Quantum Natural Language Processing

Venkatesh Kannan[1], Lee James O'Riordan[1], Myles Doyle[1], Fabio Baruffa[2]

[1]Irish Centre for High-End Computing, Ireland
[2]Intel Deutschland GmbH

Motivation

Natural Language Processing (NLP)
- Key benefits include sentiment analysis, relationship extraction, semantic disambiguation, automatic summary.
- Limitations of traditional "bag of words" approach
  - Lacks information about grammatical rules of language.
  - Increase in problem complexity reduces quality of results.
- A recent "Distributed compositional semantics" approach[1]
  - Uses grammatically informed algorithms to compute sentence meanings.
  - Implementation requires large classical computational resources.

Quantum Computing
- Potentially offers dramatic speedup to algorithms which can exploit quantum parallelism.
- Requires quantum versions of classical algorithms.
- Development of software applications
  - Limited by scale, reliability and coherence of quantum devices.
  - Software quantum simulators allow proof of concept applications.

Project Objective
- Implement quantum versions of distributed compositional semantics algorithms to analyse sentence meaning.
- Develop and evaluate solution on the Intel® Quantum Simulator[5] deployed in Irish national supercomputer "Kay".

Partnership
- Irish Centre for High-End Computing & Intel® Corporation.
- Co-funded by Enterprise Ireland & Intel® Ireland.

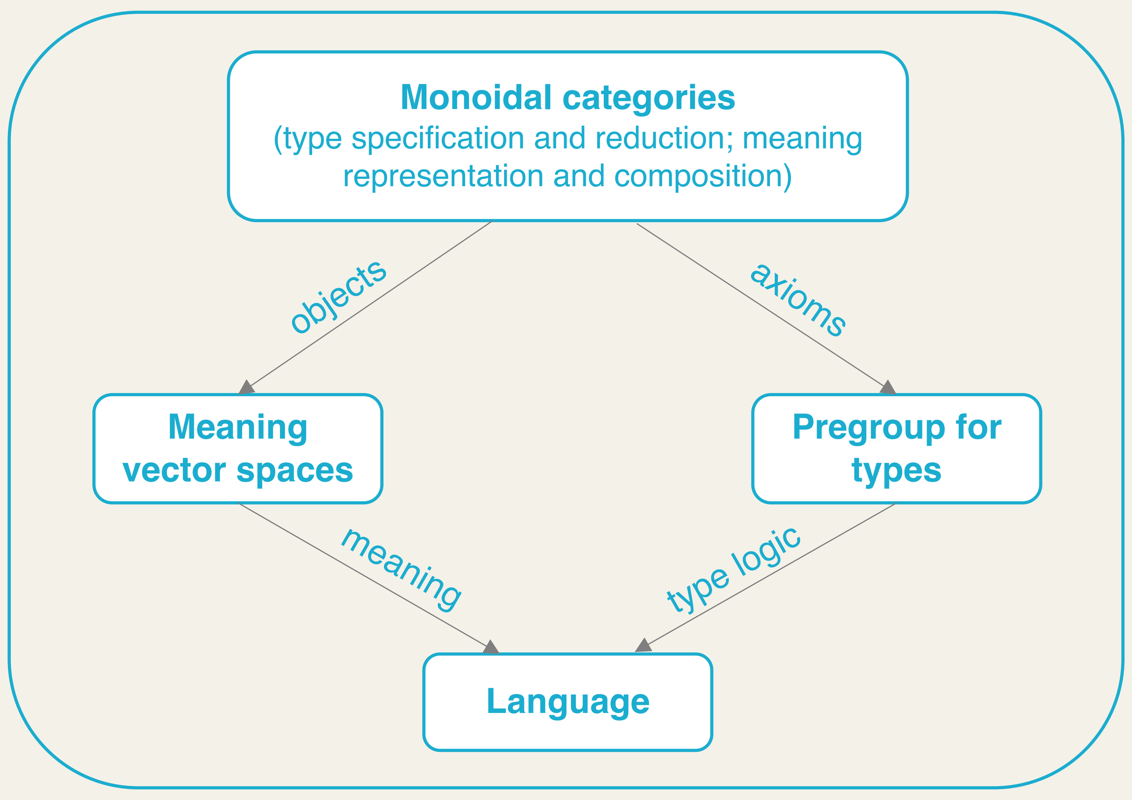
Project Execution
- January 2019 to March 2020.

