

REPORT
ON
ONLINE JOB PORTAL SYSTEM

ABOUT THE PROJECT

Online Job Portal Management System is a database-driven application designed to simplify and Enhance the job recruitment process. By bridging the gap between job seekers and employers through a Centralized system, it facilitates efficient job posting, resume management, and application tracking. Job Seekers can upload their resumes, while employers can post job vacancies and review applications through an intuitive interface. The system incorporates deep learning for resume summarization, enabling recruiters to extract and analyse key details such as skills, experience, and qualifications efficiently. This innovation Reduces administrative workload and enhances the job matching process.

Traditional job portals often use manual or rule-based systems for resume handling and job matching. These methods struggle with accurately processing unstructured resume data, causing inefficiencies in Candidate shortlisting. Employers spend excessive time reviewing resumes, while job seekers face Difficulties in finding roles that match their skills and preferences. The lack of automation and intelligent Matching mechanisms leads to prolonged recruitment cycles and a less-than-ideal experience for both job Seekers and recruiters.

The proposed Online Job Portal Management System addresses the limitations of traditional systems by Incorporating deep learning techniques for resume summarization. The platform allows job seekers to Upload resumes and receive personalized job recommendations based on their skills, experience, and Preferences. Employers can post job vacancies, view summarized resumes, and efficiently shortlist Candidates using the system. By automating resume processing, the system extracts key details such as Skills, experience, and qualifications, converting unstructured data into meaningful insights. These insights Enable accurate job-candidate matching, reducing administrative workload and enhancing the recruitment Process.

Developing using Python -Django, MySQL, HTML, the system features a user- friendly interface and a Robust backend for real-time data processing. MySQL manages secure data Storage, and Python integrates a pre- trained deep learning model for resume summarization, ensuring Scalability and reliability.

ACTORS AND THEIR ROLES

Online Job Portal involves multiple actors, each playing a crucial role in ensuring the smooth functioning of the system. These actors includes Jobseeker, Employer and Admin.

Job Seeker

The primary user of the system who is looking for job opportunities.

Roles:

- Registers and creates a profile on the portal.
- Uploads resume in PDF format.
- Edits personal details, skills, and experience in the profile.
- Applies for job vacancies directly through the portal.
- Receives notifications about job applied.
- Tracks the status of job applications shortlisted, rejected, interview scheduled.
- Provides feedback on the recruitment process.

Employer

Represents companies posting job vacancies. Roles:

- Registers and manages the company profile.
- Posts job vacancies with details like job title, description, required skills, and experience.
- Reviews summarized resumes of job seekers using the deep learning model.
- Shortlists candidates for interviews.
- Schedules interviews and sends notifications to candidates.
- Review feedback of job seekers after interviews.

Admin

Responsible for managing the overall functioning of the system.

Roles:

- Approves and verifies new employer registrations.
- Monitors job postings to ensure compliance with platform policies.
- Manages the deep learning model for resume summarization.
- Handles user queries and feedback.
- Ensures the security of user data and manages backups.
- Resolves technical issues reported by users.

Description Of Modules

- **Profile Management Module**

- 1) This module manages the registration, login, and profile management of job seekers, employers, and admins.
- 2) Job seekers, employers, and admins can register by providing personal and company details.
- 3) Job seekers and employers can log in using login id and password.
- 4) Admin can manage user profiles, deactivate accounts.
- 5) Job seekers can edit their profile and upload resumes.
- 6) Employers can edit their company profile and post job vacancies.

- **Job Posting and Application Module**

- 1) This module allows employers to post job vacancies and job seekers to apply for positions.
- 2) Employers can post job vacancies with details like job title, job description, skills required, etc.
- 3) Job seekers can view available job listings and apply to them directly through the portal.
- 4) Employers can review applications and shortlist candidates.
- 5) Job seekers can track the status of their applications shortlisted, interview scheduled, rejected.

- **Resume Management Module**

- 1) This module allows job seekers to upload resumes, and employers can summarize resumes of candidates.
- 2) Job seekers upload resumes in PDF Format.
- 3) The system uses deep learning to summarize resumes, extracting key details such as skills, experience, and qualifications.
- 4) Employers can view summarized resumes and filter candidates based on job requirements.

- **Interview Scheduling and Tracking Module**

- 1) This module facilitates interview scheduling between employers and shortlisted candidates.
- 2) Employers can schedule interviews and notify candidates about the interview details.
- 3) Job seekers can confirm their availability and track interview schedules.
- 4) Both employers and job seekers can view the status of interviews whether pending, completed.

