



D6.7: WP6 QNLP Alpha Prototype

Document Properties

Contract Number	951821
Contractual Deadline	M20 (29/04/2022)
Dissemination Level	Public
Nature	Other
Editors	Antonio Villalpando, ICHEC Kaspars Balodis, Tilde Rihards Krišlauks, Tilde Mārcis Pinnis, Tilde Venkatesh Kannan, ICHEC
Authors	Antonio Villalpando, ICHEC Kaspars Balodis, Tilde Rihards Krišlauks, Tilde Mārcis Pinnis, Tilde Venkatesh Kannan, ICHEC
Reviewers	Andrés Gómez, CESGA Vedran Dunjko, ULEI
Date	27/04/2022
Keywords	Quantum Natural Language Processing, prototype
Status	Final
Release	1.0



This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 951821



History of Changes

Release	Date	Author, Organisation	Description of Changes
0.1	14/03/2022	Venkatesh Kannan	Template initiated
0.2	12/04/2022	Antonio Villalpando, Kaspars Balodis, Rihards Krišlauks, Mārcis Pinnis, Venkatesh Kannan	Submitted for First Review
0.3	20/04/2022	Antonio Villalpando, Kaspars Balodis, Rihards Krišlauks, Mārcis Pinnis, Venkatesh Kannan	Submitted for Second Review
1.0	29/04/2022	Antonio Villalpando, Kaspars Balodis, Rihards Krišlauks, Mārcis Pinnis, Venkatesh Kannan	Submitted on EC portal



Table of Contents

1. Executive Summary	4
2. The QNLP Alpha Prototype	5
3. Using the QNLP Alpha Prototype	6
List of Acronyms	7
Bibliography	8



1. Executive Summary

The NEASQC project aims at demonstrating and advancing the capabilities of NISQ-era devices through the development of practically-relevant use-cases. Under the category of Symbolic AI and Graph Algorithmic algorithms, one of the use-cases that is being developed is for Quantum-enabled Natural Language Processing (QNLP). The objective of the QNLP use-case in NEASQC is to investigate, develop and compare existing methods in classical NLP with a QNLP approach to encode and process sentences in a hybrid classical-quantum workflow.

This software deliverable D6.7 titled “QNLP alpha prototype” is an incremental update to the pre-alpha prototype delivered through D6.3 (Villalpando et al., [2021a](#), [2021b](#)). The alpha prototype goes beyond the DisCoCat method (Coecke et al., [2010](#)) used in the pre-alpha prototype to explore and implement an alternate method based on Dressed Quantum Circuits.



2. The QNLP Alpha Prototype

Being a software deliverable, D6.7 “QNLP alpha prototype”, is an intermediate version composed of the approaches that are being explored and evaluated for the targeted QNLP tasks – parallel data extraction and intent detection.

The deliverable D6.3 “QNLP pre-alpha prototype” (Villalpando et al., 2021a) developed an early implementation of the DisCoCat-based model using parameterised quantum circuits to encode a pre-defined dataset of sentences, and trained the parameterised circuits which were evaluated using a test dataset for true/false conditions. Further description of the DisCoCat model is available at (Coecke et al., 2010).

This deliverable D6.7 is a software prototype that adds an additional approach based on Dressed Quantum Circuits (Mari et al., 2020) in which pre-trained classical models are used as pre-processing layers in a transfer learning fashion. Neural Networks are implemented to transform pre-trained word vectors into usable lower dimensional vectors acting as the parameters for the rotations of variational quantum circuits.

The software implementation of this approach is available for internal evaluation at the [NEASQC QNLP GitHub repository](#) under a dedicated [releases](#) branch tagged `v0.2-alpha-d0.9`.

An outline about the Dressed Quantum Circuits approach is summarised at the [README.md](#) file of the alpha prototype, and the dedicated [Jupyter notebook](#).

An outline of instructions for downloading and running the associated Jupyter notebooks for the pre-alpha and alpha prototypes are provided in [README.md](#) file of the NEASQC QNLP GitHub repository, and listed in Section 3.

It is to be noted that both the pre-alpha and alpha prototypes of the QNLP software are works in progress and intended for internal evaluation and testing of the theoretical/algorithmic approaches.



3. Using the QNLP Alpha Prototype

Following is a summary of steps to set up the environment and use the Dressed Quantum Circuits method that is currently under evaluation as a part of the alpha prototype:

1. Ensure python 3.9 is available
2. Install pip (documentation available [here](#))
3. Install JupyterLab (documentation available [here](#))
4. Install Virtualenv (documentation available [here](#))
5. Clone the NEASQC QNLP alpha prototype from releases tagged v0.2-alpha-d0.9
 - `git clone --branch releases https://github.com/NEASQC/WP6_QNLP.git`
 - `git checkout tags/v0.2-alpha-d0.9`
6. Create a new virtual environment within WP6_QNLP
 - `cd WP6_QNLP`
 - `python3.9 -m venv venv_alpha`
7. Activate virtual environment
 - `source ./venv_alpha/bin/activate`
8. Install ipywidgets
 - `pip install ipywidgets`
9. Install virtual environment in JupyterLab
 - `python -m ipykernel install --user --name=venv_alpha`
10. Install base and alpha prototype requirements
 - Purge pip cache (optional) `pip cache purge`
 - `pip install -r requirements.txt`
 - `cd my_lib/Alpha/DressedQuantumCircuit/`
 - `pip install -r requirements.txt`
 - Download spaCy NLP library models `python -m spacy download en_core_web_lg`
11. Run the Dressed Quantum Circuit Jupyter notebook for the alpha prototype
 - `jupyter-lab`
 - Ensure that the `venv_alpha` kernel is used
 - Run the cells in the notebook



List of Acronyms

Term	Definition
NLP	Natural Language Processing
QNLP	Quantum Natural Language Processing

Table 1: Acronyms and Abbreviations



Bibliography

- Coecke, B., Sadrzadeh, M., & Clark, S. (2010). Mathematical foundations for a compositional distributional model of meaning. *arXiv preprint arXiv:1003.4394v1*.
- Mari, A., Bromley, T. R., Izaac, J., Schuld, M., & Killoran, N. (2020). Transfer learning in hybrid classical-quantum neural networks. <https://quantum-journal.org/papers/q-2020-10-09-340/>
- Villalpando, A., Balodis, K., Krišlauks, R., & Kannan, V. (2021a). NEASQC D6.3 QNLP pre-alpha prototype. https://www.neasqc.eu/wp-content/uploads/2021/09/NEASQC_D6.3_WP6_QNLP_PreAlpha_final.pdf
- Villalpando, A., Balodis, K., Krišlauks, R., & Kannan, V. (2021b). NEASQC D6.3 QNLP pre-alpha prototype. https://github.com/NEASQC/WP6_QNLP/tree/v0.1-prealpha-d0.3