A Comprehensive Analysis of the Quantum-like Approach for Integrating Syntactic and Semantic Information

Anonymous ACL submission

Abstract

Transformers have proved effectiveness in understanding and deciphering the intricate context of languages. This success is achieved by those models that lack explicit modeling of syntactic structures, which were hypothesised by decades of computational linguistic research to be necessary for logical text understanding. In this work, we present a comprehensive analysis of syntactic and semantic context integration by proposing Compressed Phrase Embedding and adopting quantum-like methods for text classification. We first introduce Compressed Phrase Embedding (ComPhE) by integrating syntactic parsing and semantic contextual information. We test those with two types of quantum-like approaches, 1) quantum-like input processing (DisCoWord) and 2) quantum-like attention (QSA), and discuss the contribution of compressed phrase syntactic and semantic integration towards the model performance on different text classification benchmarks.

1 Introduction

Transformer has dramatically enhanced the performance of tasks in NLP, but they only focus on capturing the contextual semantics of the language, even though the syntactic structure of the sentences are also an essential aspect of a language. Previous studies (Clark et al., 2019; Htut et al., 2019; Mareček and Rosa, 2019; Rogers et al., 2020; Sharma et al., 2022; Zheng and Liu, 2023; Ma et al., 2023) have shown that pretrained transformer-based models like BERT can implicitly capture partial syntactic information. However, this substantial information is a by-product of learning language semantics via self-supervised learning, not by explicitly trying to understand the syntactic structure. Several recent studies (Hu et al., 2020; Du et al., 2023; Zhu et al., 2022b; Li et al., 2022; Zhang and Li, 2022) try to make use of Graph-based Neural Networks (GNNs) to inject such syntactic information into Transformer explicitly. However, GNN research can lead to higher time and space complexity. Few works in machine translation apply the simple token grouping (Bharadwaj and Shevade, 2022) and attention masks (Hou et al., 2022), and there is still room for improvement. Hence, how to incorporate the syntactic and semantic aspects effectively into transformers is also unsettled. Recently, gradient-based training of quantum circuits has been successfully adopted to generate joint distributions over multiple aspects and variables (Delgado and Hamilton, 2022; Zhu et al., 2022a).

This paper presents a comprehensive analysis of syntactic and semantic context integration by proposing Compressed Phrase Embedding and adopting quantum-like methods for text classification. At first, we introduce Compressed Phrase Embedding (ComPhE) by integrating syntactic parsing and semantic contextual information. To integrate the syntactic and semantic aspects of inputs into transformers effectively, we then apply and test two types of quantum-like approaches: quantum-like input processing and quantum-like attention. Firstly, for the quantum-like input processing, we adopt the concept from DisCoCat, which is a computing framework for compositional sentence meaning (Clark et al., 2008; O'Riordan et al., 2020; Lorenz et al., 2023; Kartsaklis et al., 2021). We introduce a new quantum-based representation called DisCoWord to help ComPhE better gather semantic contextual information from the syntactic aspect. Secondly, for the quantum-like attention, inspired by the available quantum modelling of transformer structure (Di Sipio et al., 2022; Cherrat et al., 2022; Shi et al., 2023), we propose a Quantum Self-Attention (QSA), which uses the measurements of the quantum qubits to compute the attention scores. The main contributions are as follows: We investigate the benefit of quantum-like approaches for syntactic and semantic integration by proposing the ComPhE, and adopting quantum-like components on several text classification benchmarks. Note that we introduce all new components, DisCoWord and QSA, to efficiently apply to text classifications, inspired by the existing quantum-like models.

2 Method[1]

We propose Compressed Phrase Embedding (ComPhE), which merges the token embeddings within each phrase after splitting the input sentences into phrases using constituency parsing. This approach aims at syntactic and semantic integration, i.e., gathering semantic contextual information according to syntactic parsing. Then, we apply ComPhE with two quantisation approaches: Firstly, we augment the vanilla token embeddings by adding DisCoWord, another word embedding that uses quantisation to combine both syntactic and semantic context information. Secondly, we adopt quantisation to improve the efficiency of the self-attention module. Specifically, we use quantisation in the attention score calculation to encode 2^n dimensional Q/K vectors into n qubits quantum states and use quantum states to model the semantics. We adopt the joint learning ability of quantum circuits since those have been adopted to generate joint distributions over multiple aspects and variables.

