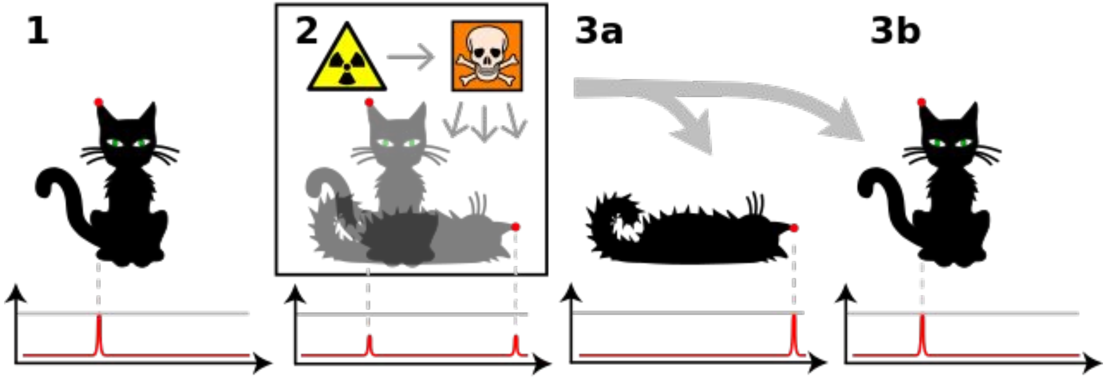
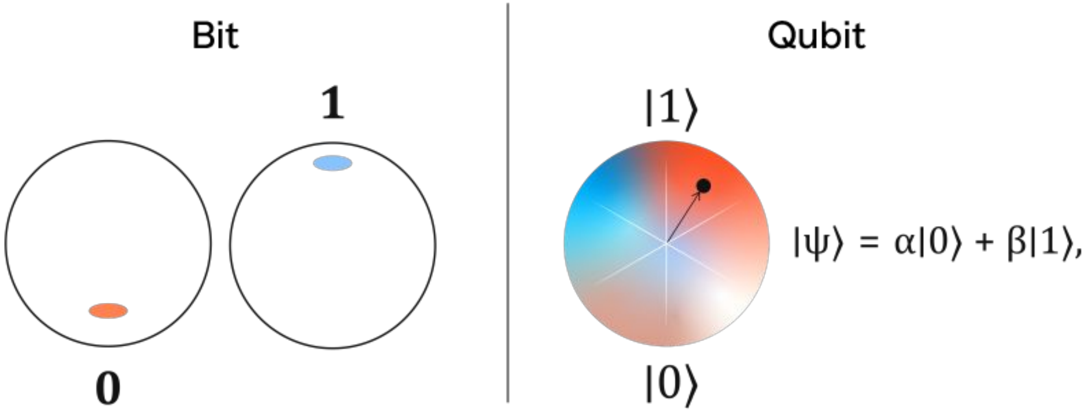
An Effort Towards Quantum Tokenization

Charis Graham

Quantum Computing 101

- Qubits: classical bits in superposition
    - 2-dimensional Hilbert space
    - |0> = [1, 0] and |1> = [0, 1]
    - Measurement: reading the value of a qubit in state |ψ>
    - Joint state: quantum state relating to *n* qubits, denoted |φ>
    - Quantum system/circuit: a computation mapped by quantum logic gates