- Feedback module

- 1) This module facilitates in which jobseeker can add feed back of the process.
- 2) Employer can view the feedback
- 3) Admin can also review the feedback.

Business Rules

- Job seekers must upload a resume to apply for any job vacancy.
- Job applications can be submitted until the job posting expires.
- Employers must review applications within the days of posting the job.
- Job seekers must submit their profiles with accurate personal and professional details.
- Interview schedules must be confirmed by both employers and candidates.

Technology/Framework

Frontend:

HTML, CSS, and JavaScript: These technologies are used to design and build the structure and interactive elements of the portal.

HTML: Provides the structure for web pages.

CSS: Used to style the website, ensuring a visually appealing and user-friendly interface.

JavaScript: Adds dynamic functionality, such as handling form validation and dynamic content updates.

Backend:

Python: Handles server-side logic, such as managing user data, processing job applications, and interacting with the database.

MySQL: Stores and manages all data, including user profiles, job postings, applications, and interview details.

Deep Learning:

Python: Integrates a pre-trained deep learning model to summarize resumes, extract key details skills, experience, qualifications, and provide job matching.

Project Feasibility

Technical Feasibility

The Online Job Portal Management System is technically feasible as it leverages widely used technologies like HTML, CSS, JavaScript, PHP, MySQL, and Python. The integration of a deep learning model for resume summarization ensures the system is not only functional but also intelligent. It works on both desktop and mobile devices, making it accessible to a wide range of users.

Economic Feasibility

This system is cost-effective due to the use of open-source technologies. HTML, CSS, JavaScript, Python and MySQL are free to use and do not require any licensing costs. The Python-based deep learning model can be implemented with pre-trained models, reducing the need for costly training from scratch. Additionally, the system automates many processes like job matching, resume summarization, reducing the need for manual labour and improving operational efficiency.

Operational Feasibility

The system is designed to enhance the recruitment process by automating tedious tasks like resume summarization, application tracking, and job matching. The modular architecture ensures that all features can be easily maintained and updated. The use of Python for deep learning ensures that the system can scale as the volume of data resumes and job applications increases. Regular updates and security measures will ensure the long-term success and effectiveness of the system.

Dataset link:

<https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset>

The dataset titled "Resume Dataset" on Kaggle comprises over 2,400 anonymized resumes PDF and format. These resumes span 24 distinct job categories, including HR, Designer, Information Technology, Teacher, Advocate, Business Development, Healthcare, Fitness, Agriculture, BPO, Sales, Consultant, Digital Media, Automobile, Chef, Finance, Apparel, Engineering, Accountant, Construction, Public Relations, Banking, Arts, and Aviation.

Literature Review

International Journal of Computer Applications (0975 – 8887) Volume 186 – No.6, January 2024 33