Distributed Compositional Semantics

Distributional Model
- Algorithms based on "bag of words" approach.
- Meaning of words represented by frequencies of "nearby" words in a corpus.

Compositional Model[2]
- Algorithms derive meaning of sentences or phrases from known meanings of component words.
- Embeds types of words and grammatical structure.

Unified DisCo Model[1]
- Combines both approaches to introduce grammatical form to the composition of word meanings.
- Allows computing meanings of two sentences and decide if their meanings match.



Advantages of Quantum NLP

Challenge
- Classical implementation of DisCo model is expensive in terms of storage requirements and computational complexity.

Quantum advantage for storage[1]
- For example, given the word meaning of a corpus is based on 2000 most common words.

Classical vs. Quantum Storage Requirements

	1 transitive verb	10K transitive verbs
Classical	1 GB	10 TB
Quantum	33 qubits	47 qubits

Quantum advantage for computation[1]
- For N-dimensional M meaning vectors.
- $O(NM)$ for classical DisCo model algorithms.
- $O(c\sqrt{NM})$ for quantum versions of DisCo model algorithms.

Intel® Quantum Simulator

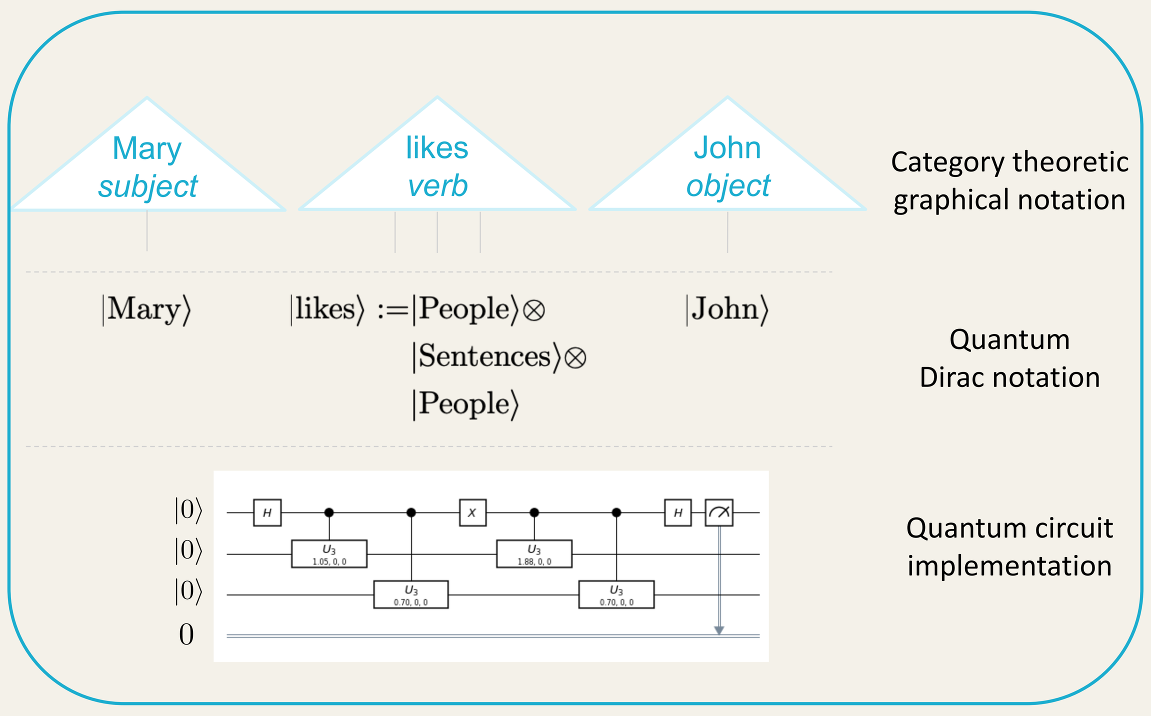
Intel® Quantum Simulator[5]
- Quantum High Performance Software Testing Environment.
- Distributed high-performance implementation of a quantum simulator on a classical computer.
- Out-of-the-box simulation of general single-qubit and two-qubit (controlled) gates.
  - Rotation, Hadamard, Pauli, Square Root of Pauli, Toffoli, SWAP, Square Root of SWAP.
- Single- and double-precision for qubit registers.

