Word and Text Similarity Using Classical Word Embeddings in Quantum NLP Systems

Damir Cavar
Indiana University
Bloomington, IN, USA
dcavar@iu.edu

Koushik Reddy Parukola
Indiana University
Bloomington, IN, USA
koparu@iu.edu

***Abstract*—Word embeddings and vector representations for linguistic units are important in Natural Language Processing (NLP) and AI systems. Engineering of such models involves significant effort, large amounts of data, and costly computations. Many models and systems have been released to the general public, enabling significant research and development in the classical NLP domain. We demonstrate how word and text embeddings as n-dimensional dense vectors of n real numbers in classical computers can be mapped to highly compressed quantum states or quantum embedding using $log(n)$ qubits. In similarity experiments performed on classical and quantum computers, we show no significant information loss that would affect vector distance scores representing words' semantic similarity. The results show that existing NLP embedding models in classical computing environments can be used in quantum computing. We discuss the issues and limitations of this approach in the context of current quantum computing environments.**

***Index Terms*—Word embeddings, Semantic Similarity, Quantum Embeddings, Natural Language Processing**

I. Introduction

In current Natural Language Processing (NLP) and AI applications, vector representations of linguistic units, for example, words, sentences, or paragraphs, are crucial for search algorithms, neural network architectures, and machine learning approaches to language and text processing. There are several reasons for preferring vector representations of these linguistic units. A key reason is technical, i.e., neural network models, which drive most SOTA NLP applications, have a pre-defined and invariant architecture during training and induction phases. These models require input vectors with a specific dimensionality. Since natural language expressions vary in length, encoding them as fixed-length vectors simplifies their use in neural models, particularly in current SOTA Large Language Models (LLMs).

Vectors embedding approximations of word meanings in a semantic vector space are typically trained using distributional semantics assumptions [1]. The distributional properties of words extracted from text corpora provide the empirical basis for training dense vector representations in a self-supervised setting, capturing aspects of word meanings such as similarity in common contexts.

Current SOTA Word embedding models thus provide vector representations for linguistic units geometrically similar to other word embeddings if they share similar contexts in text corpora used for training. For example, *cat* and *dog* share more contexts than *cat* and *moon*, and thus the vector representations for *cat* and *dog* should be closer in Euclidean space than *cat* and *moon*.

That is, the vectorized representations of words in dense vectors are optimized to reflect semantic properties that can be measured in terms of similarity by using term or word collocations in close proximity. In such models, semantically related words are closer to each other than semantically unrelated words, which are measured using linear algebra concepts. The similarity in these linguistic units can be measured using, e.g., *cosine similarity* or *Euclidean distance* between dense vectors (i.e., word embeddings).

Such embedding models are limited to n-dimensional vectors in classical computing environments. In Quantum Natural Language Processing (Q-NLP) or Quantum Machine Learning (QML) [2], such models could be based on much more sophisticated mathematical grounds. In particular, we could think of predicates from a theoretical semantic perspective as different from referential nominal entities or objects. In our models, we treat predicates as quantum gates, i.e., as matrices rather than vectors (see, for example, [3]).

Classical NLP approaches use n-dimensional vectors as word embeddings. Such embeddings could be used in a quantum computer to perform common NLP tasks, such as text classification. However, quantum vector spaces, being physical Hilbert spaces of quantum states [4], provide a much broader framework for modeling semantic properties of natural language units. In NLP, vector space modeling for semantic properties [5] does not use this broad capacity of quantum computing environments.

Given the effort and costs associated with generating language models and word embeddings, reusing existing models is naturally a priority if it is technically feasible and if no significant information loss in the embedding model is associated with the transfer from classical to quantum computers. The work here focuses on, i.e., evaluating the semantic loss involved in encoding classical word embeddings as quantum states.

