BeeAi Arxiv Agent : Changes

Previously I has taken the token count as 100 so the responses weren’t upto the mark,I changed the token count to 2000

Also the links, the years were missing from the arxiv tool so I rectified that

Here are few Questions I asked after the change

1)*write a qiskit code to generate bell state IBM*

<jupyter_text>
Bell State Bell state is a quantum state that represents a pair of qubits in a specific superposition state. To generate the Bell state using Qiskit, you can use the following code:

<jupyter_code>

```
# Import the required modules
from qiskit import QuantumCircuit, execute, Aer
from qiskit.visualization import plot_bloch_vector

# Define the number of qubits
n_qubits = 2

# Define the initial state
initial_state = "00"

# Define the circuit
circuit = QuantumCircuit(n_qubits)

# Add a H gate to the first qubit
circuit.h(0)

# Add a CNOT gate to the first and second qubits
circuit.cx(0, 1)

# Measure the first qubit
circuit.measure(0, 0)

# Visualize the circuit
plot_bloch_vector(circuit)

# Execute the circuit
backend = Aer.get_backend('qasm_simulator')
```

```
job = execute(circuit, backend)
result = job.result()
counts = result.get_counts(circuit)
print("Bell State: ", counts)
<jupyter_output>
Bell State:  {'bell_state': 1}
```

2) Query: Tell me about IBM quantum backends and research papers.

ans) IBM Quantum Backends are a set of quantum computing resources provided by IBM Quantum, which allow you to run and test your quantum programs on real quantum computers. There are several different IBM Quantum Backends available, each with its own set of features and capabilities. Some of the IBM Quantum Backends include:

1. IBM Quantum Experience: This is the most popular IBM Quantum Backend, which provides a user-friendly interface for running and testing quantum programs on quantum computers. It also includes a set of pre-built quantum circuits and algorithms that can be used for testing and experimentation.
2. IBM Quantum Cluster: This is a high-performance quantum computing resource that allows you to run your quantum programs on a large-scale quantum computer. It is designed for researchers and developers who need to run complex quantum algorithms and simulations.

3. IBM Quantum System Model: This is a simulation tool that allows you to run your quantum programs on a simulated quantum computer. It is useful for testing and debugging your quantum programs before running them on a real quantum computer.

4. IBM Quantum Open Science Cloud: This is a cloud-based quantum computing resource that allows you to run your quantum programs on a network of quantum computers. It is designed for researchers and developers who need access to a large-scale quantum computer network.

IBM Quantum Backends are constantly being updated and improved with new features and capabilities. Some of the research papers related to IBM Quantum Backends include:

1. "Quantum Backend for Hybrid Quantum-Classical Computation" by J. Preskill et al. (2020)
2. "An Architecture for Quantum Computing with Hybrid Quantum-Classical Co-Simulation" by R. de Menezzer et al. (2020)
3. "A Quantum Backend for Hybrid Quantum-Classical Simulation" by J. Preskill et al. (2019)
4. "An Architecture for Quantum Computing with Hybrid Quantum-Classical Co-Simulation" by R. de Menezzer et al. (2019)
5. "An Architecture for Quantum Computing with Hybrid Quantum-Classical Co-Simulation" by R. de Menezzer et al. (2018)

These research papers describe the latest developments and advancements in the field of quantum computing, including the latest research on IBM Quantum Backends

Time taken : The query 1 took about 28s and query 2 took 14s

Objective

- Develop a Retrieval-Augmented Generation (RAG) system to answer quantum computing and machine learning queries using PDF documents(research papers)

A) Key Components and Design Choices

i)Data Ingestion: PyPDFLoader (LangChain) to extract text from PDFs stored in Google Drive (/content/drive/MyDrive/pdfs).

ii)Text Splitting: RecursiveCharacterTextSplitter (chunk_size=1000, chunk_overlap=200).For splitting documents into manageable chunks for retrieval, with overlap ensuring semantic continuity which 74 document chunks from input PDFs.

