Career Copilot Technical Design Proposal

1. Introduction

In this document, I have outlined the technical approach for building Career Copilot, an AI-driven career guidance tool. This project aims to provide users with highly personalized career guidance by matching user resumes with relevant job postings and offering insights based on labor market information (LMI). The solution leverages contextual embeddings using SentenceTransformers and retrieval-augmented generation (RAG) systems with Pinecone.

2. Embeddings Generation

2.1 Data Extraction (1st part in Fig 1)

Based on the data assumptions outlined in the assessment, I am using job postings with structured details like job title, description, required skills and submission links. For the purpose of assessment, I have extracted job postings from LinkedIn with web scraping using **BeautifulSoup.** With which I extracted 278 job descriptions, later on these job descriptions (corpus) are used as non-parametric memory for LLM. Please find the Data Extraction code in Github.

2.2 Model Choice and Preprocessing (1st part in Fig 1)

As I have got all the data from the beautiful soup I have mostly cleaned the data while extracting but there is some basic hygiene preprocessing (using NLTK lib) that needs to be done before embedding the data.

Sentence transformers map text to a 384 dimensional dense vector space, they outperform most of the embedding settings in the market[paper] .We could also use the Open AI embeddings which is trained on much bigger dataset(text-embedding-ada-002) as I am building all my open source we use the former. This is an efficient model to crunch the embeddings to 384 and has good identification in semantic similarity tasks as mentioned in the paper. These embeddings were strong enough to give a very good response to the for our final output produced.

There are also options where the job description could be broken into smaller chunks sizes with partial overlap and then embed.As of now in the code the Job descriptions are converted to embeddings directly. The choice depends upon the Quality of output that we receive from the model evaluating between models.

If we have already extracted standardized info from Job descriptions which could make our final model even more accurate than that I have created with all the Text into the Vector DB.

LLM model: llama 7b chat, the model might be small but as I have to bring all the params in the colab's GPU i have to quantize the model to be able to even generate some text from the data we have. A GPT3.5 , GPT 4 or Llama 70B could give a way better result with AWS cloud in service we could host the weights in them. Please find the **Embedding Generation LLAMA2** code.

