

[Open in app](#)

10



Specs for 'Chatbot on Resumes'

Product Requirements Document with hints of Implementation



Yogesh Haribhau Kulkarni (PhD)

Published in Technology Hits

8 min read · Just now



Listen



Share



More

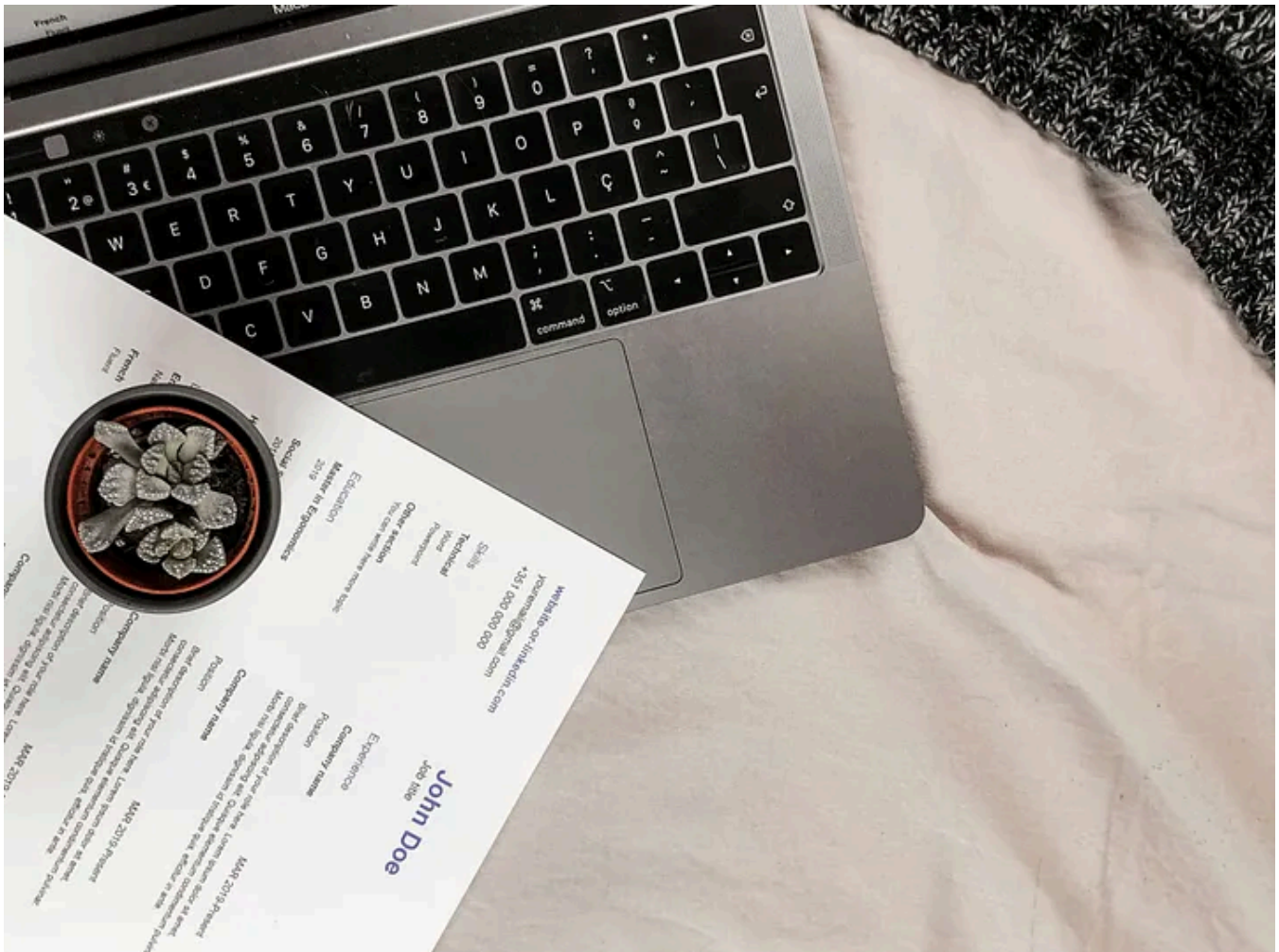


Photo by [João Ferrão](#) on [Unsplash](#)

