Quantum Natural Language Processing with Lambeq

By Quantinuum

Quantum Natural Language Processing with Lambeq for Sentiment Analysis and Depression Detection

Team (QLinguists)

- Amirali Malekani Nezhad
- Yousra Bouakba
- Vishal Mandal
- Mostafa Shabani
- Pavan Sai Kumar Singavarapu
- Rishab Ghosh

Model Experiments

Preprocessing

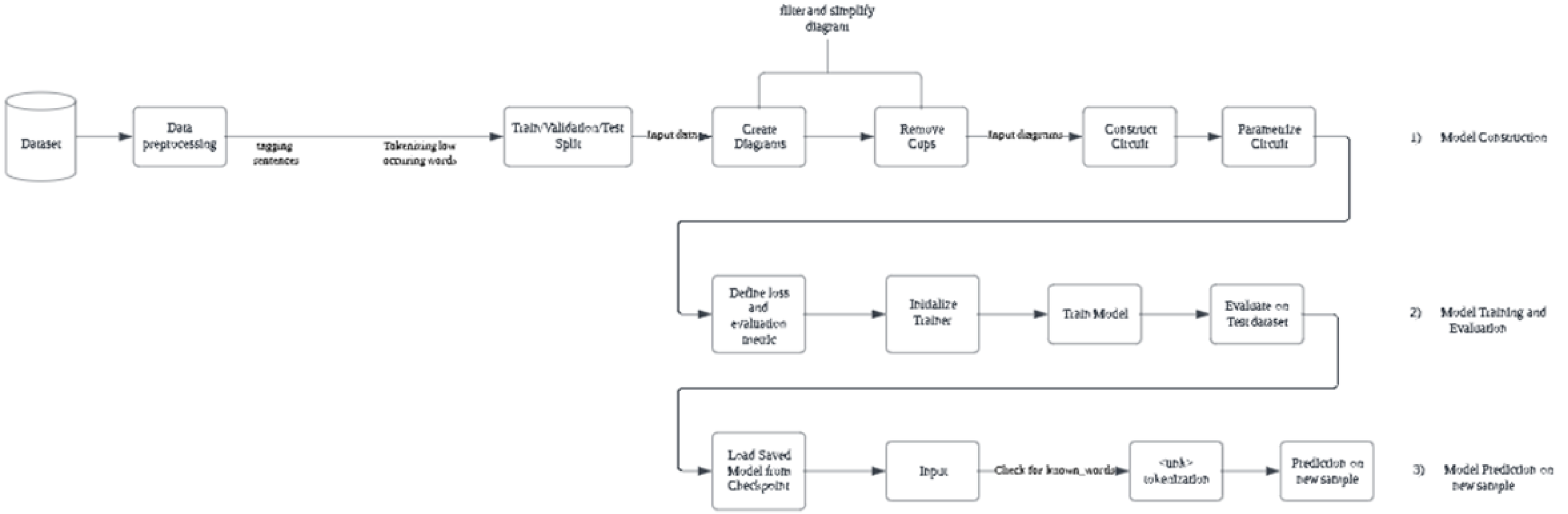
- Preprocessing at sentence level
- Rewriting at diagram level
- Tokenization

Parsing

- Sequence Independent Parsing
- Sequence Dependent Parsing

Circuit mapping

- IQP Ansatz
- Original Ansatz



Model Pipeline

Preprocessing

At Sentence Level

Spelling and Emoji

Correct dictation and remove emojis

Connector

Remove connectors of "that"