II. Previous Work

[6] show that storage for word embeddings can be significantly reduced using the quantum-inspired methods *word2ket* and *word2ketXS*. These methods are using amplitude encoding

in a similar way as proposed here, achieving a hundred-fold reduction of storage space for the embeddings "with almost no relative drop in accuracy in practical natural language processing tasks."

[7] demonstrate that encoding methods that map dense classical vectors to quantum embeddings, e.g., basis encoding, angle encoding, or amplitude encoding used in popular Machine Learning (ML) algorithms, can not only be used successfully in Quantum Computing (QC), but the resulting quantum embeddings can in fact "contribute to improved classification accuracy and F1 scores" in the relevant applications.

[8] use amplitude encoding for words in a binary sentence classification task, deriving the word meaning using two different quantum-inspired models. They utilize the Hilbert Space representation of particles in Quantum Mechanics, assigning each word a relative phase (a complex number).

A recent paper discusses quantum embeddings in the context of computer vision [9]. While this is similar to our work here, which only uses quantum embeddings for semantic representations, it does not use quantum embeddings for words or text representations.

To the best of our knowledge, we do not have found experimental results measuring the information loss related to semantic similarity in the different encoding methods used to transfer machine learning models like classical embedding vectors to a quantum computer encoding word embeddings as quantum states.

III. Quantum Word Similarities: Methodology

Classical word embeddings are n-dimensional vectors of real numbers. We use Linear Algebra computations based on the embedding vectors for those words to compare the similarities of two words in the classical environment within one model. Transferring the classical vectors to quantum environments requires an encoding strategy as, for example, basis encoding, angle encoding, or Amplitude Encoding [7], [10] used in popular Machine Learning (ML) algorithms [11].

We encode these vectors as quantum embeddings using Amplitude Encoding. We then measure the quantum similarity between two quantum embeddings using the SWAP test [12], [13].

The classical similarity between two words in our experiments was computed using Cosine Similarity and the model-specific embedding vectors for those words; see Equation 1. Some models are normalized, e.g., the *voyage-3* embeddings. Thus, the Dot-product similarity score with *voyage-3* embeddings is sufficient.

$$cos(\vec{v},\vec{w}) = \frac{\vec{v}\cdot\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}} \tag{1}$$

Our approach to defining the quantum similarity between two words is based on the SWAP test [12], [13]. The SWAP test can be performed between any two circuits S and T of the same number of qubits and can measure the difference between S and T. Suppose the qubits in S are named $s_0, s_1, \ldots, s_{n-1}$ and those in T are $t_0, t_1, \ldots, t_{n-1}$ with an ancillary qubit q_0, the SWAP test performs first the Hadamard gate then the controlled SWAP gate from q_0 to s_i and $t_i, i = 0, \ldots, n-1$ and again the Hadamard gate. It then measures the value of q_0. The circuit for the SWAP tests is plotted in Fig.1.

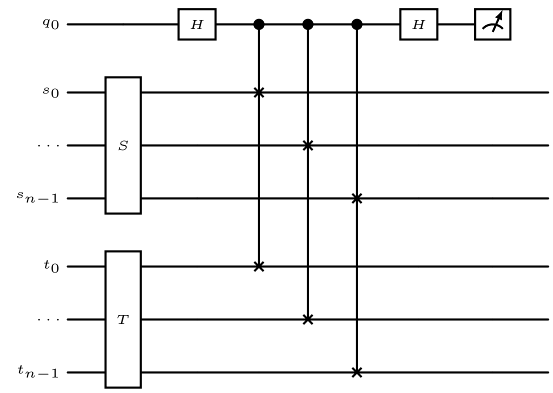

Fig. 1. Quantum circuit for the SWAP test between two circuits S and T

Suppose we use amplitude encoding from [14] to encode 2 $N = 2^n$ real vectors into the circuits S and T. If we write each state $|s_i\rangle$ as