2.1 Compressed Phrase Embedding

We propose Compressed Phrase Embedding (ComPhE) that utilises constituency parsing to gather semantic contextual information from the syntactic aspect. Based on the constituent tree of the inputs, we design two phrasing methods for splitting the words into phrases. The first is Top-to-Bottom (T2B), which generates phrases whose tag range is all Penn Treebank II Constituent Tags. The second is Bottom-to-Top (B2T), which only generates NP, PP, and VP phrases. In detail, T2B goes through the tree except for leaf nodes in a level order traversal from the second to the bottom layer and checks the subtree starting from each no-leaf node. If there is no non-root node with the same tag as the subtree's root, the subtree is determined as a phrase. B2T goes through the tree except for leaf nodes in a level order traversal starting from the bottom to the second layer, splitting all of the minimal NP and PP. All of the other words will be seen as the VP phrases. In addition, both methods merge the consecutive single words as a new phrase. We compare the two phrasing methods to demonstrate which syntactic aspects are more critical to gathering semantic contextual information. After splitting, we apply stemming and stopword removal in phrases. Then we tokenise the words in phrases. Finally, we obtain ComPhE by summing the token embeddings phrase-wise.

[1]The Overview of Model and Test Architecture can be found in Appendix A.1

2.2 DisCoWord Representation

We introduce a new quantum-based representation called DisCoWord to help ComPhE better gather semantic contextual information from the syntactic aspect. DisCoWord can represent the semantic meanings of a word according to its contextual grammar structure, i.e., the pregroup grammar (Lambek, 1999) of the word in a context. This approach is inspired by DisCoCat (Lorenz et al., 2023), a distributional compositional categorical model which uses pregroup grammar to compute the meaning of words in a quantum way and can learn the grammar-based contextual representations for words. Based on DisCoCat, we use pretrained word embeddings to initialise the parameters of the quantum circuit representing the word, which make the representations hold both the syntactic and semantic information based on their contextual grammar structure and pretrained word embeddings. See more details in Appendix A.2.

2.3 Quantum Self-Attention

We propose Quantum Self-Attention (QSA), which uses the measurements of the quantum qubits to compute the attention scores, instead of using dot-product between Q and K in a head. In more detail, we design a quantum circuit containing n input qubits and m output qubits to help with that. First, we use a quantum feature map to transform each Q/K vector belonging to a head to a quantum state of the n qubits. After that, we apply parameterised quantum gates on the n input qubits and the m output qubits, of which the parameters will be trained in the training process. In the end, we use measurement values of the m output qubits for each Q/K vector to compute the attention score.

3 Experiment Setup

We articulate how to evaluate our ComPhE with DisCoWord and QSA on text classification.

3.1 Datasets

We use four widely used text classification benchmark datasets, including Movie Reviews (MR)

Statistics	MR	Twitter	SST-2	OffensEval
Split	7108/3554	8000/2000	6920/1821	11916/1324
Doc.	10.78/4.96	5.32/3.21	9.91/5.01	8.57/6.14
Emb.	9.94/26.71	7.20/21.70	9.87/26.82	7.84/19.64
Dropped	172	913	128	573

Table 1: The summary statistics of datasets. The split represents the Train/Test split. Doc. and Emb. represent document and embedding length, with average/standard deviation values. We dropped the documents which failed to be transformed into diagrams or with the word having lengthy states (longer than 256).

(Pang and Lee, 2005), OffensEval (Zampieri et al., 2019), SST-2 (Socher et al., 2013) and Twitter [2] in our experiments. These datasets are binary classified, and their text lengths are short enough for running quantum computing on classical computers with the quantum simulation software. All the datasets are preprocessed by lowering case, stemming, removing punctuation and removing stopwords. The statistics of the four datasets and the DisCoWord embeddings for each dataset are presented in Table 1.

3.2 DisCoWord Training

For DisCoWord training, we use BobcatParser[3], the state-of-the-art statistical Combinatory Categorial Grammar (CCG) parser (Clark, 2021). We choose the same ansatz (Hadfield et al., 2019) for each grammar type (Lambek, 2008) used in (Lorenz et al., 2023) with 1 qubit for resource saving. Then we use the Simultaneous Perturbation Stochastic Approximation (SPSA) (Spall, 1998) algorithm and CrossEntropy loss for optimisation. The SPSA algorithm is an efficient gradient approximation method only using the value of the object function, i.e., applying the random perturbation on the parameters and calculating the approximated gradient. Thus, it can deal with quantum simulation optimization, which is challenging to calculate gradients directly and usually has noise. The hyperparameters of the SPSA algorithm are referred to as the one from (Spall, 1998). As for pretrained word embeddings, we use glove-wiki-gigaword-50, glove-twitter-25, fasttext-wiki-news-subwords-300, and fasttext-twitter-100 (Camacho-Collados et al.) to initialise the parameters of the quantum circuit representing the word and choose the best DisCoWord according to the test accuracy. See more details in Appendix B.2.

