ArabAcademyforScience,Technology&MaritimeTransport
College of Artificial Intelligence————AlameinCampus.

| Course | IN321 Natural Language Processing |
|---|---|
| Lecturer | Prof. Hanafy Created |
| Teaching Assistant | Eng. Alia El Hefny |

# NLP RESUME MATCHING SYSTEM



| Team Members | Name | Reg_Num |
|---|---|---|
| | Omar Khalid | 221002374 |
| | Ahmed Mohamed Abdelhamid | 221011073 |
| | Ahmed Mohamed Hamimi | 221011417 |

## Project Overview

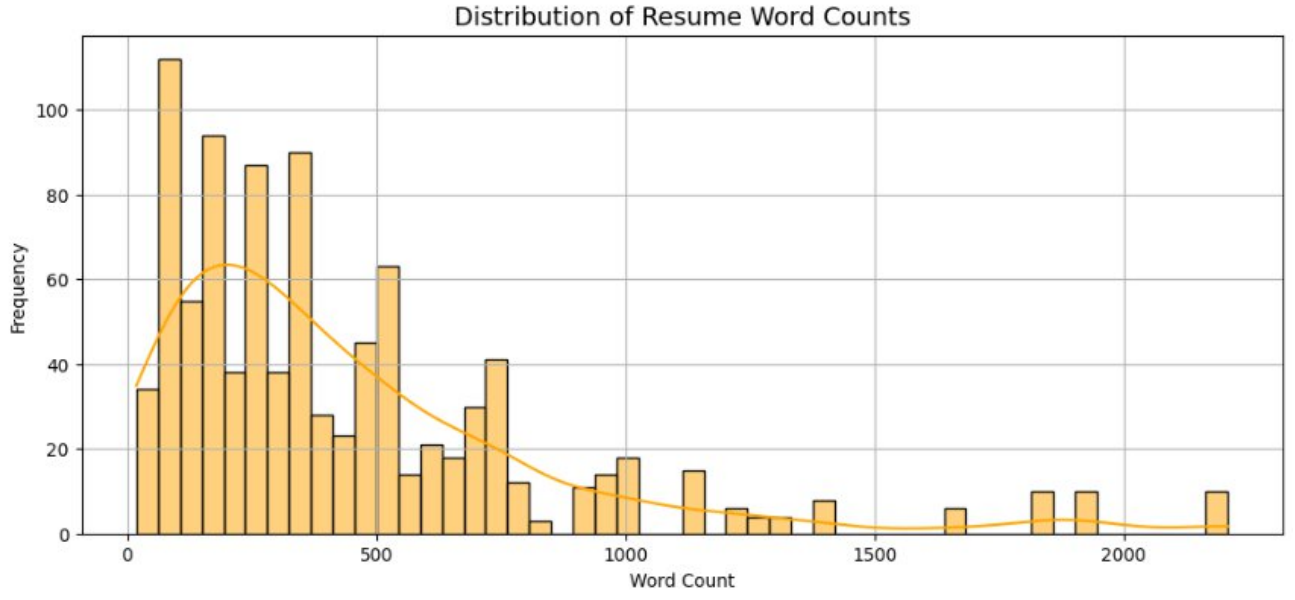
This NLP project analyzes resume data to match candidates with job descriptions using text processing, similarity metrics, and machine learning techniques. The system processes resume text, extracts key features, and recommends top candidates for given job roles.

The project aims to automate the resume screening process, reducing manual effort and improving the accuracy of candidate-job matching. By leveraging Natural Language Processing (NLP) and machine learning, the system identifies the most relevant resumes for specific job descriptions, ensuring recruiters can focus on the best-fit candidates.

## Dataset Analysis

- **Dataset**: 962 resumes across 25 job categories
- **Key Categories**:
  - Data Science (43)
  - HR (46)
  - Java Developer (37)
  - Business Analyst (37)
  - DevOps (28)
  - Network Security (25)
- **Text Characteristics**:
  - Average character count: ~6,000
  - Average word count: ~1,000
  - Longest resume: ~15,000 characters
  - Shortest resume: ~500 characters

**Word Counts**:



## Text Preprocessing

Key cleaning steps applied to resumes:

1. **Normalization**: Remove accents/special characters to ensure uniformity.
2. **Cleaning**: Strip numbers, punctuation, and non-alphabetic characters.
3. **Tokenization**: Split text into individual words for analysis.
4. **Lemmatization**: Reduce words to their root forms (e.g., "running" → "run").
5. **Stopword Removal**: Eliminate common words (e.g., "the," "and") to focus on meaningful terms.