Introduction

In today's competitive job market, companies often receive a deluge of resumes for open positions, making it a daunting task to review and process each applicant's

information efficiently. Resumes, while providing valuable insights into a candidate's qualifications, skills, and experiences, are typically unstructured documents that require manual parsing and comprehension. This process can be time-consuming and prone to errors, especially when dealing with a large volume of applications. To address this challenge, we propose the development of a "Chatbot on Resumes" system, which aims to automate the extraction of relevant information from resumes, construct a knowledge graph, and enable natural language interactions with the structured data.

Problem Statement

The primary goal of this project is to streamline the resume review process by leveraging natural language processing (NLP) and knowledge graph technologies. The system will tackle the following key challenges:

Information Extraction: Extracting relevant information from resumes in various formats (e.g., docx, pdf, txt) and structuring it in a standardized JSON format. This includes extracting details such as personal information, education, work experience, skills, honors, and other relevant sections.

Knowledge Graph Construction: Designing and implementing a knowledge graph schema that accurately represents the extracted information from resumes. The schema should capture entities like the applicant (SELF), educational institutions, companies, projects, technologies, and the relationships among them.

Natural Language Query Processing: Developing an NLP pipeline to enable users to interact with the knowledge graph using natural language queries. This involves translating natural language queries into SPARQL queries, executing them on the knowledge graph, and presenting the results in a human-readable format.

By addressing these challenges, the "Chatbot on Resumes" system aims to provide a seamless and efficient way for recruiters and hiring managers to navigate through resumes, extract relevant information, and gain insights into candidates' qualifications and experiences through natural language interactions.

Specifications

Input: Job applicants often provide their resumes in various formats, such as Microsoft Word documents (docx), Portable Document Format (PDF), or plain text files (txt). To illustrate the typical structure of a resume, let us consider the

following sample format, which is representative of the content after converting the original file to a textual representation:

MyFirstName MyLastName

Address: 1111, ColonyName, AreaName, Village, City, Country

Mobile: +91 999 999 9999

Email: myfirstname@example.com

Summary:

12+ years in QA automation in various domains. Exploring a Senior Technical rol

Skills:

Certifications: Certified Scrum Master, ISTQB

Programming: Python, Perl, Java

Experience:

Sep 2015 QA Manager, Company1, City1, India.

Till Date - Responsible for managing a QA team: planning, execution of release

- UI automation and REST API automation using Selenium and Cucumber.
- Implemented Continuous Integration using Jenkins and headless browser.
- Python and shell scripting to automate QA tasks.

:

Education:

1999 -00 Diploma in Advanced Computing, Edu1, City1, India, Distinction.

:

Honors:

Multiple recognitions in the form of Spot awards.

:

Output — Extractions: When it comes to the extraction process within our “Chatbot on Resumes” system, we go beyond mere Named Entity Recognition (NER) techniques. Our approach involves capturing not only the entities present in the resume but also the inherent structure, composition, and hierarchical relationships within the content. The output of this extraction phase is a JSON (JavaScript Object Notation) format, which represents the information as a list of nested dictionary objects. This structured representation allows for a more comprehensive understanding and organization of the resume data, enabling efficient storage, retrieval, and querying within our knowledge graph architecture.

```
"FirstName": "MyFirstName",
"LastName": "MyLastName",
"Address": "1111, ColonyName, AreaName, Village, City, Country",
"Mobile": "+91 999 999 9999",
"Email": "myfirstname@example.com",
"Summary": "12+ years in QA automation in various domains. Exploring a Senior T",
"Skills":
{
  "Certifications": ["Certified Scrum Master", "ISTQB"],
  "Programming": ["Python", "Perl", "Java"]
},
"Education": [
{
  "Diploma": "Diploma in Advanced Computing",
  "Institute": "CDAC",
  "Dates": "1999 -00",
  "Domain": "Edu1",
  "Location": "City1 India",
  "Result": "Distinction"
},
{
}
],
"Experience": [
{
  "Company": "Company1",
  "Dates": "1999 -00",
  "Designation": "QA Manager",
  "Location": "City1 India",
  "Projects": ["..", "..."],
  "Technologies": ["Python", "Perl"]
},
{
}
],
"Honors": []
```