Punctuation

Remove dots and commas

Determiner

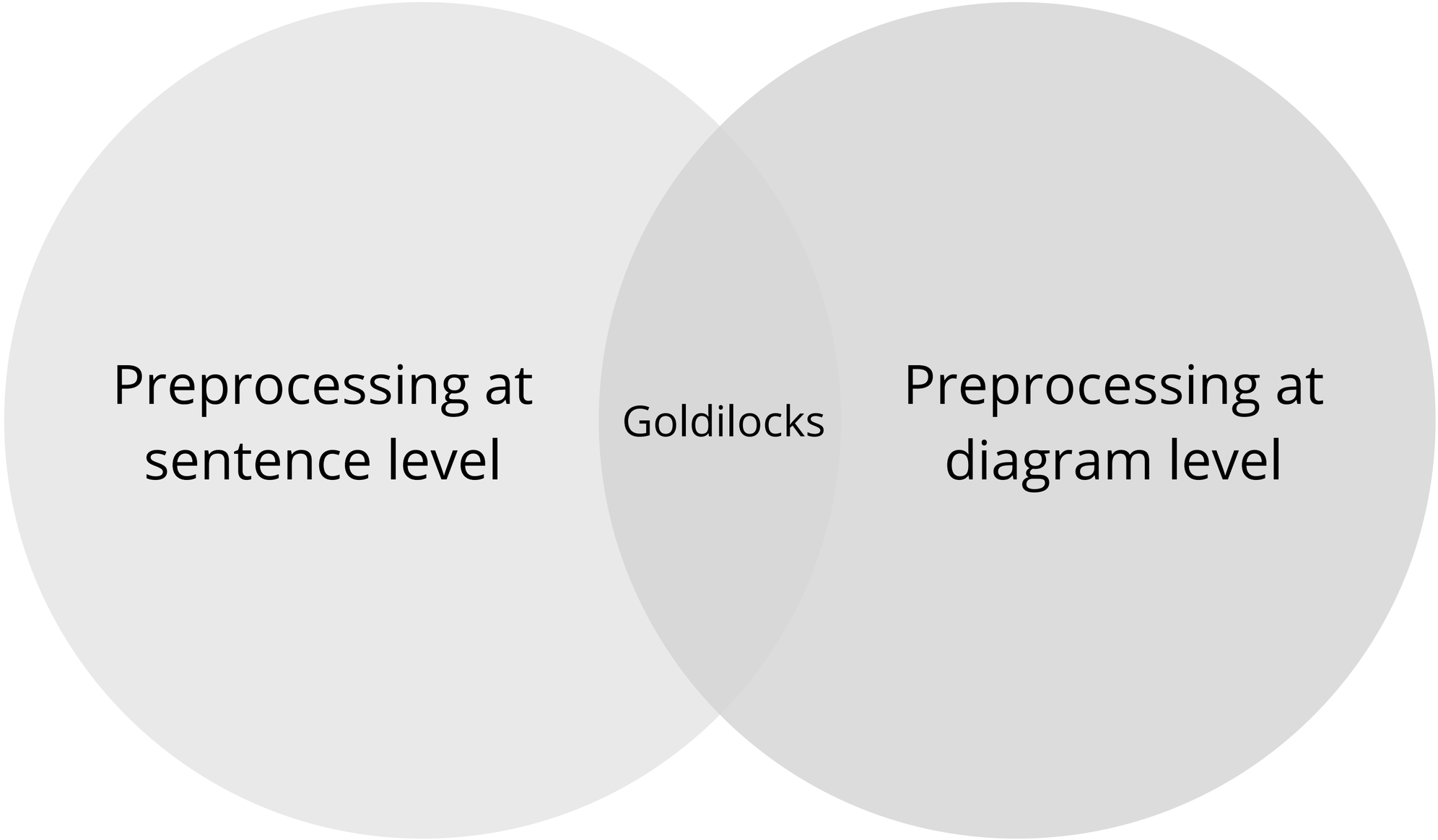
Remove "the" determiner

Auxiliary

Remove auxiliary like "am","is", "are"

Suffix

Tokenize that "I'm" is "I am"


Preprocessing at sentence level
Goldilocks
Preprocessing at diagram level

Parsers

Spiders_reader

Bag of words model, which is sequence independent.

Cups_reader

Simple Syntax model, which is sequence dependent.

Tree_reader

Simple Syntax model, which is sequence dependent.

Bobcat Parser

Heavy Syntax model, which is sequence dependent.

