PROJ -00141 References Guide

Contents

Project Management Details

Research Statement and Objectives:

- **Quantum Computing Project Code:** PROJ-00141
- **Statement and Objective:** https://quantumopenai.com/quantum-nlp-proj-00141/
- **Mentor:** Amrit Chhetri, Quantum Security Researcher| Quantum Cognition Researcher | Editor of IQCRJ

Project Management and Steps and Requirements (00141):

All Scholars of participants of this Project can follow these steps to start with:

- Prepare ***'Software Design Specs'***. in docx for the given Problem Statement and Objectives of the project . It should be minimum of 50 pages, including Hardware Software Specification (qLab), QNLP Libraries and Frameworks), deployment details, Design details, etc.
- In Approaches section of Specs. Add applicable Quantum ways/algorithms for text classification (***QSVM, QGAN***) and Hybrid Approaches (**QVE**) and suggest the best through by study/researches and best practices. Also, add existing Classical NLP Techniques and plan for Benchmarking, with details benchmarking Frameworks.
- Make your MLOps/DevSecOps Settings- QLab, GitHub, etc are ready and update them in the specs document.
- Now, you and your team are ready to code in WFM and to complete Beta/Draft version with 40% Buffer, ie. to have the first release within 28 days
- Iterate as needed for further tuning, fixing and adding new things after that.
- Make your Project Readme attractive and complete with all necessary details
- Also, create an account with IBM Q, Classiq to design, create and develop Quantum Circuits of your new approaches https://quantum-computing.ibm.com/,

Software and Libraries Details:

In this Project, you all need to create different software and here are some of them in tabulated manner:

#	Type/Category	Software	URL
1	IDE	Jupyter Notebook (QLab)	http://api.quantumopenai.com:9595, https://hypy.loophole.site https://white-bobcat-16.telebit.io/
2	QNLP Libraries	Lambeq	https://github.com/CQCL/lambeq
3		TorchQuantum	https://github.com/mit-han-lab/torchquantum
4	Code Management	GitHub or GitLab	https://github.com/, https://about.gitlab.com/
5	Benchmaking Framework	QC-App-Oriented-Benchmarks	https://github.com/SRI-International/QC-App-Oriented-Benchmarks

Project Management and Activities:

Create Project Management Plan and include following, add as your need. You can use any Software Management Tool preparatory or open software

1. Project Activities Breakups Sheet
2. Current Status activities should have been completed
3. Software Environment Configuration - QLab, QNLP Libraries, IBM Q Access or Classiq or HyPy Cloud
4. Plan for Advanced Quantum Circuits for new QNLP Models
5. Planning and Strategies of Performance Tuning and Comparison Matrix- Classical and Quantum
6. Execution of Models/Algorithms/Circuits(new if any) in IBM Hardware
7. Platform/Software Mentioned
    1. IBM Quantum: https://www.ibm.com/quantum
    2. Classiq Quantum IDE: https://www.classiq.io/
    3. HyPy Cloud: https://hypy.quantumopenai.com/
    4. IBM Free Credits : https://quantum-computing.ibm.com/credits-program
    5. Quantum Hardware: https://www.quantinuum.com/

Reference Codes and Account Details

Technological Readiness and Checks:

These resources can be useful to make the above suggested steps/approaches:

1. Introduction to QNLP: An introduction to Quantum Natural Language Processing | by Qiskit
2. Research Paper: https://arxiv.org/abs/2202.11766 (If discrepancies, update here)
3. Code Example(1): https://github.com/PaulaGarciaMolina/QNLP_Qiskit_Hackathon
4. Code Example(2): https://github.com/AkimParis/quantumNLP_jp
5. Code Example(3): https://github.com/gamatos/qnlp-binary-classification
6. qNLP Libraries- Lambeq, QNLP, Quantumtorch
    a. Pipidog/QNLP : https://github.com/pipidog/QNLP
    b. Lambeq : https://cqcl.github.io/lambeq/
    c. NLP with PennyLane-Lambeq : https://cqcl.github.io/lambeq/
    d. Torch Quantum : https://github.com/mit-han-lab/torchquantum