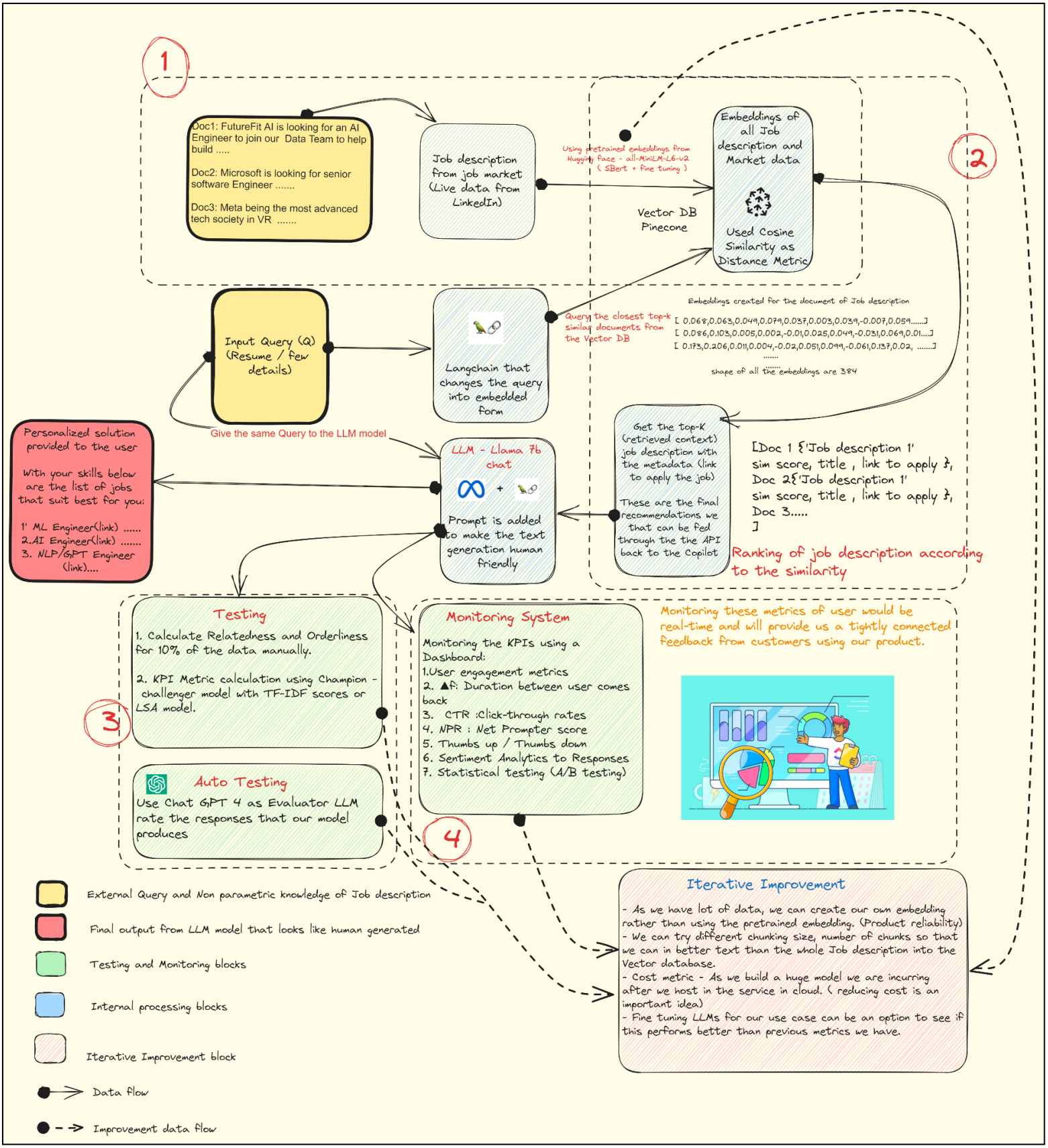


Fig 1. Technical Architecture

3. Matching & Recommendation Mechanism

3.1 Semantic Similarity (2nd Part in Fig 1)

After embedding all the text, I pushed all the data to Pinecone (free api available) Vector Database. I choose cosine similarity for the Pinecone index similarity metric. This is because the thumb rule to choose a distance metric among cosine, dot product, euclidean, is to match it to the embedding model that was trained. We know that all-MiniLM-L6-v2 is trained using cosine similarity as mentioned in the paper above. We could get the top-k documents from Vector DB for the resume that is provided with the score to see the top most and best related data from Job description.

In the code I have put my resume as the input query to check the outputs and as most of my experience is in NLP and LLMs all the Job descriptions generated are very closely aligned with my previous experience. This generated a very good result if even a smaller text is given without a resume.

3.2 Recommendation System (2nd Part in Fig 1)

The data flows as shown in Fig1, after getting the top k recommendations from the Vector Database we feed them to the LLM models with a prompt. The Prompt could help us make the result more human friendly.

```
Prompt: "You are a Job recommender agent, if these are the jobs and their links post a friendly message to the user"
```

I have made sure to have the link of the jobs in the metadata of vector database as shown,

```
"[(Document(page_content=Job Description that is been extracted from the linkedin job posting metadata={source: "link of the job ,title: 'Machine Learning Engineer'}),
  0.617726),
(Document(page_content=Job Description that is been extracted from the linkedin job posting metadata={source: "link of the job ,title: 'Machine Learning Engineer'}),
  0.5796),....."
```

When feeding this into an LLM and retrieving data with the query we will be able to get a very accurate response that is having human touch to the data, hence making the copilot look very promising.

4. Testing & Validation (3rd Part in Fig 1)

As we dont have one size fits all texting metrics we create our own metric for generation tasks.

