In the Medium article "An Introduction to Quantum Natural Language Processing," Amin Karamlou, Marcel Pfaffhauser, and James Wootton introduce quantum algorithms for generating natural language and music using IBM Quantum computers. They provide a high-level overview of their work, with more detailed information available in their paper and conference talk recording.

The article discusses Procedural Generation, a field focusing on algorithmically generating content using computers. Quantum computers have been successfully applied to procedural generation tasks, including image manipulation and geopolitical map creation. They also touch upon Natural Language Generation (NLG), which merges procedural generation and Natural Language Processing (NLP), offering applications such as dialogue generation for video games and automatic news article creation.

Their quantum algorithm for sentence generation takes a topic as input and produces a short sentence about that topic. It builds upon previous work in quantum sentence classification, which involves annotating sentences with topics and converting them into parameterized quantum circuits. The Lambeq library facilitates this conversion, allowing sentences to be represented as string diagrams and transformed into quantum circuits.

The DisCoCat model, used in Quantum Natural Language Processing (QNLP), represents sentence meaning based on words and their grammatical relationships. These string diagrams can be transformed into parameterized quantum circuits, encoding sentence meaning. Quantum sentence classification involves finding optimal parameters to classify sentences based on their topics.

A diagram of a diagram

Description automatically generated*(Wires are annotated by types, for instance “s” for a whole sentence, or “n” for noun. If a letter is followed by a “.l” or a “.r” this means that an element of that type is expected either on the left (l) or right (r).)*

For sentence generation, their algorithm treats it as a combinatorial optimization problem. It iteratively generates candidate sentences, evaluates their fitness using quantum circuits, explores neighbouring solutions, and updates the candidate sentence until a suitable sentence is found. This approach is inspired by techniques like hill climbing and simulated annealing.

Algo:

1. Generate a random initial candidate sentence. This could be the sentence “Johnson continues new policy”. Let us call this sentence C.
2. Create the parameterized quantum circuit corresponding to C using Lambeq.
3. This circuit is run many times with the optimal parameters obtained in step 1. The percentage of the runs where the outputted bitstring corresponds to the correct topic is recorded. Let us call this value P(C). So, if we run the circuit 100000 times and observe the bitstring “00” which corresponds to “sports” 12000 times P(C) = 12%.
4. While the value P(C) is less than some threshold, say, 90%, perform the following actions in a loop.
   1. Generate a “neighboring” sentence of C called C’ by either inserting a new random word into C, deleting a random word from C or replacing a random word from C with a new random word.
   2. If P(C’) > P(C) then set C = C’.

