Engineering of a QNLP pipeline for privacy NFR classification

Software Engineering for Artificial Intelligence - Project Report vers. 0.1

Marco Calenda
Departement of Computer Science
University of Salerno
Fisciano, Italy
m.calenda10@studenti.unisa.it

Github Repository

***Abstract*—The abstract will be provided in the final report.**
***Index Terms*—**

I. Introduction

In recent years, language models based on Self-Attention [1], namely BERT or GPT-3, putted in the shade simpler neural network architecture such as Recurrent Neural Network (RNN) and Long-Short Term Memory (LSTM) for what concerns Natural Language Processing (NLP) tasks. Most of the popular models are freely accessible, the user data and interactions play a crucial role in shaping their behavior, with all the security issues that come with it. More than that, the involved neural networks require large amounts of data to function properly and have reached a huge amount of parameters (in the order of trillions) leading to high complexity and open researches on the interpretability of the learned information and the explainability of the model responses.

A different and not fully explored alternative, gaining popularity and achievability in the last few years, consists in the application of Quantum Computing to NLP, namely QNLP, in order to overcome the computational complexity of many algorithm theoretically able to disrupt the current NLP capabilities by enhancing the reasonability. The idea aims to model natural language and the underneath compositional structure (grammar and semantics) combining both (i) the distributional approach which relies on statistics about the contexts in which words occur and (ii) the symbolic approach arguing that the meaning of a sentence is the result of the meanings of the words composing it taking into account the grammar rules. So far, the latter has obtained less success in NLP applications while the distributional approach, exploited by Machine/Deep Learning with empirical data, is the pivot of the current state-of-the-art language models. The *de facto* framework used to modelling language in QNLP is the CAtegorical Distributional Compositional (DisCoCat) that allows encoding sentences as string diagrams and monoidal categories [2] based on the pregroup grammar formalism proposed by [3]. Those have a mathematics foundation perfectly suitable for a quantum approach.

The second main area related to this project is software requirements engineering. During the latter software developement phase, one of the most time-consuming and experienced task is the classification of Non-Functional Requirements (NFRs). Unlike the functional ones, that are centrally written and well-formed in the requirement documents, NFRs are not always explicit and most of the times they are concealed into natural languages sentences that are ambiguous and imprecise leading to a not clear idea of what to implement in the final product. Furthermore, the lack of knowledge about NFRs. Furthermore, as shown in many researches [4], [5], the lack of knowledge of NFRs in the early stages of IT projects has a huge impact on the total cost and the failure rate. Accurately automating identification and classification of NFRs is the key for significantly improve and speed up the whole requirements engineering process.

This project aims to exploit the QNLP potential for improving the effectiveness and efficiency of NFRs engineering in systems where machine learning is involved. In particular, it is intended to engineer a QNLP pipeline able to shape privacy NFRs as tensor network and quantum circuit, following the DisCoCat framework, and train/test a classical binary classifier.

This report is structured as follows: Section II introduces the goals of the project both for the SE4AI course and the future work; Section III describes the methodological steps in order to reach the goals. More sections will be included in the Final Report scheduled for the end of May.

II. Goals

This study aims at evaluating how does the DisCoCat framework when applied in a NLP task of text classification. To carry this out the project is intended to build a QNLP pipeline in order to preprocess privacy NFRs, computing the meaning of the sentence following the DisCoCat framework, and prepare them for a binary classification. The output will

Fig. 1. QNLP pipeline with lambeq

be suitable for experiments on (i) classical hardware, shaping string diagram as tensor network, and (ii) quantum computers, by involving a quantum compiler and optimizer in order to produce quantum circuit. In the scope of the course, it will be engineered a classical machine learning pipeline able to elaborate tensor networks in order to train and test a binary classifier and evaluate its performance. Also, the project aim to analyze the impact of the configuration of parameters on the performance of the QNLP solution. The latter will concern the optimisation choices taken during the QNLP pipeline and the hyper-parameters of the machine learning model in order to make feasible a quantum approach on a classical hardware.

Furthermore, extra work will be planned (formalized as research questions) and executed during my MSc thesis study. This will involve quantum simulations both noisy shot-based simulation (a simulation as close as it could be on actual quantum run) and the noiseless one. The experiments could also explore real executions on IBM quantum computers. These next studies will aim to:

- Conduct a comprehensive analysis about the pros and cons of following a quantum approach over classical and explore the capabilities of QNLP.
- Execute a qualitative investigation aiming at understanding in which way a model involving both distributional and symbolic approach is more explainable than a state-of-the-art NLP model that rely mostly on the statistical information of empirical data.

III. Methodological Steps

The main steps in order to meet the goals are illustrated below:

1) Ensure a good knowledge base of what concerns Quantum Computing, QNLP, the DisCoCat framework and all the mathematical/physics aspects behind tensor networks and quantum circuits. This will also include an in-depth analysis of the state-of-the-art techniques related to the problem of interest and a feasibility analysis.
2) Explore the technical usage of (1) lambeq, a Python library for QNLP, (2) PyTorch, a machine learning framework, (3) PennyLane, a framework for quantum machine learning and (4) Qiskit an open-source SDK for working with quantum computers through simulations. This will be limited to ensure the necessary skills in order to complete the SE4AI project by the end of May while further objectives, and so the notions of the mentioned tools, will be deepened post-course.
3) Search for an available dataset of privacy NFRs or extract privacy requirements from a general NFRs dataset. If the search is not successful, it will be ad-hoc created.
4) Prepare the requirements in input for the training by means of engineering a QNLP pipeline. In particular, using lambeq and its backend, Discopy, the requirements will be converted into string diagram and preprocessed as showed in Fig. 1. In the parameterisation stage it will be tuned the *ansätze* for converting tensors into various forms of matrix product states (MPSs) and overcome the tensor dimensional increase of certain words in the sentences that would make the computation infeasible.
5) Train a machine learning model using PyTorch. Afterward, the model performance will be evaluated on the test set, including metrics such as accuracy, precision, recall, and F1 score.
6) Conduct an empirical study to compare the developed DisCoCat-based model and the statistical language models that rely on word-embeddings. Also, analyze the limitations of the classical hardware when following a quantum approach and facing real-word NLP task.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.
[2] B. Coecke, M. Sadrzadeh, and S. Clark, "Mathematical foundations for a compositional distributional model of meaning," 2010.
[3] J. Lambek, "From word to sentence: A pregroup analysis of the object pronoun who(m)," *Journal of Logic, Language, and Information*, vol. 16, no. 3, pp. 303–323, 2007.
[4] R. R. Maiti and F. J. Mitropoulos, "Capturing, eliciting, predicting and prioritizing (cepp) non-functional requirements metadata during the early stages of agile software development," in *SoutheastCon 2015*, pp. 1–8, 2015.
[5] V. Bajpai and R. Gorthi, "On non-functional requirements: A survey," pp. 1–4, 03 2012.