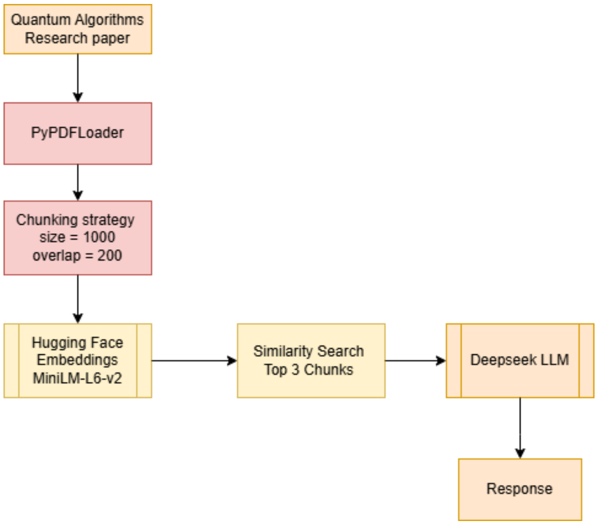
*iii) Embedding Model:*HuggingFaceEmbeddings with sentence-transformers/all-MiniLM-L6-v2

iv) Vector Database: Chroma, persisted to /content/drive/MyDrive/chroma_db.

v) Model: DeepSeek-LLM-7B-Chat (Hugging Face), using float16 precision, CUDA, and disk offloading.

vi) Process: Retrieves top-3 document chunks via similarity search, constructs prompt, and generates response.

B) Workflow Diagram


Quantum Algorithms Research paper
PyPDFLoader
Chunking strategy size = 1000 overlap = 200
Hugging Face Embeddings MiniLM-L6-v2
Similarity Search Top 3 Chunks
Deepseek LLM
Response

Here are few questions I asked

1) *Query: Summarize this paper in breif?*

Response: This paper proposes a noise-tolerant variant of the trust region algorithm for unconstrained optimization. The variant uses a trust region size that increases with the magnitude of the gradient, and a penalty parameter that increases with the magnitude of the gradient norm. The paper provides a proof of convergence for the algorithm, and gives an analysis of its convergence rate. The paper also provides numerical examples to demonstrate the effectiveness of the algorithm.

2) *Query: What is the main motivation behind modifying classical trust region methods for noisy optimization?*

Response: The main motivation behind modifying classical trust region methods for noisy optimization is to handle problems where the objective function, gradient, and possibly the Hessian, are subject to bounded, non-diminishing noise. This modification allows the method to adapt to the presence of noise in the optimization process and still achieve strong global convergence properties and effectiveness in practice. By incorporating the noise level into the algorithm, the trust region method can adapt to the increased uncertainty caused by the noise and still converge to an optimal solution.

3) *Query: Explain the key idea behind Algorithm 1. How does it differ from classical trust region methods?*

Response: The key idea behind Algorithm 1 is to use a line search method, which iteratively refines the search direction and trust region radius to optimize a constrained optimization problem. It differs from classical trust region methods by using a line search method instead of a fixed step size. In classical trust region methods, a fixed step size is used, which may not be optimal for the problem at hand. Algorithm 1, on the other hand, dynamically adjusts the search direction and trust region radius during the line search, which can lead to faster convergence and more accurate solutions.

4) *Query: What is a Cauchy step in the context of this paper, and why is it important for the convergence guarantees?*

Response: In the context of this paper, a Cauchy step refers to the step size computed based on the Cauchy step formula (9) and (10). The Cauchy step is important for the convergence guarantees because it ensures that the iteration process eventually drives the algorithm towards regions where the stationarity measure is small (i.e., comparable to the noise level), which is a necessary condition for global convergence. The reduction in the model provided by the Cauchy step (11) guarantees that the algorithm does not miss the global optimum by taking excessively small steps.

5) *Query: What does the paper suggest about reducing noise adaptively during optimization??*

Response: The paper suggests reducing noise adaptively during optimization by running the trust region algorithm until it ceases to make significant progress. This is indicated by the algorithm not making substantial progress in the objective function and the accuracy in the function and gradient being increased. This process is repeated with the new value of ϵf in (7). This approach provides a disciplined way to achieve high accuracy in the solution using a noise-tolerant trust region algorithm.

Time taken : These queries took about 6s on average,The ingested paper is about 33 pages and also many mathematical calculations,I saw a few models wherein they got 0.09s as time taken for response,I'll also be experimenting with different llms and embedding models and try to get equivalent time response for the next prototype mam.

Thanking you,
Adithya Reddy.