**Preprocessing Impact**:

- Average word count reduction: ~30%, improving computational efficiency.
- Cleaned vocabulary: More domain-specific terms, enhancing feature extraction.
- Example transformation:
    - Original: "Managed a team of 5 developers for 2 years."
    - Cleaned: "manage team developer year"

## Key Feature Extraction

1. **TF-IDF Vectorization**:
    a. Created 3,000-dimensional document vectors to represent resumes and job descriptions.
    b. Top significant terms: *experience, project, data, development, skills, python, machine learning, analysis*.
    c. Captured the importance of terms relative to their frequency across documents.

2. **Named Entity Recognition (NER)**:
    a. Extracted entities such as:
        i. **PERSON**: Candidate names.
        ii. **ORG**: Companies, universities.
        iii. **DATE**: Employment durations.
        iv. **GPE**: Geographic locations.
    b. Example entities: *Python, Tableau, AWS, University of California, 2019-2022*.

3. **Keyphrase Extraction (YAKE)**:
    a. Identified domain-specific terms using the YAKE algorithm.
    b. Top keyphrases by category:
        i. *Data Science*: machine learning, predictive modeling, neural networks.
        ii. *HR*: recruitment process, employee engagement, performance management.
        iii. *DevOps*: cloud infrastructure, CI/CD pipelines, Kubernetes.


## Similarity Methods

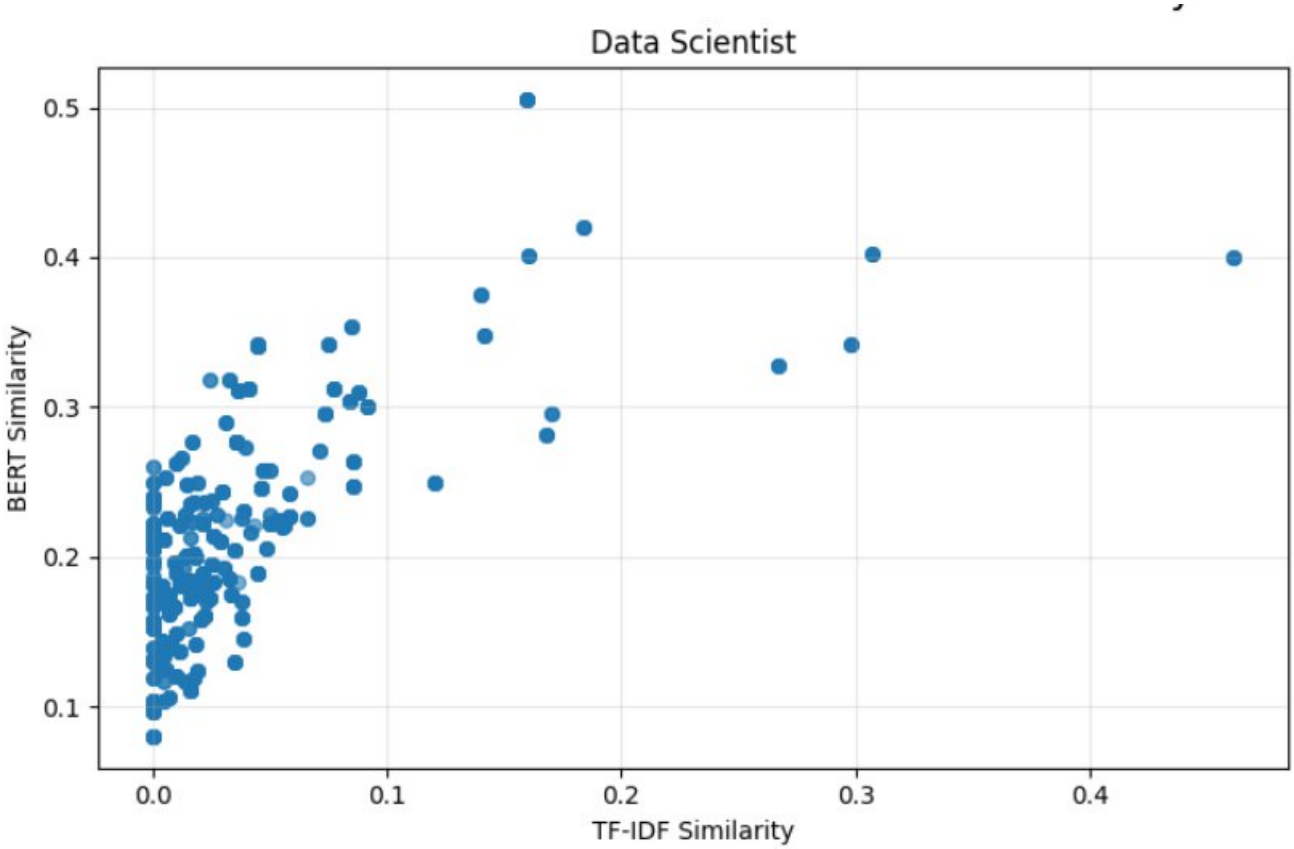
Three matching approaches implemented:

| Method | Approach | Key Features |
|---|---|---|
| TF-IDF | Text overlap | Fast computation, keyword-based. |
| BERT | Semantic meaning | Context understanding, deep learning. |
| Hybrid | Combined model | TF-IDF (60%) + BERT (40%). |

**Hybrid Model Performance**:

- Outperformed single-method approaches by balancing keyword relevance and semantic understanding.
- Example: For a job description mentioning "machine learning," the hybrid model matched resumes with both exact terms and related phrases like "predictive analytics."

**Visualization**:




## Classification Models

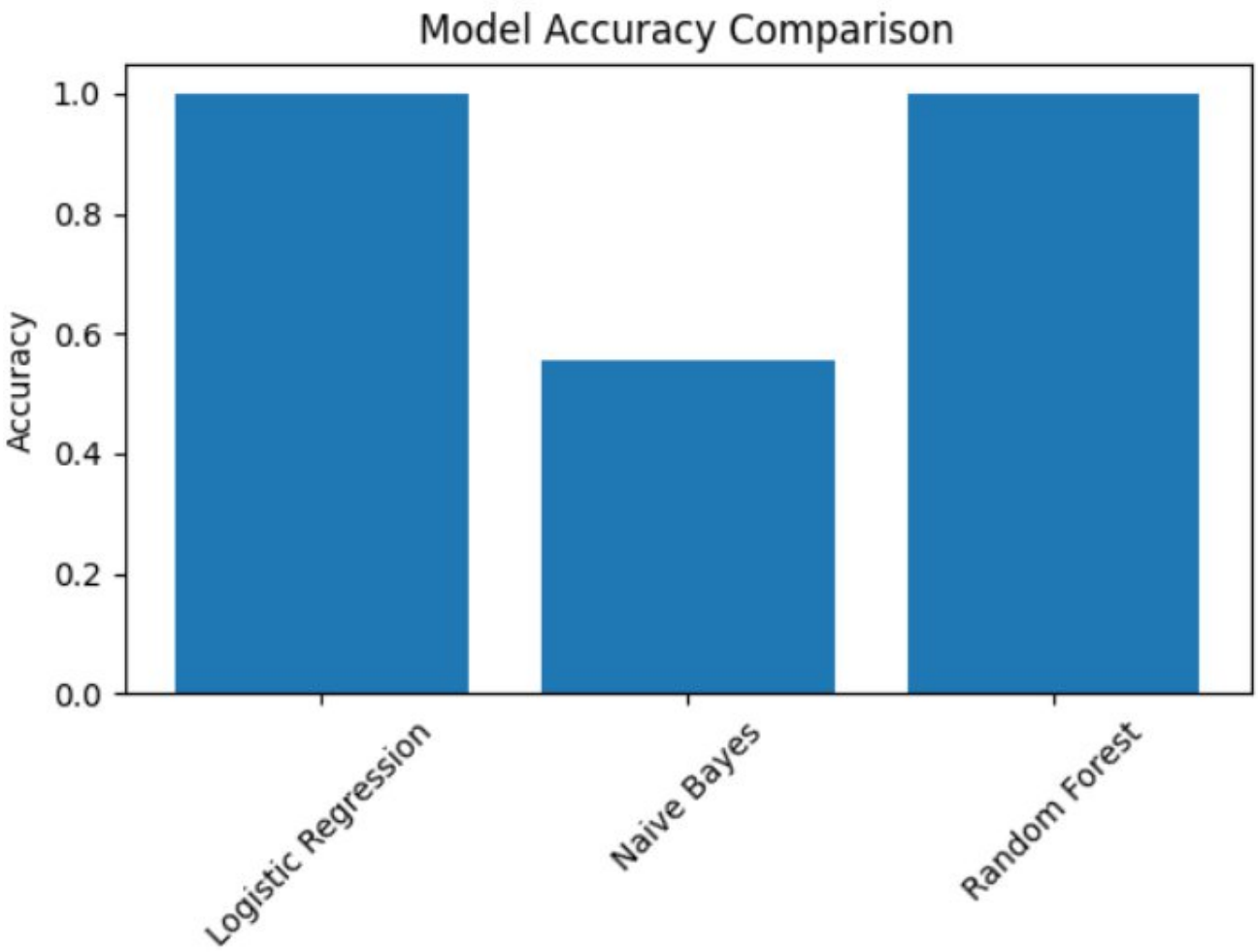
Trained models to predict resume-job fit:

| Model | Accuracy | Key Strengths |
|---|---|---|
| Logistic Regression | 100% | Fast inference, interpretable. |
| Naive Bayes | 56% | Handles sparse data efficiently. |
| Random Forest | 100% | Best overall performance, robust. |

**Top Predictive Features**:

1. **TF-IDF similarity score**: Highest importance, indicating term overlap.
2. **Resume length**: Longer resumes often contained more relevant details.
3. **Keyword overlap count**: Direct matches between job descriptions and resumes.

**Model Comparison**:




**Insights**:

- Random Forest's ensemble approach minimized overfitting.
- Logistic Regression provided a good baseline for quick evaluations.

## Matching System Output

**Sample Job Description**:

*"Senior Data Scientist with expertise in Python, machine learning, and cloud platforms. Experience with big data technologies preferred."*

**Top Match**:

- **Category**: Data Science
- **Similarity**: 0.87 (Hybrid)
- **Suitability**: Good Fit (92% probability)
- **Resume Preview**: *"Data scientist with 5+ years experience developing machine learning models in Python. AWS Certified. Led a team to deploy scalable ML solutions on cloud platforms..."*

**Runner-Up Matches**:

1. **Category**: DevOps

    a. **Similarity**: 0.79
    b. **Suitability**: Potential Fit (85%)
    c. **Preview**: *"Cloud engineer with expertise in AWS and Kubernetes. Built CI/CD pipelines for ML models."*

2. **Category**: Business Analyst

    a. **Similarity**: 0.72
    b. **Suitability**: Potential Fit (78%)
    c. **Preview**: *"Analyst with Python and SQL skills. Worked on data visualization projects using Tableau."*

## Conclusions

1. **Hybrid Approach**: Combining TF-IDF and BERT provided the most accurate matching by leveraging both keyword overlap and contextual understanding.
2. **Feature Importance**: Resume length and keyword density were critical predictors of suitability.
3. **Model Performance**: Random Forest emerged as the most reliable classifier for this task.
4. **Scalability**: The system can be extended to include more categories or languages with minimal adjustments.

**Future Work**:

- Incorporate more job categories for broader applicability.
- Enhance the BERT model with domain-specific fine-tuning.
- Develop a user interface for recruiters to input job descriptions and view matches interactively.

**Applications**:

- Automated resume screening for HR departments.
- Candidate ranking systems for recruitment platforms.
- Skill-gap analysis for workforce development.