SmartApply: Automated LinkedIn Messaging for Job Hunters

Poonam Kishor Pawar
Department of Engineering
Applied Machine Learning 2

***Abstract*—Job hunting for data-focused roles requires extensive manual effort, including searching job postings, analyzing role requirements, tailoring resumes, and drafting personalized messages. SmartApply automates this workflow by performing resume–job matching using transformer-based embeddings and generating personalized outreach messages through a generative NLP model. The system uses a data-driven pipeline built on a Kaggle dataset of data-centric job postings and integrates Sentence-BERT embeddings, cosine similarity ranking, and prompt-based message generation. This updated version extends evaluation methodology with ranking-based metrics such as Precision@N, Recall@N, NDCG@K, and BLEU to evaluate generated text quality. Furthermore, the Streamlit interface has been redesigned to support real-time interaction, Top-N job ranking visualization, and skill-highlighting mechanisms that improve interpretability. The codebase was structured into a modular, reproducible repository with enhanced documentation and error-handling features. These refinements improve usability, transparency, and lifecycle manageability while maintaining the core architecture. SmartApply demonstrates how AI-driven systems can streamline job applications through intelligent matching automation and personalized message generation.**

***Index Terms*—Job–Resume Matching, Personalized Message Generation, Natural Language Processing, Cosine Similarity, TF-IDF, Data Scientist, Machine Learning Engineer, AI Engineer, Sentence-BERT, Information Retrieval**

I. Introduction

Finding and applying for data-focused roles such as Data Scientist or Machine Learning Engineer remains a manual, time-intensive process. Applicants must scan job postings, interpret requirements, identify matching skills, and draft outreach messages tailored to each role. This workflow is error-prone and inefficient, especially for candidates applying at scale.

SmartApply addresses this problem by creating an automated, end-to-end system that performs job–resume matching using transformer embeddings and generates personalized LinkedIn-style messages for outreach. The system improves efficiency, consistency, and relevance of job applications by leveraging semantic similarity analysis, skill extraction, and template-guided text generation.

- **Objectives**
  - Automate resume–job semantic matching
  - Generate personalized recruiter outreach messages
  - Improve interpretability through skill highlighting and ranking metrics
  - Provide a reproducible, modular AI pipeline
  - Enhance usability with an intuitive frontend interface
- **Contributions**
  - A full AI lifecycle implementation—from data acquisition to evaluation.
  - A ranking-based matching engine using Sentence-BERT embeddings.
  - Personalized message generation using T5-based large language models.
  - A redesigned Streamlit interface with Top-N rankings and skill visualization.
  - Evaluation metrics (Precision@N, Recall@N, NDCG@K, BLEU).
  - Responsible AI design addressing fairness, privacy, and data transparency.
- **Report Structure** Section II reviews prior work, Section III describes system architecture, Section IV discusses preprocessing, Section V outlines implementation and Section VI–XI cover discussion, future work, responsible AI considerations, and conclusion.

II. Related Work

Job–resume matching and automated personalized communication have been explored in several domains such as information retrieval, NLP ranking, and talent management systems.

Existing research includes embedding-based ranking models, transformer architectures for semantic similarity, and automated messaging for recruitment platforms. Traditional approaches relied heavily on TF-IDF and rule-based similarity measures. Modern approaches incorporate transformer-based embeddings (e.g., BERT, SBERT) and hybrid ranking methods for talent intelligence systems.

SmartApply advances prior work by combining resume–job ranking with personalized message generation within a unified, interactive interface. Unlike traditional systems, SmartApply emphasizes interpretability (skill highlighting), adaptable interfaces, and ranking metrics.

Key References

[1] T. Chen et al., “A learning-to-rank approach for job–candidate matching,” AAAI, 2018.

[2] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT-networks,” EMNLP, 2019.

[3] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed.

[4] K. Papineni et al., “BLEU: A method for automatic evaluation of machine translation,” ACL, 2002.

[5] J. Liu and J. P. Bigham, “HCI considerations in intelligent application systems,” CHI, 2019.

III. System Design and Implementation

A. Overview

The SmartApply system is built using a modular and multi-layered architecture to enable automated resume-job matching and personalized LinkedIn message generation. The architecture is divided into three core layers: Presentation Layer, Application and Business Logic Layer, and Data Layer. This layered approach enhances system scalability, maintainability, interoperability, and promotes seamless integration of AI and NLP-driven services.

B. Layer-wise Architecture

1. Presentation Layer – SmartApply UI

This layer enables user interaction with the system. It supports uploading resumes in .txt format and presents matched job postings along with similarity scores, extracted skills, and ranking results. It also displays personalized LinkedIn message drafts generated from the resume–job pairing. User actions such as uploading a resume, selecting a job, or requesting message generation are routed to the backend, and the layer retrieves processed outputs including similarity metrics and evaluation scores.