[2]A built-in dataset in NLTK (Bird et al., 2009) library.

[3]we use the BobcatParser implemented in lambeq (Kartsaklis et al.) library.

3.3 Quantum Self-Attention Design

For the quantum circuit we used in the self-attention module, the number of input qubits n is determined by the number of attention heads, the dimension of Q/K vectors, and the future map. In our experiments, we use AmplitudeEmbedding feature map (Jaeger, 2007; Möttönen et al., 2005) to encode the 2^n dimensional Q/K vector to the quantum state of the n input qubits. To save hardware resources, the number of output qubits m is set to 1, i.e., the last input qubit is set as the output qubit. In that case, each Q/K vector will be transformed into a measurement value. We then apply the block including a $R_X(\theta)$ gate, a $R_Y(\theta)$ gate and a $CNOT$ gate on every two qubits from top to bottom, where θ is the randomly initialised trainable parameter. Finally, we apply the $Pauli - Z$ gate (DiVincenzo, 1998) as the measurement operator on the output qubit. The details of the quantum circuit can be found in Appendix B.3.

3.4 Method Verification

We verify our method in the text classification task on the four aforementioned datasets. We use Self-Attentive Encoder (SAE) constituency parser proposed by (Kitaev and Klein, 2018) for ComPhE. We use AdamW (Loshchilov and Hutter) optimiser and CrossEntropy loss in training. We use early stopping to get the test accuracy as the evaluation metric. We have three variations for testing: ComPhE (Vanilla), which applies ComPhE on Vanilla Transformer. ComPhE (DisCoWord), which applies ComPhE on Vanilla Transformer with DisCoWord augmented input. ComPhE (QSA), which applies ComPhE on the QSA augmented Transformer with Vanilla Transformer input.

4 Results

4.1 Performance Evaluation

We compare the performance of baselines and our ComPhE with variations on four text classification datasets. From Table 2, we can see that our ComPhE variations are better than all baselines, which proves the ability to improve text classification. More specifically, using ComPhE (Vanilla) is better than the Vanilla Transformer and most baselines, demonstrating the effectiveness of integrating semantic context information from the syntactic aspect. ComPhE (DisCoWord) does not improve the accuracy except on the Tweet dataset.

Methods	MR	Tweet	SST-2	OffensEval
TFIDF+LR	75.5	68.4	80.1	77.6
CNN-Rand	70.9	99.1	75.3	72.7
CNN-Pretrained	72.0	93.7	74.9	70.0
LSTM-Rand	66.4	92.5	67.8	62.5
LSTM-Pretrained	71.5	87.0	69.6	64.4
Vanilla Transformer	74.8	99.8	73.2	76.9
ComPhE (Vanilla)	75.1	99.8	80.1	**78.0**
ComPhE (DisCoWord)	74.1	**99.9**	78.5	77.9
ComPhE (QSA)	**75.9**	**99.9**	**80.8**	73.7

Table 2: Overall performance comparison with the baselines and the ComPhE variations. The ComPhE variations are better than the baselines overall; TFIDF+LR and CNN-Rand are competitive. Note we mainly focus on comparing with Vanilla Transformer and other classical methods to demonstrate the effectiveness of our approach. We mainly focus on Transformer variants because our proposed methods can be added to other Transformer-based models.

We believe the reason is that the quality of DisCoWord is not good because we use as few qubits as possible and split the training process due to the limited hardware resources. ComPhE (QSA) gets the best performance, 75.9 in MR, 99.9 in Tweet and 80.8 in SST-2. We draw the following inferences from the results. **a**) Evolution to quantum states can replace dot-product self-attention. **b**) OffensEval aims to identify offensive documents from English tweets, which may place a higher demand on text comprehension. Therefore, the simple quantum circuit used in the experiment does not convert the classical vectors of this dataset into quantum states well. As for the Tweet dataset, whose average document length is the shortest, it may be too easy to learn by the model, resulting in the close performances from the three ComPhE variations.