1. Offline Evaluation:

Validation Methodology is split few major ways:

- Manual Testing or Auto Testing -> does not need user involvement directly
- KPI metrics using models

Offline Metrics:

Assessing the quality of recommendations without involving users directly. These methods measure the quality of our model ranking and generation. Assume that for the input 'Q' to

our system, it generates the Order in the first column for a set of descriptions (Light Red). If we provide the same list of outputs to Human to order them blind folded we end up with the second column (Light Magenta). If we again provide the same set of output to Humans to categorically let them choose Y/N for the input (Q) (Light green)

This table is used to calculate the following score :
Relatedness = ⅘ = .8 (There is one 'N hence 5-1=4)
Orderliness = ⅗ = .6 (2 mismatch as marked by bold text, in case none of them match evaluators are encouraged to make them 0 to that entry)

Fig 2: Validation Example

Input Resume to the system asking for top k-jobs that I could apply? **Let the input vector be 'Q'**	**System generated Jobs similarity Order**	**Human generated job similarity Order (Order the outputs in the best way you feel?)**	**Human generated Relatedness (Is the Output related to the Q?)**
	1	3	Y
	2	2	Y
	3	1	Y
	4	4	N
	5	5	Y

This metric is time consuming to come up with but done for at least 10% of the total responses that we get from the model would help us understand how well our embeddings are able to retrieve the information from the corpus.

2. Auto testing:

- Getting human data is very time consuming and expensive. We can generate all the measures we using GPT4
    - This will provide a quick and faster convergence than human testing but it might be costly to do it on all the text we have.

3. KPI metrics using models:

- The above generated metric is from a gold dataset as its human vetted process. Here we assess our data with the text that is produced as output.
    - We can try to generate the order through TF IDF and check if the score follows the same order as we get from our system generated order. This metric might not be great as TFIDF might not be able to capture the contextual meaning well.
    - LSA is a model that could still be applied to check the order
- Most of the models will not agree 100% with the order produced above as it is not contextual meaning we can compare them for 60 % match which means our model is still a good metric.

5. Continuous Monitoring & Iterative Improvement

5.1 Monitoring System (4th Part in Fig 1)

To monitor performance and KPIs, we set up a system to track:

All these metrics must be hosted on a BI platform to monitor our product performance from time to time.

- User engagement metrics: Calculating total amount of time spent in our platform for the question they had received answers.
- Frequency between them coming back to our platform to explore the recommendation we give for their queries.
- Quick Surveys: Ask only 2 basic questions so that the user answers them with a pop-up before they start using the interface. (survey about the recommendation quality and usage difficulties, hence understanding future improvements to the product)
- Click-through rates (CTR): The links that we provide to the users if they have clicked it through them it would mean that we have provided job descriptions with high similarity which they feel comfortable looking into.
- Net Promoter score: We can get the total number of people who like our product(Promoters) and total number of people who are not liking our productI(detractors). Finding the difference between them would give a Net Promoter score.
- Thumbs up / Thumbs down : % of People who liked our recommendation vs who didn't is a clear indication in our KPI metric
- Sentiment analytics : As we have to improve the product we can ask critical questions to understand the emotion to make our product better in the respective area.
- Statistical testing (A/B testing) : To understand if change of new embeddings work we have to continuously monitor response of control and variation groups.

5.2 Refinement Methodology (Iterative improvement block in the Fig1)

To regularly update and refine the embeddings and recommendation system, we propose:

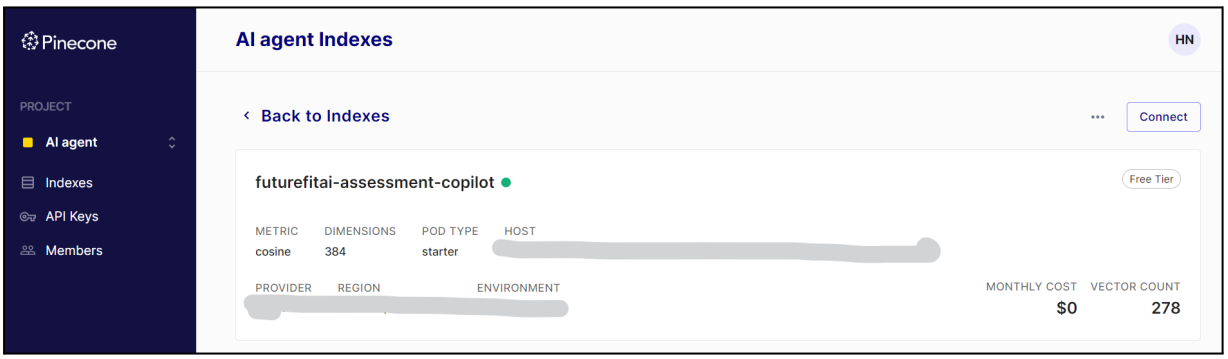
- Transfer Learning: As of now in the code we have used a pre-trained model (BERT/Open AI) embedding which is very generic, we could fine tune them for our downstream task as we have enough data with us.(As newer technologies come up in the JD and job market changes these embedding will fall short in ranking the right roles for the users)
- Trying different chunk size, no of chunks, hyper parameter, memory response to see if the model performs better than before.
- Model Re-training: As of now in the code we are using similarity metric based on embeddings, we can consider a fine tuning model the LLM for our task which will be very precise for our use cases.

6. Conclusion

Career Copilot is containerized using Docker. Then deployed on a Serverless API (AWS lambda) using ECR with Docker as runtime. Using container images with version tags will give us some leverage on model versioning.

Appendix: Code Samples and screen shots

Pinecone vectors :