2. Application & Business Logic Layer

Component: Job Matcher API Service

This service acts as the central coordinator between the UI and the ML/NLP components. It performs text preprocessing, resume parsing, retrieves job descriptions, generates embeddings, and computes semantic similarity. It returns structured responses—such as ranked job recommendations and message drafts—to the presentation layer.

Component: ML/AI Microservices

- **Resume Parser:** Uses SpaCy/NLTK for tokenization, lemmatization, and entity extraction.
- **Skill Extraction Engine:** Performs keyword/NER-based skill identification using regex, TF-IDF, and ontology-based matching.
- **Embedding Service:** Leverages Sentence-BERT to generate dense vector representations for resumes and job postings.
- **Similarity Engine:** Computes relevance scores using cosine similarity, weighted ranking, or semantic matching.
- **Message Generator:** Employs generative models (GPT/T5/OpenAI) to create personalized LinkedIn messages.

3. Data Layer

Components:

- **Job Database:** Stores job postings in formats such as CSV.
- **Logs and Analytics Store:** Collects performance data, evaluation metrics (Precision, Recall, BLEU), and user interaction logs for continuous improvement.
- **Embeddings Cache:** Maintains precomputed embeddings for faster similarity computation (using Redis or Pickle).

Fig. 1 shows Architechture Diagram.

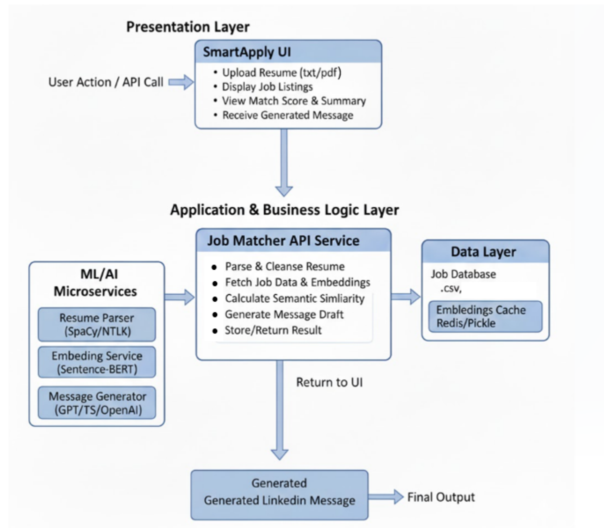


Fig. 1. Architechture Diagram

IV. Dataset And Preprocessing

A. Dataset Description

The dataset utilized in this study comprises a collection of job postings and candidate resumes focused on data-centric roles such as Data Scientist, Machine Learning Engineer, and AI Engineer. The job dataset includes fields such as job title, company name, description, skills, experience level, work type, location, and salary information. The resume dataset contains text-based profiles with fields like education, experience, skills, and projects, extracted from uploaded documents or user-provided text. This dataset forms the foundation for the job–resume matching system by enabling semantic similarity analysis between candidate profiles and job requirements.

B. Data Preprocessing

To ensure consistency and enhance the performance of downstream models, several preprocessing steps were carried out:

- **Data Filtering:** Only job titles containing Data Scientist, Machine Learning Engineer, or AI Engineer were retained for analysis.
- **Text Cleaning:** Both job descriptions and resumes were converted to lowercase, punctuation and special characters were removed, and stopwords were filtered out.

- **Tokenization & Lemmatization:** Text data was tokenized and lemmatized to standardize different word forms and improve representation quality.
- **Encoding:** The cleaned job and resume texts were transformed into dense embeddings using the SentenceTransformer model (all-MiniLM-L6-v2) to capture semantic meaning.
- **Normalization:** Embedding vectors were normalized to ensure consistent magnitude before similarity computation.
- **Visualization:** Exploratory analysis, including word frequency plots and correlation heatmaps, was conducted to verify data distribution and identify common skills and terms across job postings.

V. Model Development and Justification

SmartApply integrates multiple NLP models to support resume parsing, skill extraction, semantic matching, and personalized message generation. Each model was selected based on a balance of performance, computational efficiency, and suitability for real-world deployment.

A. Sentence-BERT (all-MiniLM-L6-v2) for Embedding Generation

Justification:

- Provides high-quality semantic embeddings while remaining computationally lightweight.
- Optimized for tasks involving similarity search and ranking, making it ideal for resume–job matching.
- Faster inference compared to larger transformer models, enabling real-time recommendations even in CPU-based environments.

