**Quantum Computing Classification Using MiniLM-L6-v2**

**1. Classification Approach & Model Selection**

For scientific text classification, we evaluated two potential approaches:

1. **Traditional TF-IDF + Machine Learning (e.g., SVM, Random Forest, Logistic Regression)**
2. **LLM-based Transformer Models**

Given the nature of the dataset (quantum computing research papers), traditional ML models were unsuitable because they rely on **word frequency statistics** rather than **semantic understanding**. Scientific terminology in quantum computing often carries nuanced meanings that require contextual awareness, making **LLMs a better fit** for this task.

**Model Selection Process**

Since our dataset contains only **≈500 rows**, and we are limited to **NVIDIA RTX 4060**, we compared multiple models:

A screenshot of a computer

AI-generated content may be incorrect.

**Final Decision**: **MiniLM** was chosen because it balances efficiency and performance while working well with small datasets.

**2. Preprocessing & Training Strategy**

**2.1. Data Cleaning**

* **Stopword Removal**: Removed common words such as *"of," "for," "to," "in," "the," "and"* to reduce noise.
* **Character Standardization**: Removed special characters, excess spaces, and ensured text consistency.

**2.2. Data Augmentation**

Given the limited dataset size, we applied **Synonym Replacement** to improve model generalization.

* Replaced **nouns only** to avoid altering quantum computing technical terms.
* Used **WordNet** to find contextually relevant synonyms.
* Conducted **A/B testing** to evaluate whether augmentation improved classification accuracy.

**2.3. Category Keyword Injection**

**Observation**:

* **"Manipulating Qubits for Computation"** and **"Building Qubits"** had significant word overlap (e.g., "Qubit"), leading to misclassification.
* LLMs tend to **focus on surface-level token similarity** rather than deeper semantic meaning.

**Solution**: **Category-specific keywords were injected** to enhance distinction:

E.g.:

category\_keywords = {

**"Models of Manipulating Qubits for Computation":** "Quantum gates, Qubit control, Quantum circuits, Quantum algorithms",

**"Methods of Building Qubits":** "Superconducting qubits, Trapped ions, Quantum dots, Semiconductor qubits",

**"Address Obstacles to Quantum Computation":** "Quantum error correction, Decoherence, Noise mitigation",

**"Applications of Quantum Computing":** "Quantum simulation, Cryptography, Optimization problems" }

By appending **these keywords** to the text input, we reduced category overlap, significantly **decreasing misclassification rates**.

**2.4. Dataset Splitting**

* **80% Training Set**
* **20% Evaluation Set**
* Ensured stratification to **maintain class balance**.

**3. Training Optimization & Fine-Tuning**

**3.1. Hyperparameter Tuning**

A screenshot of a black and white screen

AI-generated content may be incorrect.

**3.2. Preventing Overfitting**

* **EarlyStoppingCallback**: Training stops if **no improvement** is seen for **3 consecutive epochs**.
* **Learning Rate Decay**: **Gradually reduces learning rate** over time for stable weight optimization.

**3.3. Confidence-Based Classification**

* Implemented **Top-2 Category Ranking**:
  + Outputs the two most likely category predictions.
  + Helps analyze **which categories are most frequently confused**.

**3.4. Training Loss Visualization**

To determine the **optimal training epoch**, we tracked loss over time:

A computer screen with white text

AI-generated content may be incorrect.

This ensures that training stops **before overfitting occurs**.

**4. Results & Self-Evaluation**

* **Final model achieved 100% accuracy on the evaluation set.**
* The model correctly classified **every** test instance in the held-out **20% evaluation set**.
* This suggests **strong learning on the available dataset**, but also raises concerns about **overfitting** due to limited data.

**5. Future Improvements**

**5.1. Using a More Specialized LLM**

* **SciBERT** or **PubMedBERT** could further improve performance due to their **pretraining on scientific texts**.
* These models may capture **domain-specific terminology** better than MiniLM.

**5.2. Expanding Training Data with Metadata**

* **Incorporating research metadata** such as:
  + **Citations**
  + **Field classification**
  + **Author expertise**
  + **Affiliated research institutions**
* This would provide additional **contextual signals** to help the model **understand relationships between papers**.