In the “Chatbot on Resumes” system, a crucial component is the design and implementation of a robust Knowledge Graph schema. This schema defines the entities that will be represented as nodes, the relationships between them as edges, and the attributes associated with each node and edge.

Entities such as the applicant themselves (referred to as SELF) will be represented as nodes, with attributes like address, email, phone number, and others. Educational institutions attended by the applicant will also be modeled as separate nodes, with

edges labeled “education” connecting the SELF node to these institution nodes. These edges will carry attributes such as date ranges, degrees or certifications obtained, marks or grades, and areas of study.

Similarly, companies where the applicant has worked will be nodes, with edges labeled “experience” linking the SELF node to these company nodes. These experience edges will encompass attributes like start and end dates, job designations, and responsibilities.

Furthermore, the schema allows for modeling projects or products associated with each company, enabling a more comprehensive representation of the applicant’s work history. These projects or products can then be linked to specific technologies or skills, such as Cloud computing, C++, or any other relevant technologies mentioned explicitly in the resume or deduced from the text.

Finally, the schema incorporates edges labeled “skill” that directly connect the SELF node to specific technologies or skills, providing a concise way to represent the applicant’s expertise.

It’s important to note that in our Knowledge Graph schema, we make a clear distinction between node-specific attributes and edge-specific attributes. Node attributes represent properties that are inherent to the entity itself, while edge attributes capture the details of the relationship between two connected nodes.

For instance, consider the education node representing the institution “CDAC” (Centre for Development of Advanced Computing). This node may only have the “location” attribute, as details like the date range, degree or diploma obtained, and academic performance are specific to the individual applicant’s educational journey at that institution.

These applicant-specific attributes are instead captured as properties of the edge connecting the SELF node (representing the applicant) to the education node. As illustrated in the provided schema example, the edge with `edge_id` 1 connecting the SELF node to the “CDAC” education node carries attributes such as “Diploma” (the degree or diploma obtained), “Dates” (the time period of study), “Domain” (the field of study), and “Result” (the academic performance or grade).

This separation of attributes between nodes and edges allows for a more accurate and flexible representation of the resume data. Node attributes describe the entities

themselves, while edge attributes provide the context and details of the relationship between the applicant (SELF) and the other entities, such as educational institutions, work experiences, or skills.

By adhering to this design principle, our Knowledge Graph schema ensures a clear and meaningful organization of information, enabling efficient querying and reasoning within the “Chatbot on Resumes” system.

A possible schema adhering to these guidelines is illustrated below, showcasing the various nodes, edges, and their associated attributes:

```
"Nodes":
[
  {
    "node_id": 1,
    "node_type": SELF,
    "attributes":
    [
      "FirstName": "MyFirstName",
      "LastName": "MyLastName",
      "Address": "1111, ColonyName, AreaName, Village, City, Country",
      "Mobile": "+91 999 999 9999",
      "Email": "myfirstname@example.com",
      "Summary": "12+ years in QA automation in various domains. Exploring a Seni
    ]
  },
  {
    "node_id": 2,
    "node_type": EDUCATION,
    "attributes":
    [
      "Institute": "CDAC",
      "Location": "City1 India",
    ]
  },
  {
    "node_id": 3,
    "node_type": EDUCATION,
    "attributes":
    [
      "Institute": "COEP",
      "Location": "City1 India",
    ]
  },
  {
```

```
"node_id": 4,
"node_type": EXPERIENCE,
"attributes":
[
  "Institute": "Autodesk",
  "Location": "City1 India",
]
},
],
"Edges":
[
  {
    "edge_id": 1,
    "node_id_start": 1,
    "node_id_end": 2,
    "attributes":
    [
      "Diploma": "Diploma in Advanced Computing",
      "Dates": "1999 -00",
      "Domain": "Edu1",
      "Result": "Distinction"
    ]
  },
  {
    "edge_id": 2,
```

This Knowledge Graph schema design allows for a rich and structured representation of the information extracted from resumes, enabling efficient querying, analysis, and natural language interactions within the “Chatbot on Resumes” system.