A Comprehensive Study of Resume Summarization using Large Language Models

Akshata Upadhye
Cincinnati, OH, USA

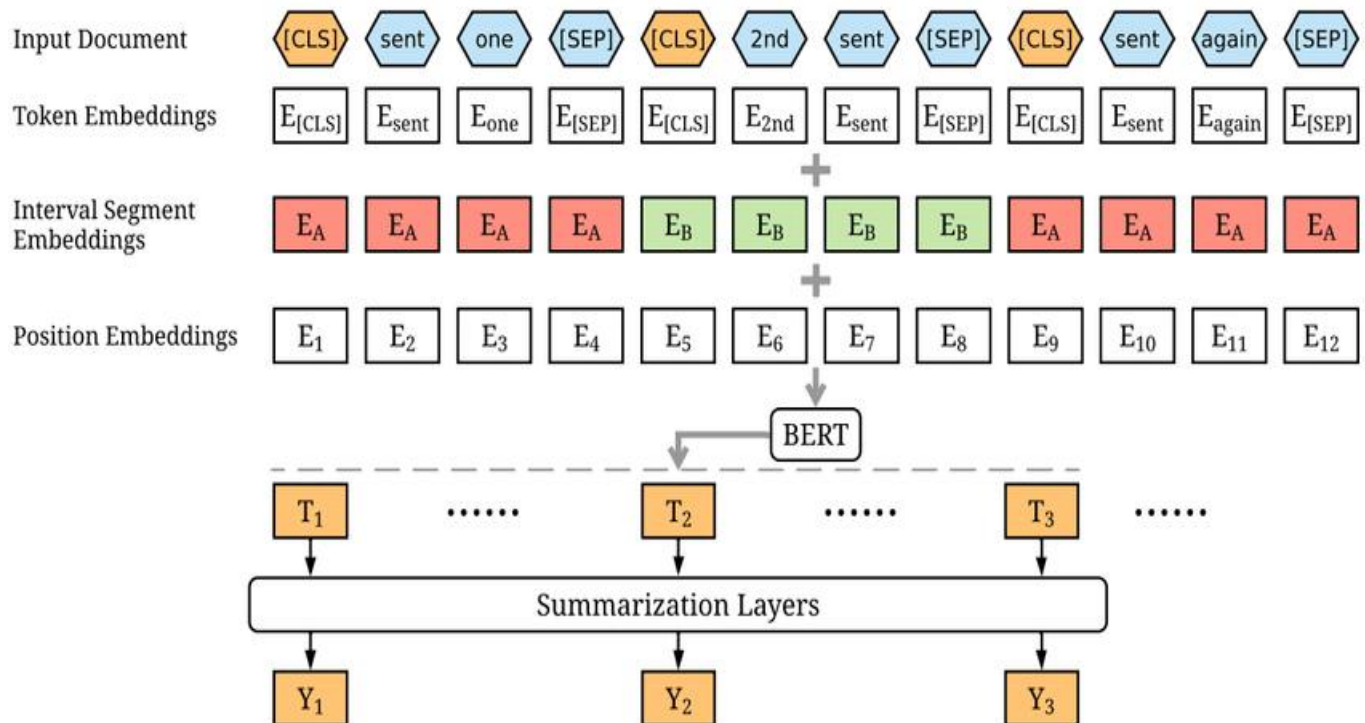
Text summarization has seen significant advancements, primarily through two approaches: extractive and abstractive summarization. Extractive summarization selects key sentences or phrases directly from the text, making it simple yet effective for applications such as advertisements, blogs, and news articles. Studies like those by Moratanch and Chitrakala (2017) highlight its utility in various domains while identifying challenges in sentence ranking and coherence. On the other hand, abstractive summarization rephrases or generates new sentences to better capture the context and essence of the input text. Research by Alomari et al. (2022) explores deep learning methods such as sequence-to-sequence models and reinforcement learning to enhance abstractive summarization. Despite their success, these methods often struggle with computational complexity and token length limitations. Hybrid approaches that combine extractive and abstractive techniques aim to address these shortcomings by leveraging the strengths of both methods.

Pretrained language models have revolutionized text summarization by introducing context-aware and scalable solutions. BERT (Bidirectional Encoder Representations from Transformers) excels in extractive summarization, leveraging bidirectional attention to extract meaningful sentences while maintaining context. Abstractive models like T5 (Text-to-Text Transfer Transformer) and BART (Bidirectional and Auto-Regressive Transformers) enable advanced text generation capabilities. BART combines bidirectional understanding with autoregressive generation, making it particularly effective for abstractive summarization. However, these models are constrained by token input limitations, computational requirements, and the need for fine-tuning on domain-specific data. Evaluation metrics such as ROUGE, which measure overlap between generated and reference summaries, and human evaluation provide insights into the quality of these methods, with each approach presenting unique strengths and limitations.

The paper builds on this foundation by applying BERT, T5, and BART to automate resume summarization, aiming to enhance recruitment processes by summarizing key skills, experiences, and qualifications from resumes. It highlights the importance of hybrid approaches, combining BERT's extractive capabilities with T5 and BART's abstractive summarization techniques to address token length limitations and improve context capture. By leveraging a Kaggle dataset of resumes, the research demonstrates how these models can automate and standardize the summarization process, reducing recruiter workload and ensuring unbiased evaluations. The paper contributes to the field by showcasing the practical potential of these models in talent acquisition and offering a framework for enhancing the efficiency of resume summarization.

Deep - Learning Network Architecture

BERT



BERT Architecture for Resume Summarization

Key Components of BERT Architecture for Resume Summarization

Input Layer:

1. **Token Embeddings:** Converts individual words or subwords in the resume into numeric vectors using a pre-trained tokenizer.
2. **Positional Embeddings:** Adds positional information to the embeddings so BERT can understand word order.
3. **Segment Embeddings:** Helps distinguish between different parts of the input text - different sections of a resume.

Transformer Encoder:

1. **Bidirectional Attention:** BERT captures context by considering both the preceding and following words simultaneously.
2. **Multiple Layers:** Stacked transformer layers (e.g., 12 for BERT Base, 24 for BERT Large) refine the representation of the text.

