveness was evaluated in terms of accuracy with the results detailed in Table 2.

Table 2.

Evaluation Results

Algorithm	Accuracy
BERt and Linear layer	72%
MLP	69%
Logistic Regression	69%

Table 2: Performance Metrics of the Neural Network Algorithms

As shown in Table 2, the BERT algorithm performed the best, achieving 72% accuracy. MLP and Logistic Regression performed close to BERT with 69% accuracy. While they are marginally lower by 3% in accuracy, they show drastically faster processing times and lower computational resource requirements, making them more practical for real-time applications.

According to BERT confusion matrix Figure 6, most of the entries that are predicted correctly in class 1 that is ‘Entry Level’ however we have to note that this class has the highest number of entries across all levels. While class 2 that is ‘Junior‘ got the lowest class to be predicted correctly where 13 entries were predicted as 1 ‘Entry Level‘, 2 were predicted as 4 ‘Mid Level‘and 2 were predicted as 4 ‘Senior‘

Fig 6

BERT Confusion Matrix

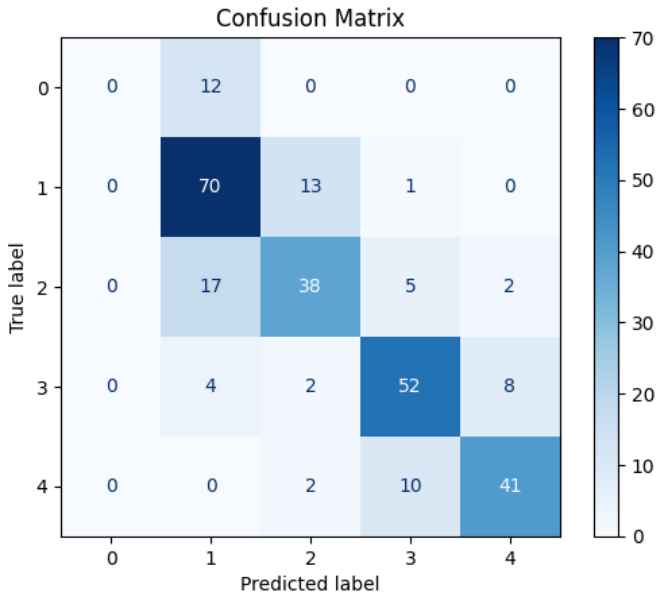

Fig 6: Confusion matrix of true and predicted labels of BERT model

The same as BERt, the MLP and Logistic Regression confusion matrix Figure 7 shows the same result but with more errors in predicting `Entry Level` and `Juinior` that it predicted 16 true `Entry level` as `Juinior` and 14 true `Junior` as `Entry Level` in the MLP model. For the Logistic Regression 19 true `Entry level` were predicted as `Junior` and 16 true `Junior` as `Entry Level`.

Fig 7.

MLP and Logistic Regression Confusion Matrix

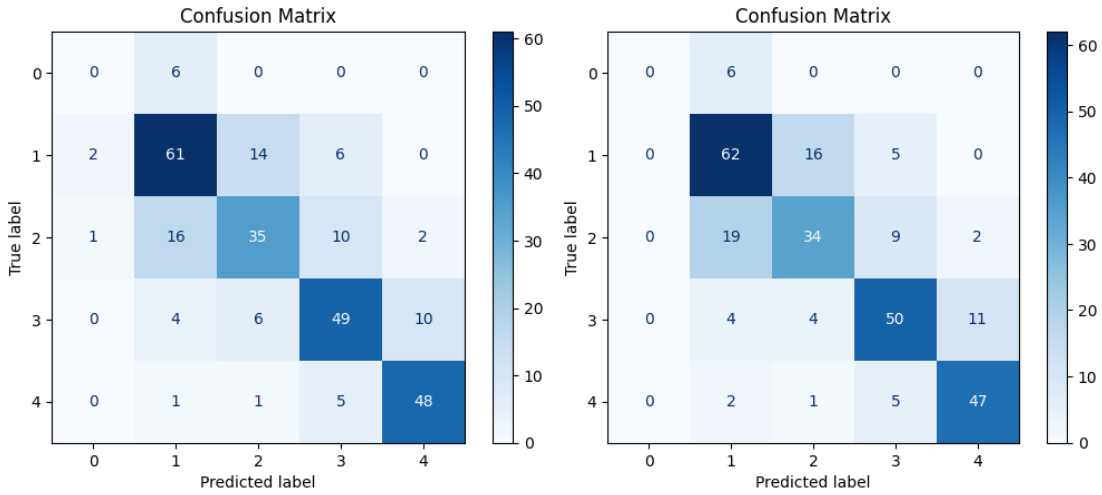

Fig 7: Confusion matrix of true and predicted labels of MLP model on the left and the Logistic Regression on the right

The learning curve for the MLP and Logistic Regression models in Figure 8, both show an improvement in training score as the number of training examples increases, indicating that the models are effectively learning from the data. However, the cross-validation score, although improving, remains consistently below the training score. This gap suggests that while the model is learning the training data well, it's not generalizing quite as effectively to new, unseen data. This implies that further model improvements might require techniques other than adding more training examples, such as feature engineering or model parameter tuning to reduce overfitting and improve their ability to generalize.