Implementation
- Single-node and multi-node implementations of qubit gate operations.
- Multi-node implementation distributes state vectors to fit per-node memory capacity to store states with optimised memory usage for improved communication.

Optimisations
- Vectorisation
  - Loops to implement single and controlled gate operations are data parallel.
  - SIMD execution of gate operation loops using 4-wide AVX-512 instruction set.
- Multi-threading
  - Loops to implement single and controlled gate operations are 2-level nested.
  - Dynamic check of workload to decide nesting level at which to parallelise using multi-threading.
- Communication
  - Gate operations reuse temporary storage for qubit simulation to avoid performance impact from paging.
- Gate Fusion
  - Limited computation by single and controlled qubit operations.
  - Huge performance loss due to memory bandwidth when leaving Last Level Cache (LLC).
  - Gate fusion used to block computations in LLC for single and controlled qubit operations.

Methodology

Problem Mapping
1. Represent category theoretic data structures and NLP operations using Dirac notation.
2. Define quantum versions of DisCo algorithms from literature using Dirac notation.
3. Map elements from Dirac notation to quantum notation (gates/registers/circuits).
4. Map elements from quantum notation to the Intel® Quantum Simulator paradigm.



Implementation Outline
1. Given a corpus $C = \{s_1, \ldots, s_N\}$ and target sentence t.
2. Pre-compute features of C and t.
   - Tag words with types, number of distinct types, number of unique words of each type.
3. Define words in C and t as quantum states.
4. Store quantum states of C and t in a quantum memory.
5. Apply sentence derivation algorithm[1] to compute "head" for each sentence in C and t.
   - Each "head" is vector representation of a word which is composed of the sentence meaning.
6. Apply quantum nearest neighbor algorithm[3,4] on heads of C and t to compute sentence(s) in C that are closest to t.
   - Sentence similarity computed based on inner product of vectors.

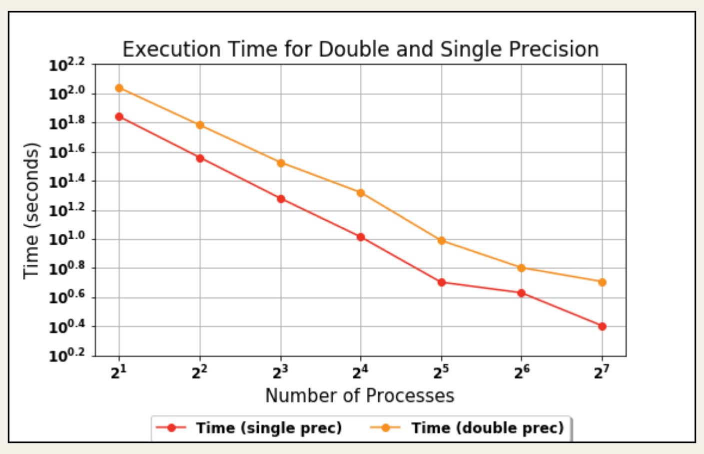
Intel® Quantum Simulator on Kay

Kay
- 336 node cluster.
- 13,440 CPU cores, 63 TB distributed memory.
- Dual-socket 20-core Intel® Xeon Gold (Skylake) 6148 at 2.4 GHz with 192 GB memory.
- 400 GB local SSD scratch.
- 100 GB Intel® OmniPath network.
- Additional partitions
  - Dual NVIDIA Tesla V100.
  - Intel Xeon Phi (Knights Landing architecture).
  - High-memory 1.5TB RAM with 1TB local SSD scratch.

