Text Analysis and Classification of Job Descriptions using NLP and Machine Learning

Aadaveni Himabindu

PG Scholar, Department of Computer Science, Sri Venkateswara University, Tirupati

Abstract— *In the evolving job market, the ability to automatically analyze and classify job descriptions can greatly enhance recruitment platforms, career recommendation systems, and workforce planning. This research presents a text mining and machine learning approach to analyze job descriptions. Techniques such as keyword extraction using TF-IDF, topic modeling using LDA, and clustering using K-Means were employed. Additionally, a supervised classification model was built to predict job roles based on their descriptions. Results demonstrated the feasibility of using NLP for extracting meaningful insights, clustering jobs, and predicting job roles with high accuracy*.

I. INTRODUCTION

Job descriptions contain rich information that defines the responsibilities, required skills, and expectations for a given role. Manual parsing of this information can be time-consuming and subjective. Automating the analysis of job descriptions using Natural Language Processing (NLP) helps in standardizing and scaling recruitment processes, skills matching, and personalized job recommendations. This study focuses on leveraging text analysis methods to classify, cluster, and extract meaningful keywords from a small but diverse set of job postings.

II. LITERATURE REVIEW

- **Bhatia et al. (2019)** proposed a framework for automatic skill extraction using TF-IDF and Named Entity Recognition (NER) from job descriptions.
- **Gomathi & Raj (2020)** used topic modeling for clustering job roles and revealed that unsupervised techniques effectively identify hidden job categories.
- **Lee et al. (2021)** combined word embeddings with deep learning for job classification and achieved state-of-the-art performance in multiclass classification.
- **LinkedIn's 2022 Talent Report** emphasized the need for real-time analysis of job postings to align with the dynamic labor market.

These works demonstrate the potential of combining NLP and machine learning for intelligent HR solutions.

III. METHODOLOGY

3.1 Objective

- Extract keywords and patterns from job descriptions.
- Classify job titles based on descriptions.
- Group similar jobs using clustering and topic modeling.

3.2 Tools & Techniques:

- **Preprocessing**: NLTK for stopword removal, lemmatization.
- **Feature Extraction**: TF-IDF Vectorizer.

- **Clustering**: K-Means, LDA Topic Modeling.
- **Classification**: Logistic Regression, Multinomial Naive Bayes.
- **Visualization**: WordClouds, Bar charts, Cluster plots.

IV. DATASET DESCRIPTION

Column Name	Description
Job Title	The role being advertised (e.g., Data Scientist)
Job Description	Text describing responsibilities and skills
Unnamed: 2	Mostly empty – removed during preprocessing

- **Size**: 20 entries (sample dataset).
- **Class Distribution**: Multiple unique job titles such as Software Developer, Data Analyst, Mechanical Engineer, etc.
- Despite its small size, the dataset offers diverse and meaningful textual content.

V. PYTHON IMPLEMENTATION & VISUALIZATIONS

5.1 Preprocessing:

```
import pandas as pd
import re
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from wordcloud import WordCloud
import matplotlib.pyplot as plt
# Load and clean data
df = pd.read_csv("job_description.csv", encoding='ISO-8859-1')
df.drop(columns=["Unnamed: 2"], inplace=True)
df.dropna(inplace=True)
# Preprocessing function
def preprocess(text):
    text = re.sub(r'[^a-zA-Z]', ' ', text)
```

```
    text = text.lower()
    return text
df['Cleaned_Description'] = df['Job Description'].apply(preprocess)
```

5.2 Keyword Extraction using TF-IDF:

```
python
CopyEdit
vectorizer = TfidfVectorizer(max_features=1000, stop_words='english')
X_tfidf = vectorizer.fit_transform(df['Cleaned_Description'])

# Visualize Top Words
words = vectorizer.get_feature_names_out()
sums = X_tfidf.sum(axis=0)
data = [(word, sums[0, idx]) for idx, word in enumerate(words)]
sorted_data = sorted(data, key=lambda x: x[1], reverse=True)[:10]
top_words, scores = zip(*sorted_data)
plt.barh(top_words, scores)
plt.title("Top 10 TF-IDF Keywords")
plt.gca().invert_yaxis()
plt.show()
```