$$|s_i\rangle = \frac{1}{\|s_i\|}\sum_{j=0}^{N-1} s_{ij}\,|j\rangle \tag{2}$$

where $|j\rangle\; j = 0, \ldots N-1$ is the standard basis of the Hilbert space $\mathbb{C}^{\otimes n} \simeq (\mathbb{R}^2)^{\otimes n} \simeq \mathbb{R}^{2^n} = \mathbb{R}^N$, we can write the quantum state $|S\rangle$ of the circuit S as

$$|S\rangle = \frac{1}{\sqrt{N}}\sum_{i=1}^{N-1} |i\rangle \otimes |s_i\rangle \otimes |b_i\rangle \tag{3}$$

where $|b_i\rangle$ is the eigenstate of the Pauli matrix σ_z of eigenvalue $y_i \in \{-1, 1\}$.After a straightforward calculation [15], we know the probability of measuring q_0 having outcome 1 is

$$\begin{aligned} P(1; S, T) &= \frac{1}{4}(1 - \frac{1}{N\sqrt{2}}\sum_{i=0}^{N-1} y_i \langle s_i|t_i\rangle) \\ &= \frac{1}{4}(1 - \frac{1}{N\sqrt{2}}\sum_{i=0}^{N-1} y_i \cos(s_i, t_i)). \end{aligned} \tag{4}$$

Applying SWAP in our experiment, suppose we have two words `word1` and `word2`, and suppose their vector representations in some model, *e.g.* `fastText` are `vec(word1)` and `vec(word2)`. After applying Amplitude Encoding to `vec(word1)` and `vec(word2)`, we denote the corresponding quantum circuits `circ(word1)` and `circ(word2)`, we define the SWAP distance `swap_dist(word1, word2)` between `word1` and `word2` to be

$$\begin{aligned} &\texttt{swap\_dist(word1, word2)} \\ &= P(1; \texttt{circ(word1), circ(word2)}' \end{aligned} \tag{5}$$

which is the probability of measuring 1 from q_0 with $S = \texttt{circ(word1)}$ and $T = \texttt{circ(word2)}$

In the following, we will describe the selection of words from the different classical word embedding models used in the evaluation.

IV. Data and Results

The data set we used in the experiments consisted of two sets of words. One set was randomly generated using the models. The criteria for random selection were:

- All words had to be nouns, i.e., functions words and other lexical categories were filtered out.
- Words had to be longer than a single character, i.e., many tokens in the different models were single-character tokens that we filtered out.
- The selected words had to be present in all models, i.e., we had to ensure that all models provided a vector for the word embedding.

We selected 100 such randomly selected words for our experiments. In addition to the randomly selected words, we hand-crafted a word list of 100 words that belong to different semantic topics or fields, for example:

```
dog cat poodle mouse bird bone food bark paw flag tail
apple tree grape vine hut house cabin condo apartment
```

The approach in this experiment is to pick words from a model and compute the similarity scores using the embeddings of these words. After migrating the embeddings to quantum embeddings, we measure the similarity scores of the quantum states. We analyze the correlation coefficient between the classical similarity scores of word pairs and the quantum similarity scores. A high correlation coefficient would indicate little information loss.

We picked the 100 manually selected words that occur in all models to guarantee that we have access to the individual word embedding vectors. Some embedding models are static, i.e., there will be no embedding for out-of-vocabulary (OOV) words. Other models can compose an embedding for OOV words from tokens that are common subword tokens of words (see, for example, [16]). Current SOTA LLMs use byte-pair encoding (BPE) that is not based on common linguistic intuitions about morphemes or words. Such BPE-based models can generate embeddings as a composition of multiple subword tokens based on BPE tokenization.

Using the randomly generated words, we generated two sets of the cartesian product of each word set with itself, i.e., pairs of words from the randomly selected set of words and pairs of words selected from the hand-picked word set.