Output — Chatbot: Once the Knowledge Graph is constructed and populated with the structured data from resumes, the next step in our “Chatbot on Resumes” system is to enable natural language interactions with this wealth of information. This is where the chatbot component comes into play.

The primary goal of the chatbot is to bridge the gap between the user’s plain English queries and the structured Knowledge Graph, allowing for intuitive and seamless access to the resume data. To achieve this, the chatbot employs a two-way translation process.

First, the user's natural language query is transformed into a formal query language, specifically SPARQL (SPARQL Protocol and RDF Query Language), which is designed for querying and manipulating data stored in Knowledge Graphs. This translation process involves analyzing the user's input, identifying the relevant entities, relationships, and constraints, and constructing the corresponding SPARQL query.

Once the SPARQL query is generated, it is executed against the Knowledge Graph, retrieving the relevant information based on the user's query. The retrieved results are then transformed back into natural language responses, allowing the user to receive the requested information in a human-readable format.

To facilitate these translations between natural language and SPARQL, we can leverage two main approaches:

Training Large Language Models: State-of-the-art large language models, such as those developed by OpenAI, Google, or others, can be fine-tuned on datasets specifically designed for natural language to SPARQL translations and vice versa. These models can learn the intricate patterns and mappings required for seamless conversions, enabling accurate and context-aware translations.

Leveraging Existing Frameworks: Alternatively, we can explore existing frameworks and libraries specifically designed for text-to-SQL (or text-to-SPARQL) translations, such as those developed by academic researchers or open-source communities. These frameworks often incorporate rule-based or machine learning-based approaches to parse natural language queries and generate the corresponding structured queries.

By integrating either of these approaches into our chatbot, users can interact with the Knowledge Graph using plain English queries, without the need for specialized knowledge of SPARQL or the underlying data structures. The chatbot acts as an intelligent intermediary, translating the user's intent into formal queries, retrieving the relevant information, and presenting it back in a comprehensible and user-friendly manner.

This natural language interaction capability of the chatbot enhances the accessibility and usability of the "Chatbot on Resumes" system, empowering recruiters and hiring managers to explore and analyze resume data efficiently, ultimately streamlining the recruitment process.

Conclusion

The proposed “Chatbot on Resumes” system leverages cutting-edge technologies in NLP and knowledge graphs to revolutionize the resume review process. By automating the extraction of information from resumes, constructing a knowledge graph, and enabling natural language interactions, the system aims to streamline the hiring process, reduce manual effort, and provide a more efficient and insightful way to evaluate candidates. With its ability to handle large volumes of resumes and facilitate natural language queries, this system has the potential to significantly improve the recruitment process, helping organizations identify the best talent more effectively and efficiently.

Click pic below or visit [LinkedIn](#) to know more about the author



Artificial Intelligence

Natural language processing

Resume

Chatbots

Knowledge Graph

[Edit profile](#)


Written by Yogesh Haribhau Kulkarni (PhD)

1.5K Followers · Editor for Technology Hits

PhD in Geometric Modeling | Google Developer Expert (Machine Learning) | Top Writer 3x (Medium) | More at <https://www.linkedin.com/in/yogeshkulkarni/>

Recommended from Medium



 Erin Anne Lynch

What I wish I'd known about unemployment

Have you ever watched a friend spend months unemployed?