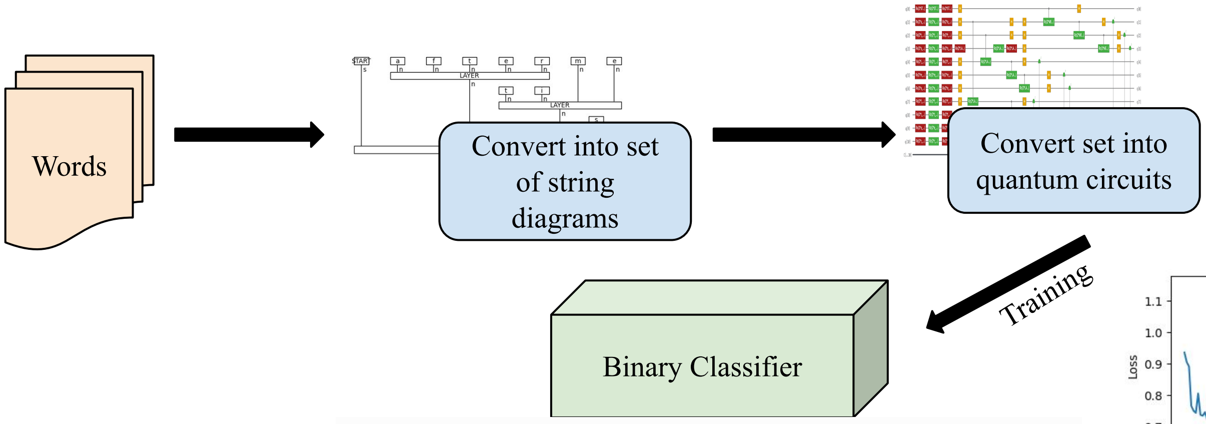




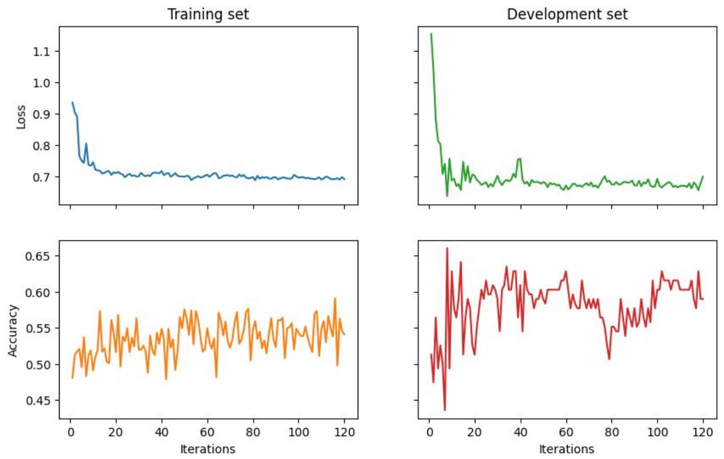
Why quantum tokenization?

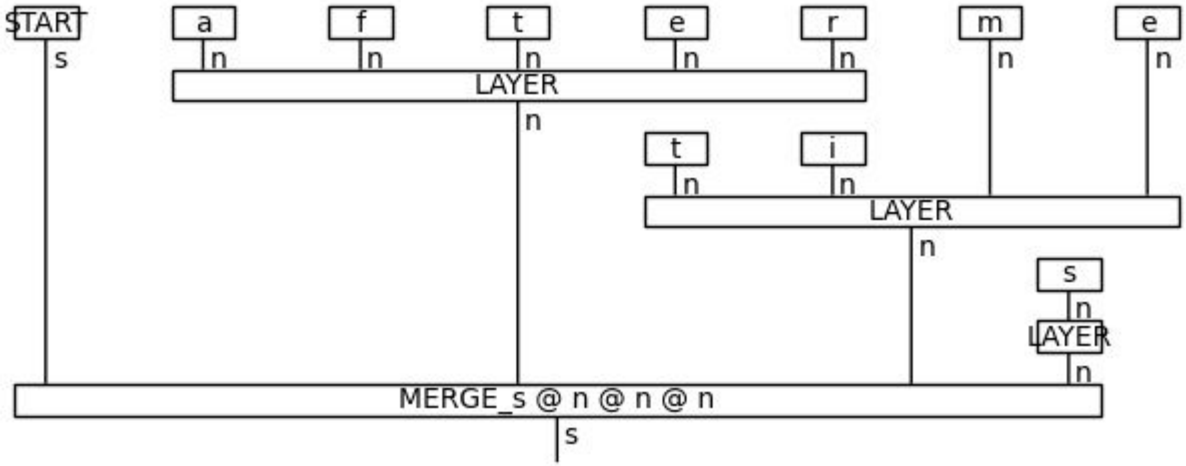
- Language lends itself to quantum representation quite naturally.
- Tokenization is all about trying to represent a large amount of sequential information, so what if we could do it all at once?
- QNLP is very new, so expansion efforts push boundaries.

So how does it all work?