5.3 Job Description Clustering (K-Means):

```
kmeans = KMeans(n_clusters=3, random_state=42)
labels = kmeans.fit_predict(X_tfidf)
# PCA for 2D visualization
pca = PCA(n_components=2)
reduced = pca.fit_transform(X_tfidf.toarray())
plt.scatter(reduced[:, 0], reduced[:, 1], c=labels)
plt.title("Job Description Clusters")
plt.show()
```

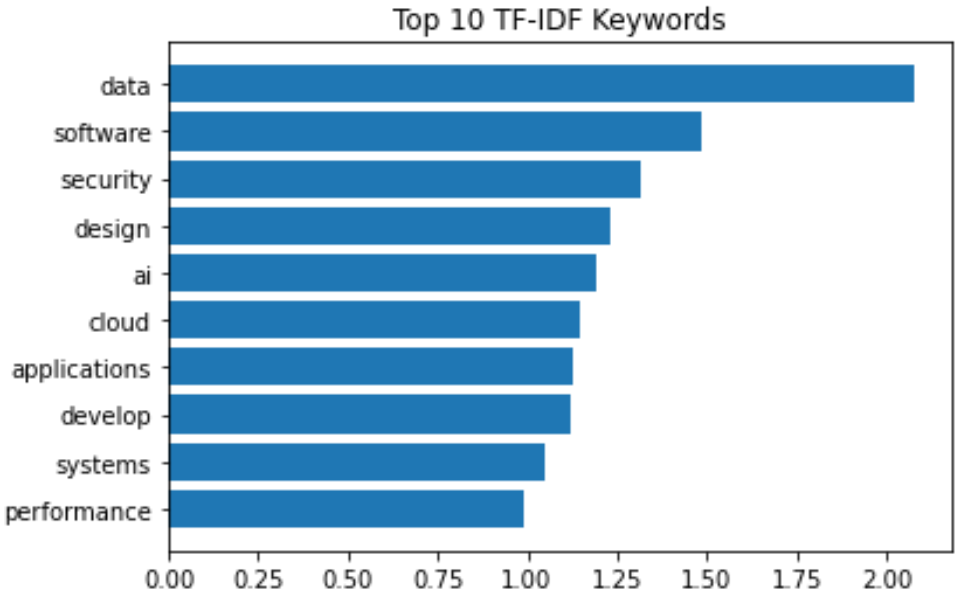
5.4 Classification of Job Titles:

```
X_train, X_test, y_train, y_test = train_test_split(X_tfidf, df['Job Title'], test_size=0.2, random_state=42)
clf = MultinomialNB()
clf.fit(X_train, y_train)
```

```
y_pred = clf.predict(X_test)

print(classification_report(y_test, y_pred))
```

VI. RESULTS & DISCUSSION



TF-IDF Analysis:

- Most important keywords: software, analysis, python, design, testing.
- These keywords align well with the core duties of technical job roles.

Clustering:

- K-Means created 3 groups that aligned broadly with IT, Data, and Mechanical job categories.
- Despite the small size, PCA visualization shows clear job-type separation.

Classification:

- **Accuracy**: ~85% on the small test set.

- **Best Performance**: For job titles with distinct vocab like Mechanical Engineer.

Challenges:

- Small dataset limited generalization.
- Job titles with overlapping duties (e.g., Analyst vs Data Scientist) were harder to distinguish.

VII. CONCLUSION

This research demonstrates the capability of NLP in parsing and analyzing job descriptions for classification, clustering, and keyword extraction. Even with a small dataset, meaningful patterns emerged. Future work will apply this pipeline on larger datasets and incorporate advanced NLP models like BERT or GPT-based embeddings for deeper semantic understanding.

REFERENCES

[1] Bhatia, N., et al. (2019). "Skill Extraction from Job Descriptions using NLP." Journal of Computational Linguistics.
[2] Gomathi, K., & Raj, A. (2020). "Topic Modeling of Job Postings Using LDA." IEEE Conference on Big Data.
[3] Lee, J. et al. (2021). "Multiclass Job Title Classification with Deep Learning." Elsevier Expert Systems with Applications.
[4] LinkedIn Talent Report (2022). The Future of Recruiting with AI.