Fig 8.

MLP and Logistic Regression Learning Curve

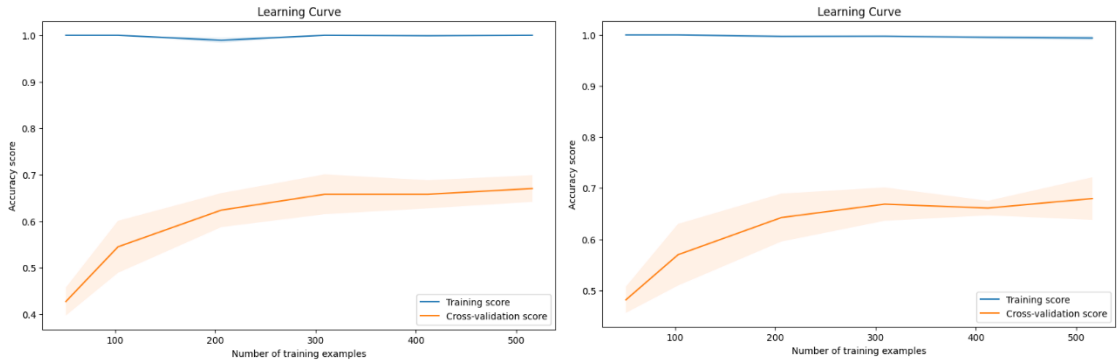

Fig 8: MLP training learning curve on the left and Logistic learning curve on the right

Finally, a Streamlit application serves as the interactive front-end of our system, designed to seamlessly integrate the sophisticated backend models with a user-friendly interface. This application is essential for processing resumes uploaded in PDF format as in Figure 9, where it extracts and structures text data for analysis, transforming unstructured resume content into a format amenable to computational processing.

Key to the application's functionality is the employment of a BERT-based classifier for predicting job levels. This classifier analyzes resumes against the backdrop of job descriptions, determining the most suitable job level for each user. The results are presented in a clear, interpretable format, aiding users in understanding their career prospects.

Additionally, the application offers job recommendations based on semantic similarity analysis. It compares the content of the user's resume with pre-encoded job descriptions, suggesting jobs that align with the user's skills and experience.

Fig 9.

Streamlit architecture

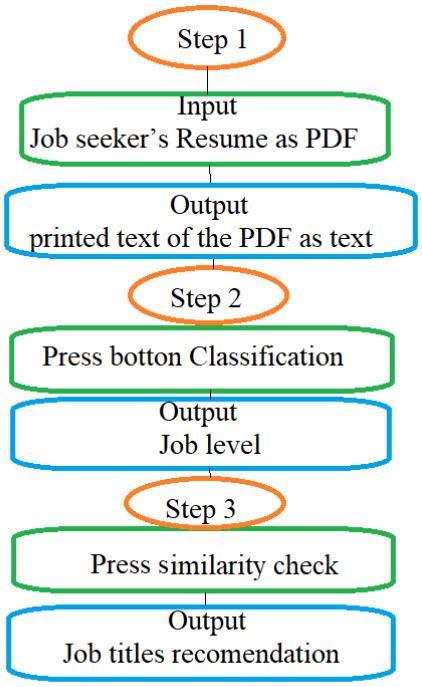

Fig 9. Final Streamlit looks like with three main functions: reading a file and tokanize, classifying job level, and suggesting job titles

This feature not only enhances job relevance but also personalizes the job-seeking experience. By helping them to gain the initiative to search specific information about the occupations that they would be interested in rather than randomly inputting keywords that would not fit them.

5. Conclusion

Natural language processing techniques such as a BERT model with a linear layer, MLP, Logistic Regression, and Naiev Base have been applied to create an advanced job classification system, achieving accuracies ranging from 64% to 72%. These models have been instrumental in accurately categorizing job levels and aligning job seekers with suitable career opportunities based on their resumes. The application of these methods

promises to enhance the job search process significantly, offering a more personalized approach that aligns with the career goals and skill sets of individuals.

A limitation of this project is the reliance on a dataset that may not encompass the full diversity of job roles and descriptions found in the broader job market. The models' performance, could potentially be improved with a more extensive and varied dataset.

5. Calculation

(170-50)/(170+30)*100 = 60

Reference

ChatGPT, personal communication, December 11, 2023

Hagan, M. T., Demuth, H. B., Beale, M. H., & Jesús Orlando De. (2016). *Neural network design.* s. n.