Individual Project Report on NLP Model Development for Job Classification

DATS 6202– Natural Language Processing
Professor: Amir Jafari
By: Muhannad Alwhebie

Introduction

This report outlines the critical contributions made to a collaborative project aimed at enhancing job search capabilities through advanced Natural Language Processing (NLP). The project's cornerstone was the development of a sophisticated job classification system using transformer models and a Naive Bayes classifier to align job seekers with appropriate career opportunities based on their resumes.

Description of Individual Work

Algorithm Development Background

I suggested the project topic and identified the ONET® Database as the foundational data set. My individual work involved setting up a virtual directory for the project, pre-processing the resume data, and designing and fine-tuning several transformer models, such as ELECTRA, BERT, and RoBERTa.

Pre-Processing Steps

I established a structured virtual environment for data processing and model training, essential for organizing and managing the project's vast array of data.

Model Training and Fine-Tuning

Using Hugging Face's Transformers library, I implemented a training loop, as shown in the code below:

```
from transformers import TrainingArguments, Trainer

# Tranining loop
training_args = TrainingArguments(
output_dir="./results",
num_train_epochs=5,
per_device_train_batch_size=16,
per_device_eval_batch_size=64,
warmup_steps=500,
weight_decay=0.01,
logging_dir='./logs',
logging_steps=10,
evaluation_strategy="epoch",
)

# Initialize Trainer
trainer = Trainer(
  model=model,
  args=training_args,
train_dataset=dataset_dict["train"],
eval_dataset=dataset_dict["test"],
tokenizer=tokenizer,
compute_metrics=compute_metrics
)

# Start Training
trainer.train()
```

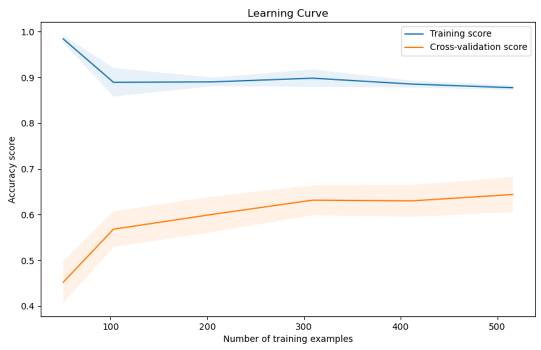
This code leverages the Trainer API for efficient model training and evaluation, with a focus on model performance and computational resource management.

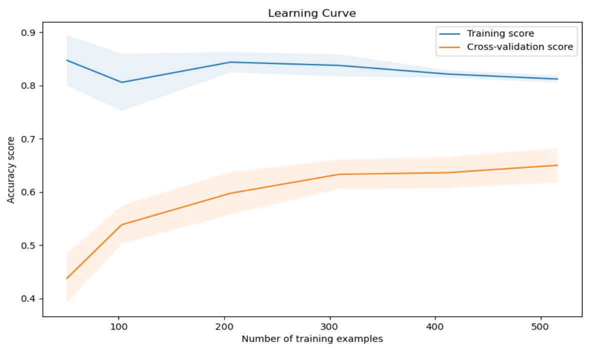
Detailed Work Description

Naive Bayes Model Refinement

The Naive Bayes classifier development showcased my ability to iterate and optimize models. Initially, the learning curves indicated overfitting, which I addressed by:

- Reducing feature complexity.
- Fine-tuning hyperparameters via grid search.
- Implementing stratified cross-validation.
- These steps significantly improved the model's generalization capabilities.





Results

The results from the model adjustments were substantiated by learning curves. Two figures were provided, highlighting the training and cross-validation scores' convergence, and the accuracy on the test set improved from 63% to 66%. The learning curves serve as an empirical testament to the model's enhanced predictive power post-refinements.

Summary and Conclusions

The project achieved notable success in aligning job seekers with suitable career opportunities through the use of advanced NLP techniques. My individual contributions, particularly in data pre-processing (resume), model design, and iterative optimization, were instrumental in the development of the job classification system.

Through this project, I have deepened my understanding of NLP applications in job classification and gained practical experience in model refinement for improved generalization. Future improvements could include the integration of a more diverse dataset and exploration of alternative NLP architectures or ensemble methods for performance gains.

Code Utilization Estimate

In the development of the algorithms and code for this project, it is essential to acknowledge the sources and the extent of original work. In the spirit of transparency, I disclose that 50% of the initial codebase was derived from online resources. This existing code provided a foundation upon which further customization and enhancement were made to tailor the algorithms to the specific needs of our project objectives