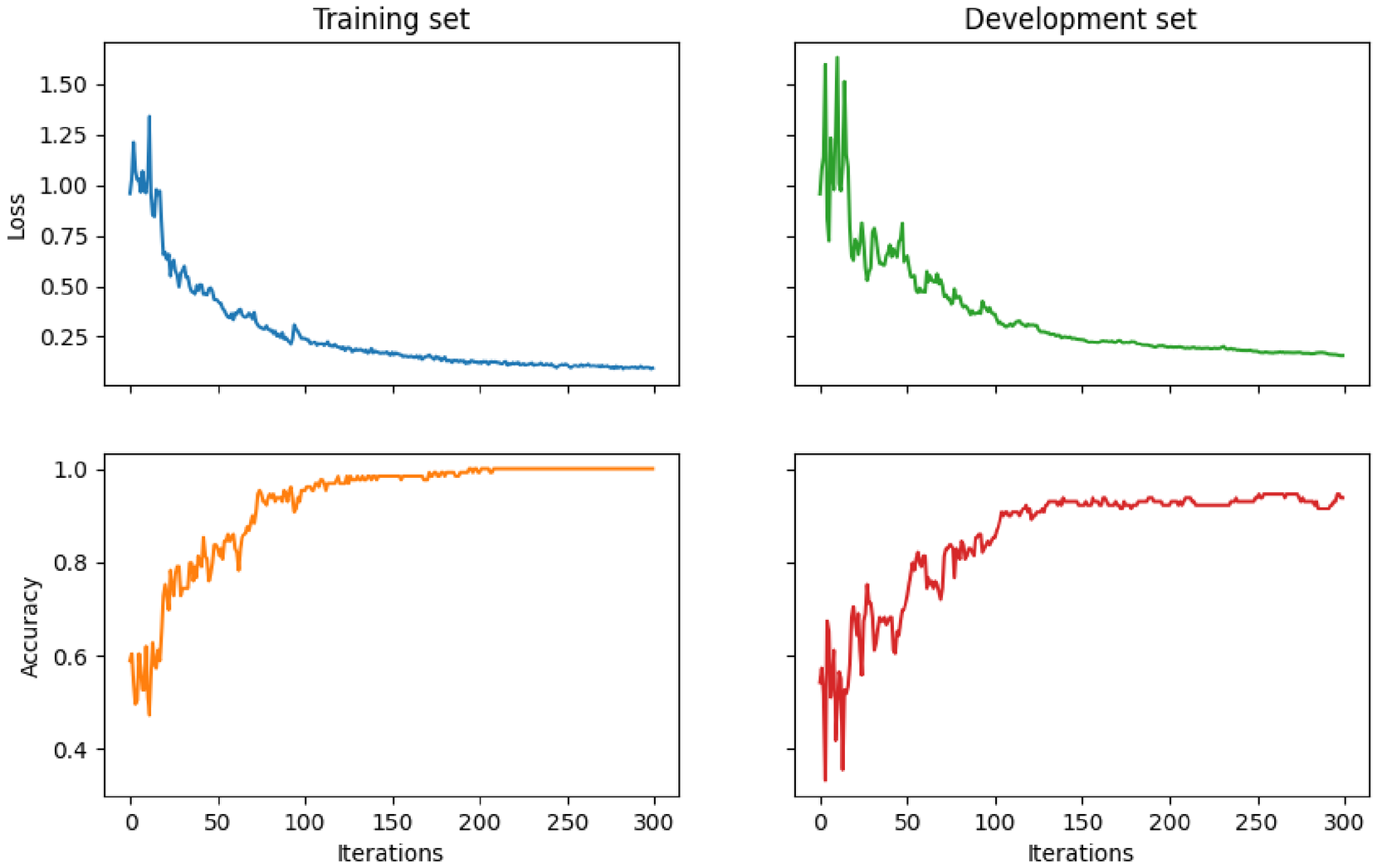
Ansatz

IQP Ansatz

IQP Ansatz model

Sim13ansatz

Self made Ansatz model



Spiders_reader

Summary

Given the field of QNLP (as well as Lambeq itself) being a quite new field of research, there are still a great amount of work to be done in order to allow it to reach its fullest potential and be comparable to classical deep neural nets. Given the use of Qubits there is still much needed hardware advance to be made in order to provide a more powerful model with more qubits and deeper circuits.

Currently the sequence independent model yields an accuracy of %100 on training set, %93 on validation set and %84 on test dataset , and for the sequence dependent model yields accuracy of %87 on training set, %67 on validation set and %55 on test set for a binary sentiment analysis depression detection model. To reach a more optimal model requires further experimentation with different models (given the initial use of TKET and it being incredibly slow currently to use, we migrated to NumpyModel, hence there is still a need to reform the models and test again based on TKET), based on different alpha values, different number of layers as well as different number of qubit representations.