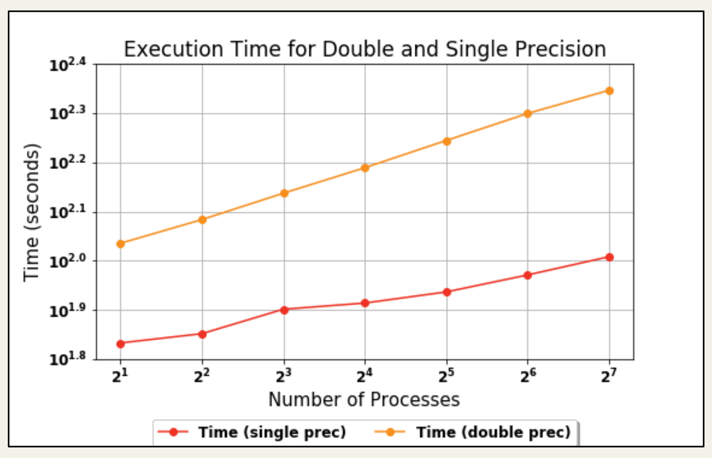
Simulation on Kay
- Up to 33 qubits on single-node executions.
- Up to 41 qubits on Kay's main partition.

Performance Scalability

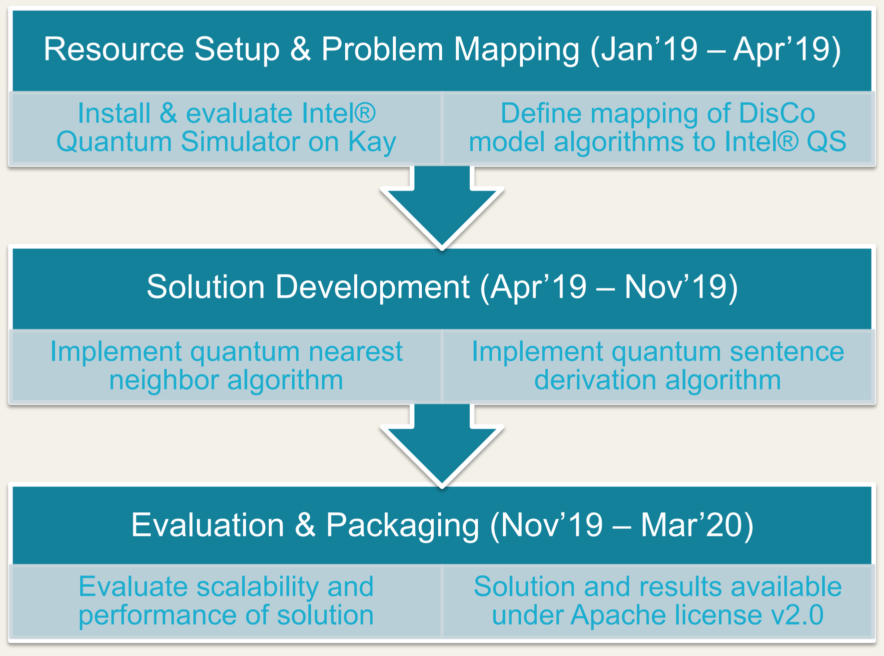
Strong Scaling



Weak Scaling



Project Roadmap



References

1. William Zeng and Bob Coecke, "*Quantum Algorithms for Compositional Natural Language Processing*", EPTCS 221, 2016.
2. Joachim Lambek, "*From word to sentence*", Polimetrica, Milan, 2008.
3. Nathan Wiebe, Ashish Kapoor, and Krysta M. Svore, "*Quantum Algorithms for Nearest-Neighbor Methods for Supervised and Unsupervised Learning*", Quantum Information and Computation, 2014.
4. Schuld, M., Sinayskiy, I. and Petruccione, F., "*Quantum Computing for Pattern Classification*", Pacific Rim International Conference on Artificial Intelligence, 2014.
5. Mikhail Smelyanskiy, Nicolas P. D. Sawaya, Alán Aspuru-Guzik, "qHiPSTER: The Quantum High Performance Software Testing Environment", arXiv:1401.2142v2, 2016.

Contact Information

Irish Centre for High-End Computing (ICHEC)
- Dr. Venkatesh Kannan, `venkatesh.kannan@ichec.ie`
- Dr. Lee James O'Riordan, `lee.oriordan@ichec.ie`
- Mr. Myles Doyle, `myles.doyle@ichec.ie`

Intel Deutschland GmbH
- Dr. Fabio Baruffa, `fabio.baruffa@intel.com`