4.2 Compressed Phrasing Analysis

To better understand the impact of the phrasing methods after constituency parsing, we apply our ComPhE with T2B and B2T phrasing methods, respectively. Table 3 shows that both B2T and T2B methods outperform Vanilla Transformer on the four datasets, and are very close while B2T is slightly better. However, when using T2B to get ComPhE, the average token length of the input is 17.6% to 31.4% less than when using B2T [4]. Therefore, we believe using the T2B method will be more advantageous in processing long text.

[4] The details of input token length statistics can be found in Appendix B.4

Method	MR	Tweet	SST-2	OffensEval
w/o-phrasing	74.8	**99.8**	73.2	76.9
ComPhE (B2T)	**75.1**	**99.8**	**80.1**	77.5
ComPhE (T2B)	74.8	**99.8**	79.5	**78.0**

Table 3: Performance comparison between ComPhE test results using different phrasing methods. w/o-phrasing represents the Vanilla Transformer.

4.3 Positional Encoding Analysis

We also test the impact of positional embedding for ComPhE input handling. As can be seen in Table 4, Rel. PE is better on the OffensEval dataset, while Abs. PE (sum) is better on MR and SST-2 datasets. We believe that this discrepancy is caused by differences in the datasets. Offenseval's data is derived from Twitter, so its text is less rigorous than that of MR and SST-2. Therefore, the relative positional embedding, added at the attention layer and can supply phrase position information, indicates that phrase position information is more important than token position information in the Twitter text. As for the absolute positional embedding, which is added to the token embeddings and influences the gathering process and the quality of the ComPhE, is better than relative positional embedding on the MR and SST-2 datasets. Since the Tweet is the dataset with the shortest average document length, positional embedding may be useless. Therefore, these positional embeddings' performances are close on the Tweet dataset.

PE	MR	Tweet	SST-2	OffensEval
w/o-PE	72.7	**99.9**	77.9	78.2
Abs. PE (sum)	**75.1**	99.8	**80.1**	78.0
Abs. PE (cat)	72.2	**99.9**	76.8	77.3
Rel. PE	74.4	99.8	78.7	**78.9**

Table 4: The positional embedding analysis results. All the experiments use ComPhE (Vanilla).

5 Conclusion

We propose Compressed Phrase Embedding, which integrates syntactic parsing and semantic contextual information, and apply it with DisCoWord and QSA on four text classification tasks. Our ComPhE, which gathers semantic contextual information using syntactic parsing, can help understand the text. Both DisCoWord and QSA can enhance the performance of ComPhE. Hence, we hope that ComPhE with quantum-like approaches will be a good reference for integrating syntactic and semantic information and introducing quantum machine learning in text classification tasks.

Limitations

The limitation of our work comes from the hardware resources for running quantisation parts. Since we use the quantum simulation software instead of the quantum computer, the memory usage increases exponentially, i.e., the space complexity is $O(2^n)$ where n is the number of qubits. Therefore, we can only use as few qubits as possible in DisCoWord training and QSA, which limits the performance and parameter searching. In addition, we do not analyse the impact of different syntactic parsing methods, which is left to future work.

References

Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M Sohaib Alam, Guillermo Alonso-Linaje, B Akash-Narayanan, Ali Asadi, et al. 2018. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*.

Shikhar Bharadwaj and Shirish Shevade. 2022. Efficient constituency tree based encoding for natural language to bash translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3159–3168.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Jose Camacho-Collados, Yerai Doval, Eugenio Martınez-Cámara, Luis Espinosa-Anke, Francesco Barbieri, and Steven Schockaert. Learning crosslingual embeddings from twitter via distant supervision.

El Amine Cherrat, Iordanis Kerenidis, Natansh Mathur, Jonas Landman, Martin Strahm, and Yun Yvonna Li. 2022. Quantum vision transformers. *arXiv preprint arXiv:2209.08167*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286.

Stephen Clark. 2021. Something old, something newgrammar based ccg parsing with transformer models.

Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. 2008. A compositional distributional model of meaning. In *Proceedings of the Second Quantum Interaction Symposium (QI-2008)*, pages 133–140. Citeseer.

Andrea Delgado and Kathleen E Hamilton. 2022. Unsupervised quantum circuit learning in high energy physics. *Physical Review D*, 106(9):096006.