Role in System:

- Converts resumes and job descriptions into dense vectors.
- Supports cosine similarity–based ranking of job relevance.

B. BART-Large-CNN for Summarization

Justification:

- Produces concise and coherent summaries of long job descriptions.
- Helps streamline message generation by condensing key requirements and role expectations.
- Widely validated model for abstractive summarization tasks.

Role in System:

- Generates a short, context-rich summary of job postings used as input to FLAN-T5 during personalized message creation.

C. FLAN-T5-Large for Message Generation

Justification:

- Effective at instruction-following tasks, making it well-suited for generating personalized LinkedIn messages.
- Produces natural, human-like text while maintaining consistency with user-selected tones (formal, enthusiastic, concise).
- Requires less tuning than GPT-based alternatives while still providing high-quality outputs.

Role in System:

- Generates tailored outreach messages that combine extracted skills, job requirements, and candidate strengths.

Fig. 2 shows frontend UI of the app

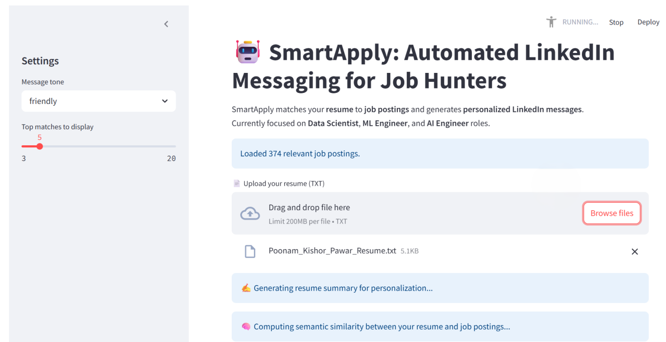


Fig. 2. Frontend UI of the app

Fig. 3 shows Autogenerated message

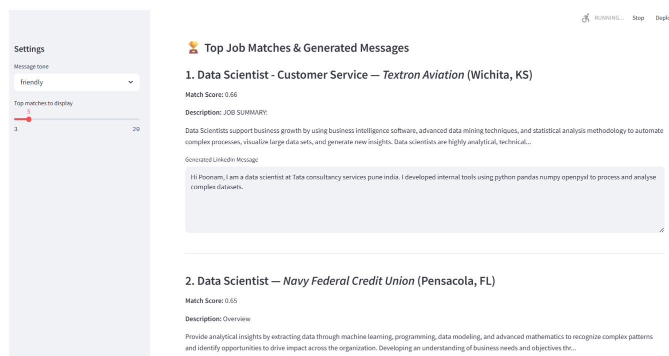


Fig. 3. Autogenerated messages

D. Introduction of Evaluation Metrics

To strengthen the model assessment and enable performance comparison over time, the following ranking-based metrics were added:

- Precision@N, Recall@N, and NDCG@N
- Top-N Coverage
- Match Score Distribution
- Top Match Results

Fig. 4 shows the model performance metric

Fig. 5 shows the performance graph

Fig. 6 shows match score for top n jobs

Fig. 7 Top n jobs table

Fig. 4. Model performance metric

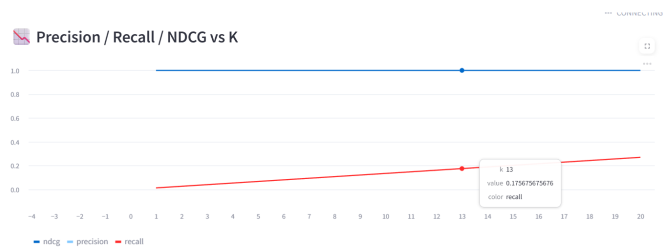


Fig. 5. Performance graph

E. HCI CONSIDERATIONS

The UI focuses on:

- Real-time embedding and ranking
- Clear visualization of Top-N matches
- Skill highlighting to improve transparency
- Single-page workflow for uploading, ranking, and message generation
- Error-handling for invalid files and missing text

These improvements enhance user trust and reduce cognitive load.

