Entangled Meanings: Classification and Ambiguity Resolution in Near–Term QNLP

Chi Zhang, Akriti Kumari, Damir Cavar

Indiana University at Bloomington - NLP-Lab

Problem Statement

A significant effort in **Natural Language Processing (NLP)** is concerned with accurately **classifying text** and **resolving linguistic ambiguities**, especially as data complexity increases [1]. Although QNLP experiments [2, 3] have been done on quantum computers, the datasets are far from realistic. We explore various combinations of word embedding, dimension reduction, and quantum encoding algorithms on standard `lambeq` dataset [4] and a more realistic Amazon review dataset [5], as well as ambiguity resolution on a larger corpus (listed in the abstract), aiming to demonstrate the practical applicability of quantum computing in NLP tasks and extend the boundaries of current QNLP research.

Methodology

- A. Classification
  - Used `lambeq` and **Amazon review** datasets, converting texts with **Word2Vec** `(gensim 4.1.2)` [6] and **spaCy** [7], and applying **PCA, LDA, t-SNE,** from `scikit-learn`[8] and **UMAP** [9] for dimension reduction.
  - Employed **amplitude and divide-and-conquer encoding**[10] with `pennylane 0.36`, training models with **15 iterations and 150 steps per iteration** using a **6:2:2 train:validation:test ratio.**
  - **Evaluated train, validation, and test accuracies** for **each dataset, vectorizer, dimension reduction, and quantum encoding combination** to identify the most efficient configurations for limited computing resources
- B. Ambiguity Resolution
  - Represented language ambiguity by **modeling nouns** as **16-dimensional vectors** and **transitive verbs** as **matrices**, inspired by early disambiguation models [3] classifying nouns into **16 categories**
  - We then **convert** these vectors and matrices into **quantum circuits** by applying **X and CNOT gates**. For example, the quantum circuit for the noun **"Fan," "Admire"** and phrase **"Admire Fan"** is shown in Figures 1, 2, and 3, respectively.

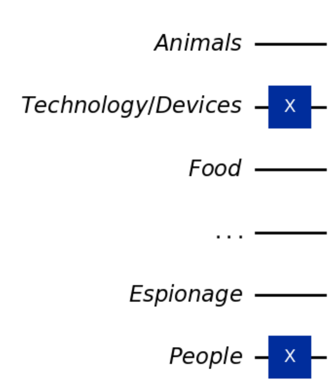

Figure 1. Quantum Circuit for *Fan*

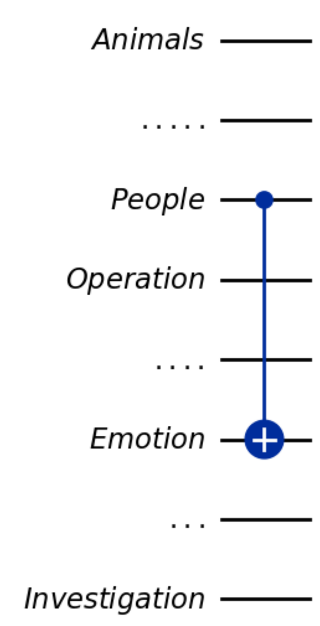

Figure 2. Quantum Circuit for *Admire*

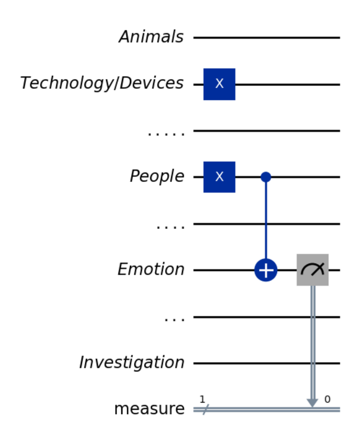

Figure 3. Quantum Circuit for phrase *Admire Fan*

Contributions & Conclusion

- Classification
  - Achieved test accuracy in Table 1 on `lambeq` and Amazon review datasets on `default.qubit` in `pennylane 0.36`, with accuracy $> 80\%$ highlighted.
    - Achieved 100% test accuracy on the `lambeq` dataset using **only 1 qubit** with UMAP and quantum encoding, while authors in [2] used 5 qubits and in [3] used 4 qubits.
    - **Test accuracy** ranged from **55% to 72.5%** on **Amazon reviews**, competitive with **57% to 62%** [3] on **IMDB** [11], highlighting both the potential and limitations of **QML**.