Riccardo Di Sipio, Jia-Hong Huang, Samuel Yen-Chi Chen, Stefano Mangini, and Marcel Worring. 2022. The dawn of quantum natural language processing. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8612–8616. IEEE.

David P DiVincenzo. 1998. Quantum gates and circuits. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):261–276.

Mingzhe Du, Mouad Hakam, See-Kiong Ng, and Stéphane Bressan. 2023. Constituency-informed and constituency-constrained extractive question answering with heterogeneous graph transformer. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems LIII*, pages 90–106. Springer.

Stuart Hadfield, Zhihui Wang, Bryan O'gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas. 2019. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2):34.

Shengyuan Hou, Jushi Kai, Haotian Xue, Bingyu Zhu, Bo Yuan, Longtao Huang, Xinbing Wang, and Zhouhan Lin. 2022. Syntax-guided localized self-attention by constituency syntactic distance. *arXiv e-prints*, pages arXiv–2210.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R Bowman. 2019. Do attention heads in bert track syntactic dependencies? *arXiv preprint arXiv:1911.12246*.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of the web conference 2020*, pages 2704–2710.

Gregg Jaeger. 2007. *Quantum information*. Springer.

Dimitri Kartsaklis, Ian Fan, Richie Yeung, Anna Pearson, Robin Lorenz, Alexis Toumi, Giovanni de Felice, Konstantinos Meichanetzidis, Stephen Clark, and Bob Coecke. λambeq-an efficient high-level python library for quantum nlp.

Dimitri Kartsaklis, Ian Fan, Richie Yeung, Anna Pearson, Robin Lorenz, Alexis Toumi, Giovanni de Felice, Konstantinos Meichanetzidis, Stephen Clark, and Bob Coecke. 2021. lambeq: An efficient high-level python library for quantum nlp. *arXiv e-prints*, pages arXiv–2110.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686, Melbourne, Australia. Association for Computational Linguistics.

Joachim Lambek. 1999. Type grammar revisited. In *Logical Aspects of Computational Linguistics: Second International Conference, LACL'97 Nancy, France, September 22-24, 1997 Selected Papers 2*, pages 1–27. Springer.

Joachim Lambek. 2008. *From Word to Sentence: a computational algebraic approach to grammar*. Polimetrica sas.

Zuchao Li, Kevin Parnow, and Hai Zhao. 2022. Incorporating rich syntax information in grammatical error correction. *Information Processing & Management*, 59(3):102891.

Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis, and Bob Coecke. 2023. Qnlp in practice: Running compositional models of meaning on a quantum computer. *Journal of Artificial Intelligence Research*, 76:1305–1342.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Weicheng Ma, Brian Wang, Hefan Zhang, Lili Wang, Rolando Coto-Solano, Saeed Hassanpour, and Soroush Vosoughi. 2023. Improving syntactic probing correctness and robustness with control tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 402–415.

David Mareček and Rudolf Rosa. 2019. From balustrades to pierre vinken: Looking for syntax in transformer self-attentions. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 263–275.

Mikko Möttönen, Juha J Vartiainen, Ville Bergholm, and Martti M Salomaa. 2005. Transformation of quantum states using uniformly controlled rotations. *Quantum Information & Computation*, 5(6):467–473.

Lee J O'Riordan, Myles Doyle, Fabio Baruffa, and Venkatesh Kannan. 2020. A hybrid classical-quantum workflow for natural language processing. *Machine Learning: Science and Technology*, 2(1):015011.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Rishab Sharma, Fuxiang Chen, Fatemeh Fard, and David Lo. 2022. An exploratory study on code attention in bert. In *Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension*, pages 437–448.

Shangshang Shi, Zhimin Wang, Jiaxin Li, Yanan Li, Ruimin Shang, Haiyong Zheng, Guoqiang Zhong, and Yongjian Gu. 2023. A natural nisq model of quantum self-attention mechanism. *arXiv e-prints*, pages arXiv–2305.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

James C Spall. 1998. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on aerospace and electronic systems*, 34(3):817–823.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. 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)*, pages 1415–1420.

Yue Zhang and Zhenghua Li. 2022. Csyngec: Incorporating constituent-based syntax for grammatical error correction with a tailored gec-oriented parser. *arXiv e-prints*, pages arXiv–2211.

Jianyu Zheng and Ying Liu. 2023. What does chinese bert learn about syntactic knowledge? *PeerJ Computer Science*, 9.