For the experiments, we selected 4,400 randomly picked word pairs from each word-pair set, guaranteeing that each word-pair exists in all models. Instead of using the complete set of 10,000-word pairs from the Cartesian product, we chose a subset to reduce the huge computational effort associated with similarity computation, particularly when using quantum simulators. This set of 4,400-word pairs was too large for efficient experimental pre-evaluation, given the complexity of mapping from classical to quantum embeddings and quantum similarity computations.

We computed the Cosine Similarity and the Quantum State Similarity for all pairs using the vector representations from the four models listed above. The quantum state in this experiment represents the quantum embedding for one word based on the amplitude encoding of a classical embedding vector for a particular word.

Using the model embeddings, we used the following models to select word pairs and measure classical similarities (Cosine Similarity). All models were English word lists, but many contained non-English words and non-words or non-linguistic expressions. BERT is not a model that consists of word and vector pairs, but rather a transformer [17] that is a deep learning [18] model that generates a vector representation for some input text.

- Word2vec, 300-dim. word vectors, [19]
- GloVe, 840 billion tokens, 300-dim. word vectors, 2.1 mil. words, [20]
- fastText, 300-dim. word vectors, 2.5 mil. words, [21], [22]
- BERT, 768-dim. word vectors, [16]
- GPT embeddings, 3072-dim., using the *text-embedding-3-large* model, see https://platform.openai.com/docs/guides/embeddings
- Anthropic / VoyageAI embeddings, 1,023-dim.

We expect differences in these models for word similarities as reflected in the geometric distance of the embedding vectors. Meanwhile, Word2vec vectors were generated using continuous bag-of-words (CBOW) and continuous skip-gram to train the word embeddings. In the CBOW algorithm, a specific word is predicted given the context, while in the Skip-Gram algorithms, the words in a local surrounding context are predicted. The resulting word embeddings are not disambiguated, i.e., the word *bank* just has one vector representation, although it is ambiguous, and the different meanings occur in specific contexts. That is, the meaning *financial institution* might occur in a context mentioning *money*, *account*, or *customer*, while the meaning *river bank* would more frequently be surrounded by words like *river* and other terms of nature.

The GloVe model improves this approach by learning word embeddings from global co-occurrence statistics of words in a large corpus. It improves word analogy learning compared to Word2vec.

The fastText model further improves the embeddings by adding sub-word embeddings. Subword embeddings provide vector representations for morphemes that are segments in words, as, for example, in *unloackable* the sequence of *un*, *lock* and *able*. This approach seems to improve the embedding properties for rare words or out-of-vocabulary words for which an embedding can be computed from the word segments.

BERT is trained using the masked word approach. This approach is comparable to the Cloze test [23], where a masked or missing word in a specific context has to be guessed by a human language learner. One of the core advantages of the BERT model is that word embeddings for specific words are dynamically computed depending on the context, providing *contextually informed word embeddings*. Since we are not using context in our experiments here, we always generate the same vector representation for a word using BERT. In subsequent experiments, we intend to test multi-word expressions and more complex utterances, which would benefit from this specific property of BERT.

Another crucial difference between all the other embedding models discussed here, and BERT is that BERT uses the subword tokenization technique WordPiece to generate embeddings for subwords. A word like *locking* could be tokenized as *lock* and *##ing*, and embeddings would be provided for both word segments (or tokens). The special markup of the token *##ing* indicates that *ing* should be treated like a word continuation in the sense of a dependent morpheme in the linguistic sense. This way, BERT reduces the unknown word problem and increases the coverage at the lexical level. The WordPiece algorithm was first laid out in [24] to cope with out-of-vocabulary words and improve the general performance of machine learning and NLP tasks.

Given the fundamentally different underlying machine learning approaches and significant differences in the amount of data and text corpora used to train the different models, we also expect a very different overall similarity score in the classical and quantum embeddings.