dataset	vectorizer	reduction	quantum encoding	num_qubits	accuracy_train	accuracy_val	accuracy_test
lambeq	Word2Vec	PCA	amplitude	4	0.859	0.6154	0.7308
		PCA	dc	4	0.7821	0.8077	0.8846
		TSNE	amplitude	1	0.6282	0.6923	0.5769
		TSNE	dc	1	0.6026	0.6538	0.5385
		UMAP	amplitude	1	0.9359	0.8846	0.9615
		UMAP	dc	1	0.8205	0.9231	0.7308
		NONE	amplitude	8	0.9359	0.9615	0.9615
	spaCy	PCA	amplitude	4	0.6795	0.8462	0.6154
		PCA	dc	4	0.6795	0.7308	0.6923
		TSNE	amplitude	1	0.5897	0.6538	0.6923
		TSNE	dc	1	0.6538	0.6923	0.6923
		UMAP	amplitude	1	1.0	1.0	1.0
		UMAP	dc	1	0.9359	1.0	0.9231
		NONE	amplitude	16	0.9615	1.0	1.0
amazon	Word2Vec	PCA	amplitude	8	0.575	0.625	0.55
		TSNE	amplitude	1	0.475	0.55	0.575
		TSNE	dc	1	0.5417	0.525	0.6
		UMAP	amplitude	1	0.475	0.6	0.575
		UMAP	dc	1	0.5083	0.6	0.725
	spaCy	PCA	amplitude	8	0.5417	0.65	0.6
		TSNE	dc	1	0.475	0.575	0.55
		UMAP	amplitude	1	0.6167	0.625	0.625
		UMAP	dc	1	0.5833	0.75	0.625
		NONE	amplitude	16	0.6167	0.7	0.575

Table 1. Classification results

- Ambiguity Resolution
  - Tested **16 ambiguous nouns, 34 unambiguous nouns, and 16 transitive verbs**, a much larger corpus than in [3].
  - Randomly selected **2 ambiguous and 2 unambiguous nouns** for testing on `ibm_kyoto` and `qasm_simulator`.
  - Achieved **100% accuracy** on `qasm_ simulator` and **82.1% to 98.9%** on `ibm_kyoto` for ambiguity resolution tasks.