Elton Yechao Zhu, Sonika Johri, Dave Bacon, Mert Esencan, Jungsang Kim, Mark Muir, Nikhil Murgai, Jason Nguyen, Neal Pisenti, Adam Schouela, et al. 2022a. Generative quantum learning of joint probability distribution functions. *Physical Review Research*, 4(4):043092.

Fangyi Zhu, Lok You Tan, See-Kiong Ng, and Stéphane Bressan. 2022b. Syntax-informed question answering with heterogeneous graph transformer. In *Database and Expert Systems Applications: 33rd International Conference, DEXA 2022, Vienna, Austria, August 22–24, 2022, Proceedings, Part I*, pages 17–31. Springer.

A Methodology Details

A.1 Architecture Overview

Our architecture overview is shown in Figure 1.

A.2 DisCoWord Supplementary

Specifically, we follow DisCoCat to transform sentences into parameterised quantum circuits according to their string diagrams produced by the combinatory categorial grammar (CCG) parser, as shown in Figure 2 and Figure 3. In this case, the word is

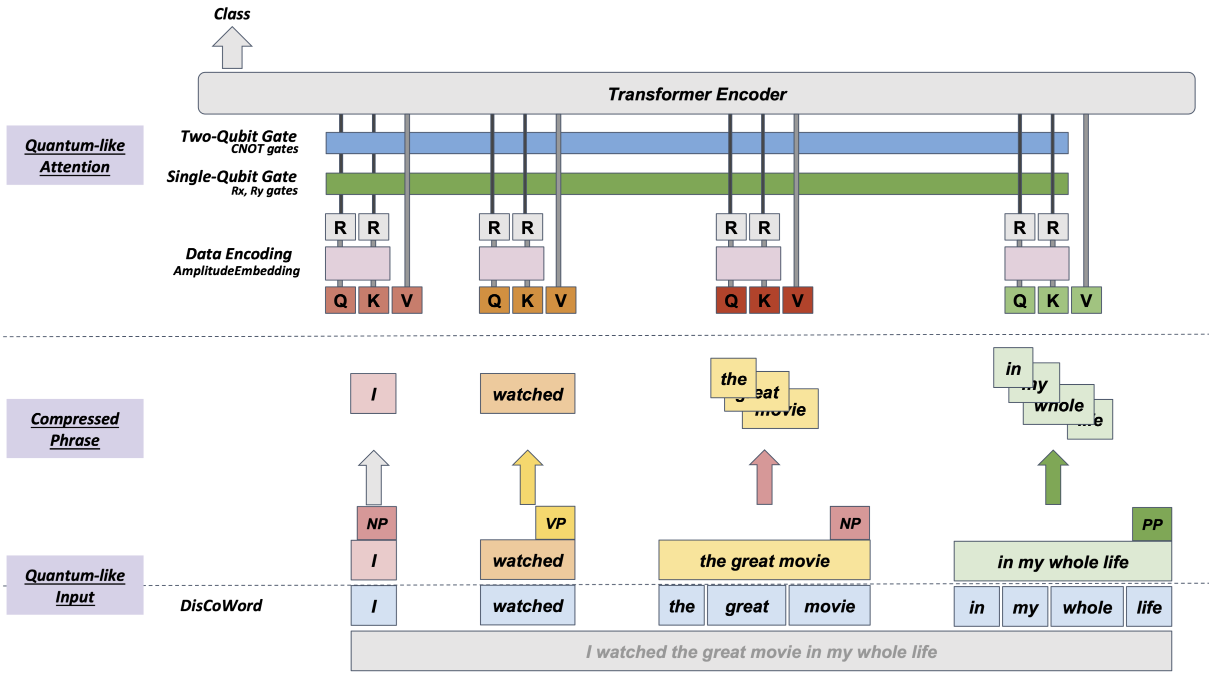
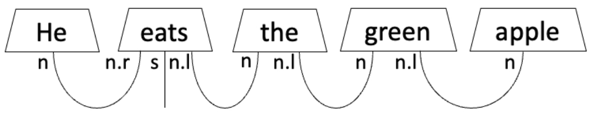
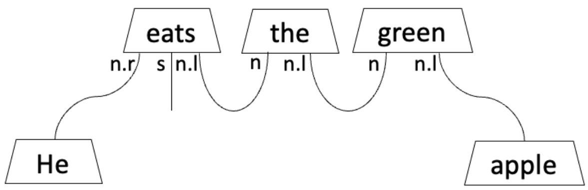