Input and Output:

```
myresume='''
Hariharavarshan Nandakumar
+1 236 995 3198| hariharavarshan944@gmail.com | LinkedIn |Huggingface

PROFESSIONAL SUMMARY
A Competent management professional and AI expert with 7+ years of cross functional experience that includes Machine
learning, deep learning, predictive modelling, Natural Language Processing and consulting in AI projects and development.
Possess excellent interpersonal, communication and organizational skills with proven abilities in team management,
customer relationship management and planning.
EDUCATION
University of British Columbia, Vancouver, Canada September 2022 – June 2023
Master of Data Science – Computational Linguistics
Projects: Fine Tuning Google LLM(FlanT5), LightGBM classification, House price prediction hosted in FlaskAPI
Anna University, India June 2012 – May 2016
Bachelors of Engineering – Electrical Engineering
TECHNICAL SKILLS
Languages & Tools: Python, SQL, R, PySpark, Pandas, Numpy, Matplotlib, Spacy, Statsmodels, Hugging face, Scikit-learn,
TensorFlow, Pytorch, Keras, Altair, MongoDB, Postgres, Docker, Github, Power BI, Linux GCP, BigQuery, VertexAI, Jira.
Techniques: Machine learning, Deep learning, A/B testing, NLP, T5, Flan T5, BERT, NLU, stochastic gradient descent, LSTM,
Large Language Models (LLM), CNN, NER, Sentiment analysis, Transformers (Attention), clustering, Trees and Graphs.
WORK EXPERIENCE – 7 years
NLP Research Intern – Seasalt.ai, Vancouver Apr 2023 – June 2023
• Researched and Finetuned LLM models such as LLAMA, Alpaca, ChatGPT API to generate Chinese data set for
training Nemo Punctuation toolkit to add Punctuation from ASR raw data with the help of Langchain package. Use
of Hugging face libraries to build RAG and Vector database to add new information to LLMs.
• Used GCP (BigQuery, VertexAI) to design, train and evaluate models on GPUs. Hosted the final fine-tuned models on
GCP for productionizing.
Data Scientist Consultant – HTC Global Mar 2022 – Nov 2022
• To read the Ingredients and Nutrition values from food items package, developed a Transformer architecture with
finetuning BERT and OCR techniques. Deployed models using IAAS such as AWS Lambda, EC2 instance.
```

```
data_vector=vectorstore.similarity_search_with_score(myresume,k=5)
```

```
data_vector
```

```
[(Document(page_content="Responsibilities: Â·Develop and implement machine learning algorithms and models to solve complex business problemsÂ·Collaborate with data
scientists, software engineers, and business stakeholders to identify opportunities to apply machine learning to improve business outcomesÂ·Design, develop, and implement
end-to-end machine learning systems, including data pre-processing, feature engineering, model training and evaluation, and deploymentÂ·Create and own cloud native API t
deploy ML ModelsÂ·Participate in code and design reviews, and contribute to the development of best practices and standards for the machine learning teamÂ·Automate
campaigns in a scalable manner to optimize compute and infrastructure cost Qualifications: Â·Bachelor's or Master's degree in Computer Science, Machine Learning, Data
Science, or a related fieldÂ·2+ years of experience in machine learning, data science, or a related fieldÂ·Strong programming skills in Python and experience with popula
machine learning libraries such as TensorFlow, PyTorch, or scikit-learnÂ·Experience with data pre-processing, feature engineering, model selection, and
evaluationÂ·Experience with cloud platforms and their AI/ML offeringsÂ·Strong understanding of statistical concepts and their applications in machine learningÂ·Excellent
written and verbal communication skills, with the ability to effectively communicate technical concepts to non-technical stakeholdersÂ·Financial industry experience is a
plus", metadata={'source': 'https://in.linkedin.com/jobs/view/machine-learning-engineer-at-indusind-bank-3482703270?
refId=QctKYCkaT8efnDeojCrgIQ%3D%3D&trackingId=kc09jK312qvj%2Ffc9amimxA%3D%3D&position=18&pageNum=0&trk=public_jobs_jserp-result_search-card', 'title': 'Machine Learning
Engineer'}),
  0.617726),
 (Document(page_content='About DyeusWebsiteWe are a digital tech, crypto & e-commerce startup studio working with multiple brands and projects on developing web apps,
mobile apps, and blockchain finance apps.\', \'Our products are used by thousands of happy users daily.\', "Our team has 7+ years of experience and has members from IIT,
who work with us to see how apps are built for a worldwide audience of millions.Activity on InternshalaHiring since June 202224 opportunities posted1 candidate hiredAbou
The Work From Home Job/internshipWe are looking for a skilled and highly motivated college-going intern, currently pursuing a degree in Computer Science, Data Science,
Mathematics or related field, who has previous experience working on GPT-3 using Python.Selected Intern\'s Day-to-day Responsibilities Include Assist the development tea
in building effective AI solutions based on GPT-3 using Python and in deploying cutting-edge deep learning models across products, end to end Design and implement
effective testing programs to ensure efficiency, usability, and accessibilityWhat We Are Looking For Strong foundation in Machine Learning (ML), Deep Learning, and NLP
Familiar with Large Language Models like GPT-3, etc.", "Should have a strong grasp of CS fundamental concepts and ML languages like Python, etc Experience with scientifi
libraries in Python and machine learning tools and frameworks Growth mindset, challenging the status quo to find new solutions and out-of-the-box ideas Ability to work
independently and take initiativeWhy We Are Awesome A 100% remote role Agile working environment with flexible working hours Constant guidance and mentorshipThis will be
a valuable experience for any student looking to learn more about the development of cutting-edge technologies in the field of NLP and AI.Skill(s) requiredMachine
Learning Natural Language Processing (NLP) PythonEarn certifications in these skillsLearn PythonLearn Voice App DevelopmentLearn Machine LearningWho can applyOnly Those
Candidates Can Apply Who are available for the work from home job/internship can start the work from home job/internship between 27th Feb\'23 and 3rd Apr\'23 are
available for duration of 2 months have relevant skills and interestsPerksCertificate Letter of recommendation Flexible work hoursNumber of openings1Additional
QuestionsSign up to continueSign up/ Login with GoogleOREmailPasswordFirst NameLast NameBy signing up, you agree to our Terms and Conditions .Already registered?",
\'LoginSign up to continueSign up/ Login with GoogleOREmailPasswordFirst NameLast NameBy signing up, you agree to our Terms and Conditions .Already registered?\',
```