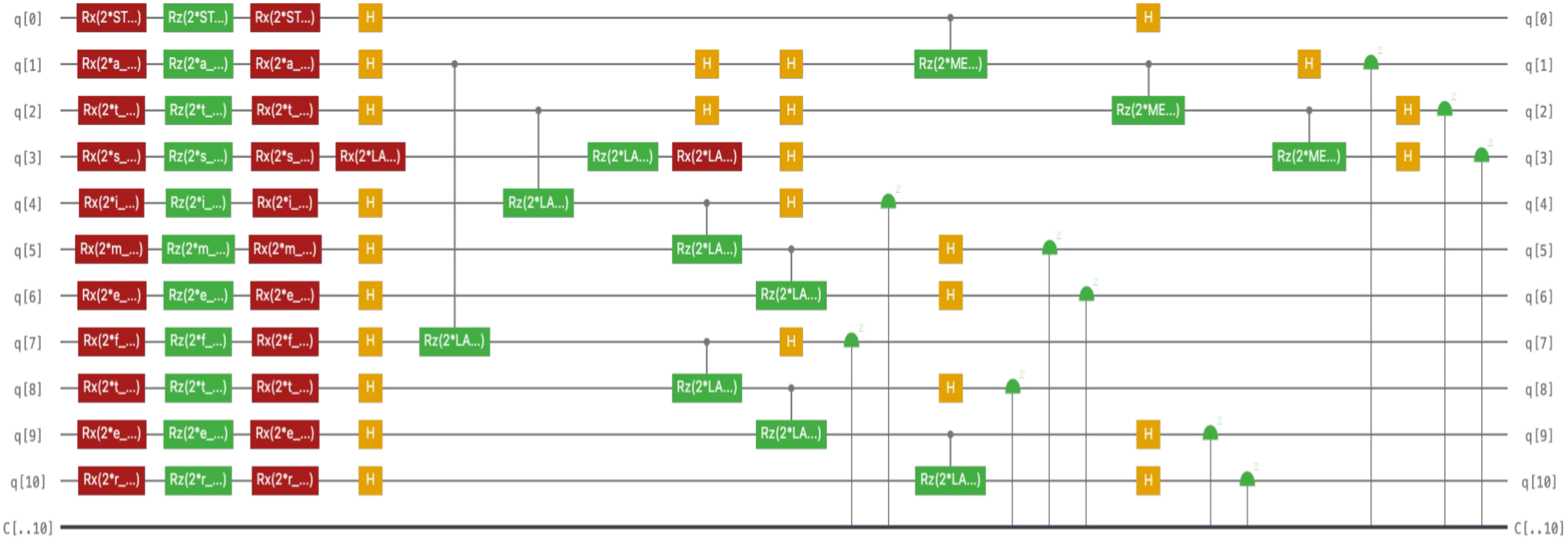


$$H(p,q) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p(y_i)) + (1-y_i)\log(1-p(y_i))]$$