Figure 1: The overview of model architecture

represented by the quantum state of the qubits of its pregroup grammar, named word state. However, DisCoCat randomly initialises the quantum circuit parameters, which means the word states are not the semantics of these words. They can only represent the 'meaning' in the specific sentence classification task and can not be used in other NLP tasks. We use pretrained word embeddings to initialise these parameters so that the word states can hold both the syntactic and semantic information based on their contextual grammar structure and pretrained word embeddings. After training, we evaluate the word states. Note that the evaluated word states are complex vectors. To use them in later task classification experiments, we concatenate the real parts and image parts as word embeddings, named DisCoWord representation.



(a) String diagram before bending noun.



(b) String diagram after bending noun.

Figure 2: String diagrams of the sentence, where n, $n.r$, $n.l$, s are the grammar types (Lambek, 2008) of words, the types under a word form its pregroup grammar (Lambek, 1999).

B Experiments Details

B.1 Computational Resource Utilization

We use four A100 GPUs in our work. It takes 20-50 GPU hours to train DisCoWord representation. For training our Transformer variants, it takes 5-90 GPU hours to train the Transformer (QSA) and 1-10 GPU hours for other variants. The reason that training Transformer (QSA) takes more time is due to the use of PennyLane (Bergholm et al., 2018) for quantum simulation.

B.2 DisCoWord Training Details

Due to the hardware resource limitation, we split each dataset into subsets and train the DisCoCat with them separately. In addition, there are three processings on the word state: **a**) If it is longer than 256 dimensions, we drop the word state evaluations to save memory. If it is less than 256 dimensions, we convert the states into 256 dimensions and then conduct a zero-padding. **b**) If the specific pregroup grammar appears in the testing set but not in the training set, we take the mean of a word's states to deal with the case. For example, the word 'like' has three pregroup grammars representing its adjective,

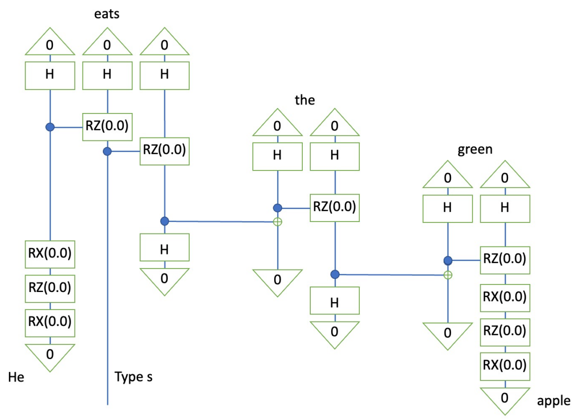


Figure 3: Quantum circuit of the sentence. The quantum state of the qubits belonging to a word represents its word state.

noun, and conjunction meaning respectively, but the conjunctive 'like' only appears in the testing set. We take the mean of the word states of the adjective 'like' and the nominal 'like' as the state of the conjunctive 'like'. **c**) Since the evaluated states are complex vectors, we concatenate the real-valued and image components for the Transformer.

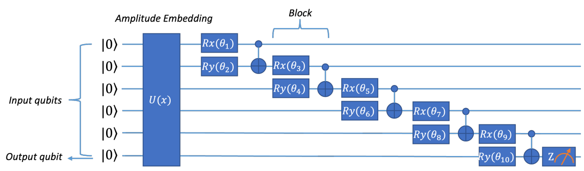


Figure 4: The quantum circuit example.

B.3 Quantum Self-Attention Design

An example of the quantum circuit for QSA is shown in Figure 4. In the Figure, x is the Q/K vector of a head, and $U(x)$ represents the AmplitudeEmbedding feature map. $\theta_1, \theta_2 ... \theta_{10}$ are the trainable parameters. In this circuit, the number of input qubits n is 6, which implies that the dimensionality 2^n of Q/K vector of a head is 64.

Method	MR	Tweet	SST-2	Offenseval
w/o-phrasing	21.35	15.27	19.6	23.57
ComPhE (B2T)	9.1	6.72	8.43	10.26
ComPhE (T2B)	6.43	5.54	6.06	7.04

Table 5: The statistics of input token length using different phrasing methods.

B.4 Statistics of Phrasing Methods

As aforementioned, phrasing methods will reduce the input token length. Here we list the statistics of the input token length using different phrasing methods in Table 5.