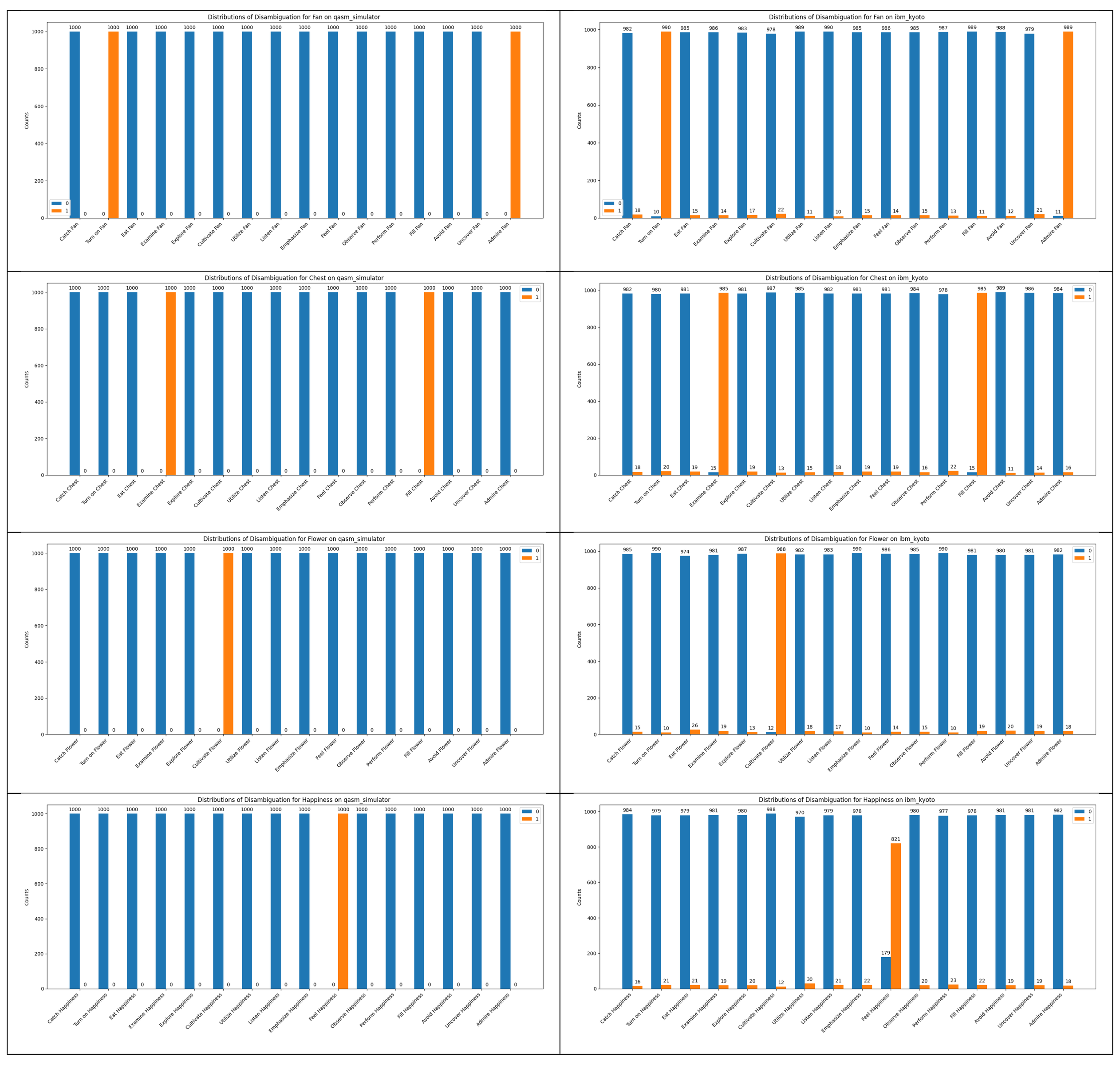

Table 2. Results for Ambiguity Resolution on *Fan*, *Chest*, *Flower* and *Happiness*

Data Availability

All code and data discussed in this work are available in the GitHub repository (`https://github.com/chizhang24/entangled-meanings`). The models used are all freely available online.

Natural Language Processing Lab

The NLP-Lab (`https://nlp-lab.org/quantumnlp/`):

References

[1] D. Peral-García, J. Cruz-Benito, and F. J. García-Peñalvo, "Comparing Natural Language Processing and Quantum Natural Processing approaches in text classification tasks," *Expert Systems with Applications*, vol. 254, p. 124427, Nov. 2024.

[2] R. Lorenz, A. Pearson, K. Meichanetzidis, D. Kartsaklis, and B. Coecke, "QNLP in Practice: Running Compositional Models of Meaning on a Quantum Computer," *Journal of Artificial Intelligence Research*, vol. 76, pp. 1305–1342, Apr. 2023.

[3] D. Widdows, A. Alexander, D. Zhu, C. Zimmerman, and A. Majumder, "Near-Term Advances in Quantum Natural Language Processing," *Annals of Mathematics and Artificial Intelligence*, Apr. 2024.

[4] D. Kartsaklis, I. Fan, R. Yeung, A. Pearson, R. Lorenz, A. Toumi, G. de Felice, K. Meichanetzidis, S. Clark, and B. Coecke, "Lambeq: An Efficient High-Level Python Library for Quantum NLP," Oct. 2021.

[5] D. Kotzias, "Sentiment Labelled Sentences," 2015.

[6] R. Rehurek and P. Sojka, "Gensim–python framework for vector space modelling," *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, vol. 3, no. 2, 2011.

[7] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength Natural Language Processing in Python," 2020.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[9] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction," Sep. 2020.

[10] I. F. Araujo, D. K. Park, F. Petruccione, and A. J. Da Silva, "A divide-and-conquer algorithm for quantum state preparation," *Scientific Reports*, vol. 11, no. 1, p. 6329, Mar. 2021.

[11] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning Word Vectors for Sentiment Analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, D. Lin, Y. Matsumoto, and R. Mihalcea, Eds. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 142–150.