V. Evaluation of Similarities

The evaluation steps in this experiment involve: a.) Generating Cosine Similarity (or simply Dot-product) scores for all word pairs using all the different models; b.) Using Amplitude Encoding to convert the word embedding vectors from the classical computing environment to quantum states; c.) Generating the quantum state similarity scores using SWAP-test-based quantum circuit/state similarities, and d.) Computing the Correlation Coefficient between the classical and quantum similarity scores.

There were multiple variables that we could control. Due to the complexity of the experiment and the computing time for the interface between classical and quantum computing, we ran only one variable setting. The quantum measures were based on 1,200 shots per word-pair similarity calculation. We only used the `qasm_simulator` from `qiskit-aer 0.14.2` in the initial phase of the experiment. Currently, we are preparing to run the same experiments on the IBM Quantum computers.

Given the Amplitude Encoding approach, we can compress a 300-dimensional word embedding vector with real number values to nine qubits. This means that, on average, a $300 * 32bit = 9,600bit$ vector can be compressed to just nine qubits. The corresponding 768-dimensional BERT vectors can be represented as ten qubit embeddings. Even a 3,072-dimensional GPT *text-embedding-3-large* embedding can be converted to a 12-qubit quantum state. This compression rate is highly significant and even existing quantum computers allow us to experiment on real hardware with these embeddings. Our goal is to evaluate whether there is some information loss as measured concerning similarity after mapping classical vectors to quantum states.

VI. Results

When interpreting the correlation results between the classical similarity scores with one embedding model and its quantum counterpart, we should be aware of some correlation coefficients between the classical models as such. For example, if we compare the correlation coefficients for the same word pair similarity scores between fastText and Glove vectors, we achieve a correlation coefficient of 0.84. This score has been computed using 4,400 word pairs and the 100 selected nouns.

The computation of quantum similarity scores in our experiments was too complex. Performing the mapping of classical embedding vectors to amplitude-encoded quantum states is a complex process using the algorithms in the IBM Qiskit library (https://docs.quantum.ibm.com/guides). Performing this mapping over the full set of 4,400-word pairs using the embeddings of all available classical models would likely take months in our supercomputer environment (Indiana University's Big Red 200 supercomputer). Thus, our report is restricted to a snapshot of 10 word-pair quantum embeddings using GloVe, BERT, and GPT embeddings.

For the selected word pairs and quantum to classical similarity correlations, we find little and - in fact, not significant - information loss using the approach we suggest. In the simulated initial phase, we measured the correlation coefficient using cosine similarities from the different classical models and the quantum similarities after mapping the classical vectors to quantum states using amplitude encoding. We achieve a correlation coefficient variation between 0.82 and 0.90 for the different models when compared to the quantum similarity using the amplitude-encoded quantum states and the SWAP test.

The loss in the mapping process can be located in all the operations performed. We expect a loss in the mapping from classical to quantum embeddings using amplitude encoding. [25] discuss the loss of encoding techniques, particularly amplitude encoding, using simulators and hardware quantum computers. All encoding techniques introduce a loss. The correlation coefficient between 0.82 and 0.90 indicates that the loss might be acceptable in real implementations of text classification algorithms or similar NLP tasks. Using these amplitude encodings, we demonstrated in [26] that a simple text classification task can outperform previous quantum implementations using embeddings based on dimension-reduced classical word embeddings.

Acknowledgment

The authors are grateful to Larry Moss, the entire NLP-Lab (https://nlp-lab.org/), and the Quantum Study Team at Indiana University Bloomington. The authors acknowledge the Indiana University Pervasive Technology Institute [27] for providing supercomputing and storage resources. This research was partially supported by the Center for Quantum Technologies (CQT) at Purdue University, Indiana University, and the University of Notre Dame.

REFERENCES

[1] A. Lenci and M. Sahlgren, *Distributional Semantics*, ser. Studies in Natural Language Processing. Cambridge University Press, 2023.

[2] 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.

[3] M. Baroni and R. Zamparelli, "Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '10. USA: Association for Computational Linguistics, 2010, p. 1183–1193.