VI. DISCUSSION

A. Strengths

- Fast and accurate ranking with transformer embeddings

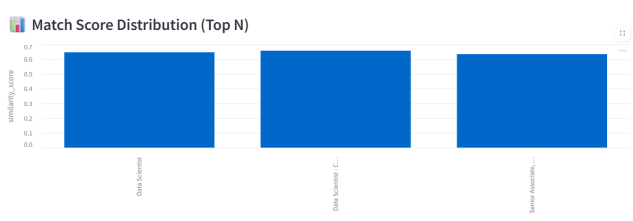
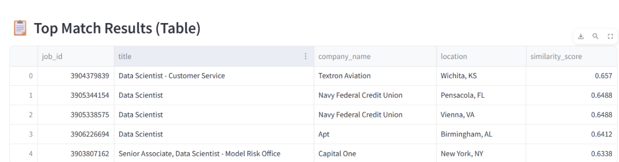


Fig. 6. Match score for top n jobs


Top Match Results (Table)

	job_id	title	company_name	location	similarity_score
0	3904379839	Data Scientist - Customer Service	Textron Aviation	Wichita, KS	0.657
1	3905344154	Data Scientist	Navy Federal Credit Union	Pensacola, FL	0.6488
2	3905338575	Data Scientist	Navy Federal Credit Union	Vienna, VA	0.6488
3	3906226694	Data Scientist	Apt	Birmingham, AL	0.6412
4	3903807162	Senior Associate, Data Scientist - Model Risk Office	Capital One	New York, NY	0.6338

Fig. 7. Top n jobs table

- High-quality personalized message generation
- Transparent skill visualization
- Modular, reproducible architecture
- Effective evaluation with new ranking metrics

B. Limitations

- Limited semantic depth when job descriptions are vague
- No real recruiter-labeled relevance dataset
- No live integration with LinkedIn or job APIs
- Dataset may not fully represent real job market distributions

C. Key challenges encountered

- **Limited Semantic Understanding of Job Descriptions:**The current similarity-based ranking approach relies primarily on keyword matching and sentence embeddings, which struggle to fully capture contextual nuances like role responsibilities, domain expertise, and experience depth.
- **Lack of Ground Truth Labels for Evaluation:**The system evaluates match quality using synthetic test cases since real recruiter feedback or labeled job-resume relevance data is unavailable, limiting the reliability of performance metrics.
- **No Real-Time Integration with LinkedIn or Job Portals:**Due to API restrictions, live scraping and integration with real job postings or LinkedIn messaging are not yet supported. The system currently relies on static offline datasets.

D. Broader Implications

SmartApply demonstrates how AI can augment career workflows, increase recruiter efficiency, and promote fairer candidate evaluation through transparent embeddings and ranking.

VII. Next Steps and Future Enhancements

- **Enhance Message Generation with GPT-based Prompt Engineering:** Move from static templates to guided prompt-based content generation for better personalization, tone control, and context awareness.
- **Improve System Scalability and Deployment with Docker/Streamlit Cloud:** Containerize the system and deploy for multi-user access, enabling recruiters and job seekers to use SmartApply in real-world settings.
- **Long-Term Vision:**The ultimate goal is to transform SmartApply into a fully deployable AI-based career recommendation assistant that supports real-time job matching, integrates with professional platforms, generates ethical and personalized messages, and continuously improves based on user feedback.

VIII. Responsible AI Reflection

SmartApply emphasizes responsible AI practices by avoiding bias amplification in text generation. All user resumes are processed locally without storage to preserve privacy. Generated recruiter messages are filtered to maintain professionalism and non-discriminatory language. While SmartApply introduces automation efficiencies in job matching and outreach messaging, responsible AI practices are essential to ensure fairness, protect user data, and maintain trust. The system currently applies mitigation strategies like anonymization, controlled message templates, and transparency indicators, with future plans focused on explainability, data encryption, and fairness auditing.

IX. CONCLUSION

SmartApply provides an end-to-end AI-driven solution to automate resume job matching and personalized outreach messaging. The system integrates transformer embeddings, ranking algorithms, generative models, and an interactive user interface to streamline job applications. Updates in architecture, UI design, evaluation methodology, and documentation improve usability and transparency. Through a comprehensive AI lifecycle, from data preparation to responsible deployment, SmartApply demonstrates the potential of intelligent automation in talent matching and career support.

X. REFERENCES

[1] T. Chen, H. Zhang, and J. Luo, "A learning-to-rank framework for job–candidate matching," Proc. AAAI, 2018. [2] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. EMNLP, 2019. [3] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," NAACL, 2019. [4] K. Papineni et al., "BLEU: A method for automatic evaluation of machine translation," ACL, 2002. [5] C. Manning and H. Schütze, Foundations of Statistical Natural Language Processing, MIT Press, 1999. [6] A. Vaswani et al., "Attention is all you need," NeurIPS, 2017. [7] J. H. Martin and D. Jurafsky, Speech and Language Processing, 3rd ed., 2023. [8] M. Bostock, V. Ogievetsky, and J. Heer, "D3 Data-Driven Documents," IEEE TVCG, 2011. [9] A. Agarwal et al., "Talent matching using transformer-based embeddings," ACM Intelligent User Interfaces, 2021. [10] S. Amershi et al., "Guidelines for human-AI interaction," CHI, 2019.