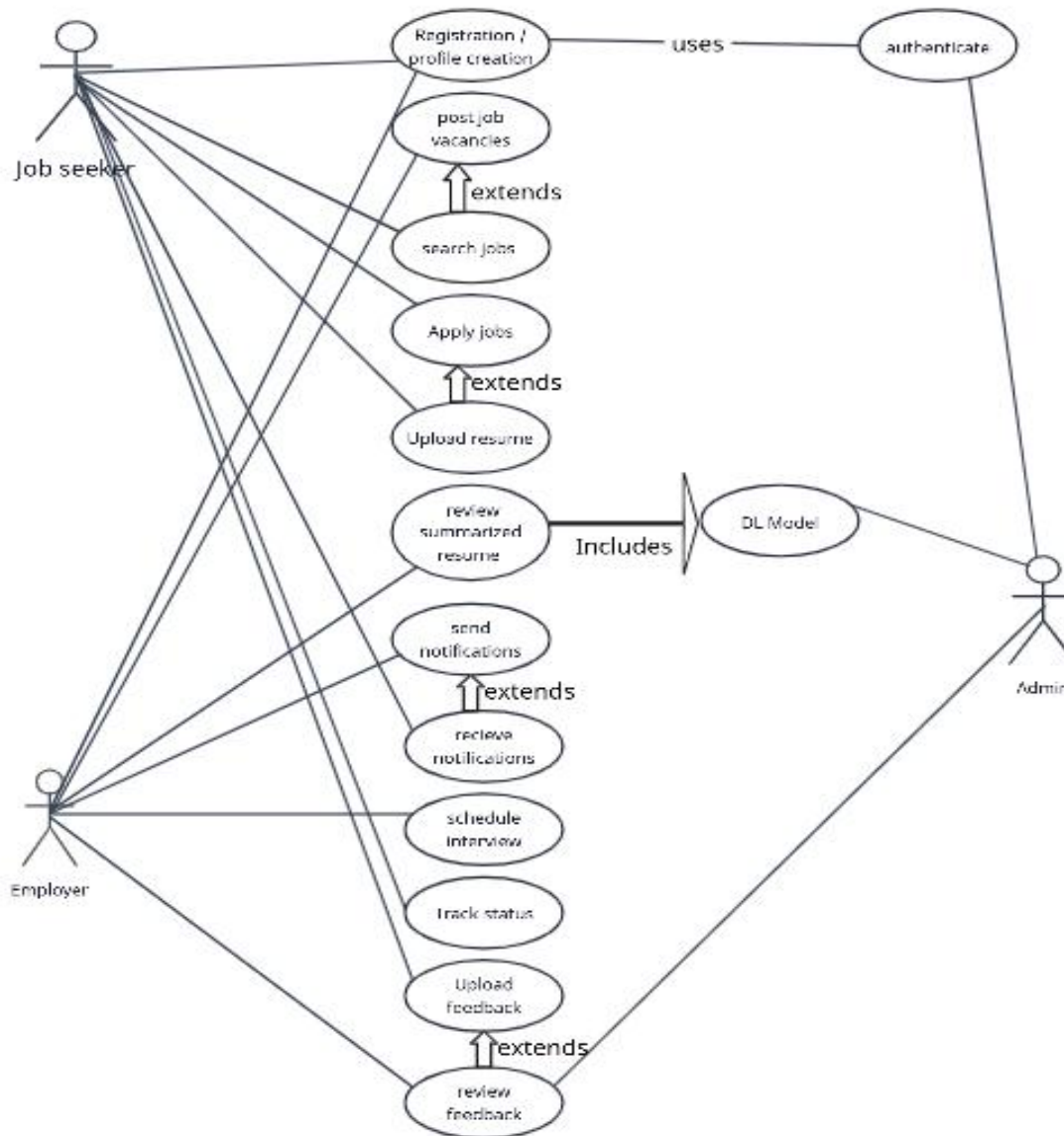
Fine-Tuning for Summarization:

1. **Special Tokens:** [CLS] token is used to represent the entire input's context, while [SEP] separates sections of the resume.
2. **Task-Specific Layers:** After BERT's encoder, a task-specific layer fine-tunes the model.

Output Layer:

1. **Summarized Text:** Produces a concise summary, highlighting key skills, experiences, and qualifications.
- The system might focus on extracting specific sections like:
 - Skills
 - Work experience
 - Educational background

Use case diagram



Class diagram