[4] P. A. M. Dirac, *The Principles of Quantum Mechanics*, ser. International Series of Monographs on Physics. Oxford, New York: Oxford University Press, Feb. 1982.

[5] D. Jurafsky and J. H. Martin, *Speech and Language Processing (2nd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2009.

[6] A. Panahi, S. Saeedi, and T. Arodz, "word2ket: Space-efficient word embeddings inspired by quantum entanglement," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=HkxARkrFwB

[7] M. Rath and H. Date, "Quantum data encoding: A comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy," *ArXiv*, vol. abs/2311.10375, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:265281517

[8] Q. Li, S. Uprety, B. Wang, and D. Song, "Quantum-inspired complex word embedding," in *Proceedings of the Third Workshop on Representation Learning for NLP*, I. Augenstein, K. Cao, H. He, F. Hill, S. Gella, J. Kiros, H. Mei, and D. Misra, Eds. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 50–57. [Online]. Available: https://aclanthology.org/W18-3006

[9] J. Zhang, J. Zhou, H. Wang, Y. Lei, P. Cheng, Z. Li, H. Wu, K. Yu, and W. An, "Applications of quantum embedding in computer vision," in *Neural Information Processing*, B. Luo, L. Cheng, Z.-G. Wu, and C. Li, Hongyi ad Li, Eds. Singapore: Springer Nature Singapore, 2024, pp. 165–177.

[10] N. Mitsuda, T. Ichimura, K. Nakaji, Y. Suzuki, T. Tanaka, R. Raymond, H. Tezuka, T. Onodera, and N. Yamamoto, "Approximate complex amplitude encoding algorithm and its application to data classification problems," *Physical Review A*, vol. 109, no. 5, May 2024. [Online]. Available: http://dx.doi.org/10.1103/PhysRevA.109.052423

[11] I. S. Maria Schuld and F. Petruccione, "An introduction to quantum machine learning," *Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015. [Online]. Available: https://doi.org/10.1080/00107514.2014.964942

[12] A. Barenco, A. Berthiaume, D. Deutsch, A. Ekert, R. Jozsa, and C. Macchiavello, "Stabilisation of quantum computations by symmetrisation," 1996. [Online]. Available: https://arxiv.org/abs/quant-ph/9604028

[13] H. Buhrman, R. Cleve, J. Watrous, and R. De Wolf, "Quantum Fingerprinting," *Physical Review Letters*, vol. 87, no. 16, p. 167902, Sep. 2001.

[14] 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.

[15] D. Pastorello and E. Blanzieri, "A quantum binary classifier based on cosine similarity," in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Oct. 2021, pp. 477–478.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. B. et al., Ed. Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013. [Online]. Available: https://arxiv.org/abs/1301.3781

[20] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[21] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[22] A. Joulin, P. Bojanowski, T. Mikolov, H. Jégou, and E. Grave, "Loss in translation: Learning bilingual word mapping with a retrieval criterion," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[23] W. L. Taylor, ""cloze procedure": A new tool for measuring readability," *Journalism & Mass Communication Quarterly*, vol. 30, pp. 415 – 433, 1953. [Online]. Available: https://api.semanticscholar.org/CorpusID:206666846

[24] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 5149–5152.

[25] N. Munikote, A. Li, C. Liu, and S. Stein, "Comparing quantum encoding techniques," 2024. [Online]. Available: https://arxiv.org/abs/2410.09121

[26] C. Zhang, A. Kumari, and D. Cavar, "Entangled meanings: Classification and ambiguity resolution in near–term qnlp," in *Proceedings of the IEEE Quantum Week 2024*, Montreal, Canada, September 2024.

[27] C. Stewart, V. Welch, B. Plale, G. Fox, M. Pierce, and T. Sterling, Bloomington, IN, 2017, indiana University Pervasive Technology Institute.