**Background**

**Natural Language Processing**
- Sentiment analysis, relationship extraction, word sense disambiguation, automatic summary generation.
- Traditional "bag of words" approach
- Lacks information about grammatical rules of language
- Increases in problem complexity reduces quality of results.
- "Distributed compositional semantics" approach
- Grammatically informed algorithms compute sentence meanings.
- Implementation requires large computational classical resources.

**Quantum Computing**
- Potential to offer dramatic speedup to algorithms which can exploit quantum parallelism.
- Requires quantum versions of classical algorithms.
- Application development
  - Limited by scalability, coherence and orientation 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.

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

**Project Execution**

**Distributed Compositional Semantics**

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

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

**Unified Disc Model**
- Combines both approaches to introduce grammatical form to the composition of word meanings.
- Allows computing meaning of two sentences and decide if their meanings match.

**Methodology**

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

**Implementation**

**Two-tier implementation**
1. Intel-QS bindings and quantum encoding implemented in C++ layer (red).
2. Pre-processing, analysis and plotting implemented in Python (blue).
3. External dependencies for respective layers indicated by colour gradients (green/red => C++, greenblue => Python).

Quantum computing applications can be written entirely in C++, or Python. The Python layer allows for a single-ended application development environment, or standard cluster submission with respective Python scripts.

**References**
2. Joachim Lambek, “From word to sentence” , Polimetrica, Milan, 2008.

**Contact Information**
- Irish Centre for High-End Computing (ICHEC)
  - Dr. Lee James O’Riordan, lee.orriordan@ichec.ie
  - Mr. Myles Doyle, myles.doyle@ichec.ie
  - Venkatesh Kannan, venkatesh.kannan@ichec.ie
- Intel Deutschland GmbH
  - Dr. Fabio Baruffa, fabio.baruffa@intel.com