8 min read · Apr 22, 2024



9.5K



259





Sufyan Maan, M.Eng in ILLUMINATION

What Happens When You Start Reading Every Day

Think before you speak. Read before you think.— Fran Lebowitz

🌟 · 6 min read · Mar 12, 2024



15.9K



312



Lists



AI Regulation

6 stories · 448 saves



Natural Language Processing

1452 stories · 959 saves



ChatGPT

21 stories · 630 saves



Generative AI Recommended Reading

52 stories · 1041 saves



 Hazel Paradise

How I Create Passive Income With No Money


many ways to start a passive income today

5 min read · Mar 27, 2024

 12.3K

 295



 mo husseini

50 Completely True Things

This is a repost of a list of posts I made to Threads last fall.

5 min read · May 3, 2024



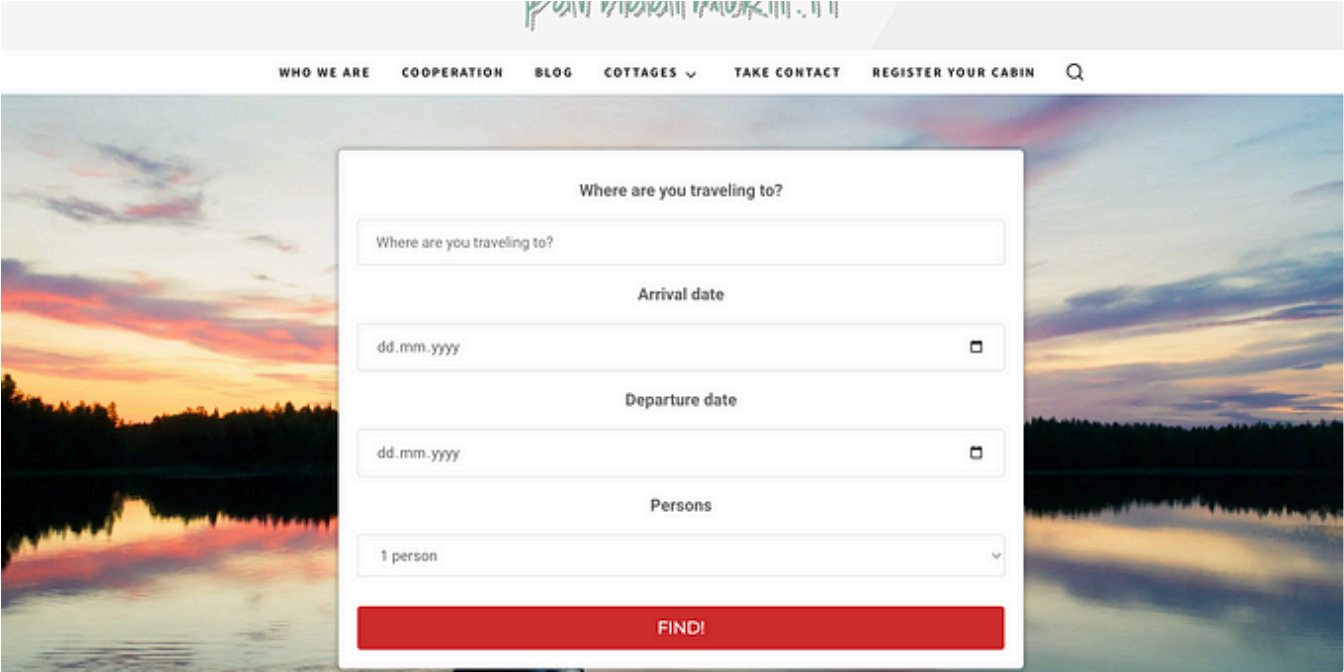
nasyassss

The art of letting people be

In this world that moves with time, I often encounter a delicate thing called “belonging”. Things that are not quite easy to accept when it...

2 min read · May 4, 2024





 Artturi Jalli

I Built an App in 6 Hours that Makes \$1,500/Mo

Copy my strategy!

🌟 · 3 min read · Jan 23, 2024

 18.1K

 194





See more recommendations