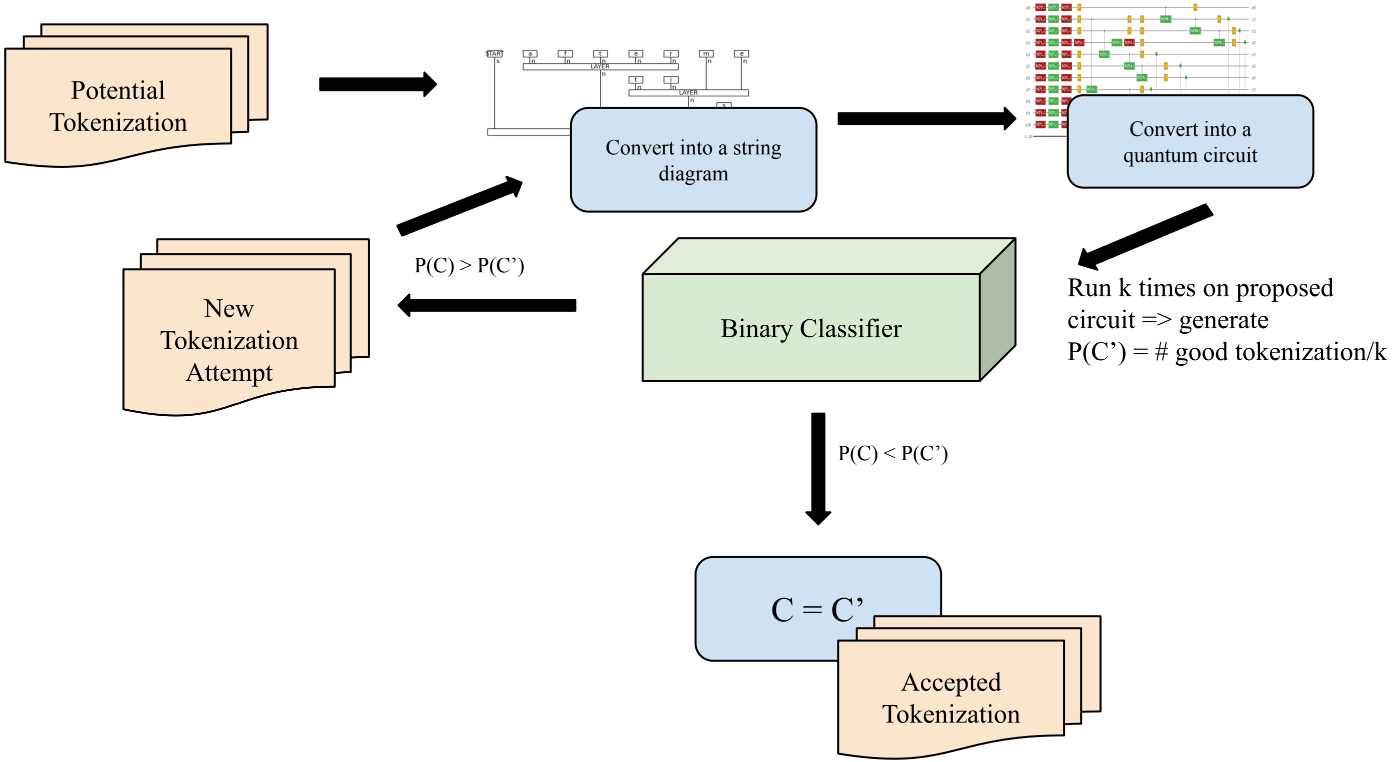


A string diagram representing the tokenization of the word “aftertimes” as “after time s.”



String diagram for “aftertimes” correctly transformed into the representative quantum circuit.

How do we use the binary classifier?

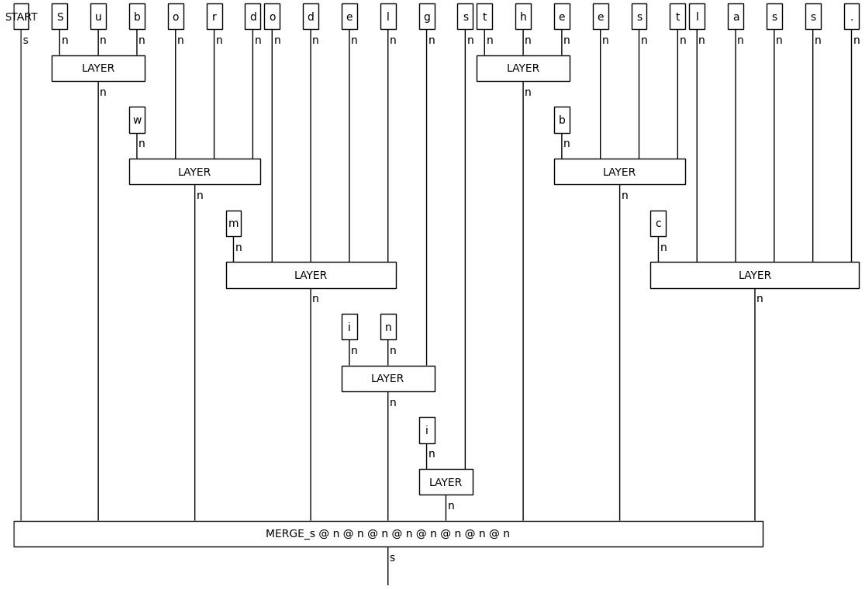


Results So Far

Model	F1 Score	Precision Score	Recall Score
Unigram Sentencepiece	0.15	0.11	0.27
Morfessor	0.11	0.19	0.08
BPE	0.22	0.16	0.36
Quantum Tokenization	0.15	0.12	0.19

Where do we go from here?

1. Try a few other string diagram configurations.
2. Attempt to test on real quantum computer.
3. Expand to sentence level testing.
4. Testing languages other than English.



String diagram for “Subword modeling is the best class.”

Questions?