Quantum Information Processing (2022) 21:126
https://doi.org/10.1007/s11128-022-03441-9

Quantum computations for disambiguation and question answering

A. D. Correia[1,3] · **M. Moortgat**[2,3] · **H. T. C. Stoof**[1,3]

Received: 8 September 2021 / Accepted: 24 January 2022 / Published online: 30 March 2022

Abstract

Automatic text processing is now a mature discipline in computer science, and so attempts at advancements using quantum computation have emerged as the new frontier, often under the term of quantum natural language processing. The main challenges consist in finding the most adequate ways of encoding words and their interactions on a quantum computer, considering hardware constraints, as well as building algorithms that take advantage of quantum architectures, so as to show improvement on the performance of natural language tasks. In this paper, we introduce a new framework that starts from a grammar that can be interpreted by means of tensor contraction, to build word representations as quantum states that serve as input to a quantum algorithm. We start by introducing an operator measurement to contract the representations of words, resulting in the representation of larger fragments of text. We then go on to develop pipelines for the tasks of sentence meaning disambiguation and question answering that take advantage of quantum features. For the first task, we show that our contraction scheme deals with syntactically ambiguous phrases storing the various different meanings in quantum superposition, a solution not available on a classical setting. For the second task, we obtain a question representation that contains all possible answers in equal quantum superposition, and we implement Grover's quantum search algorithm to find the correct answer, agnostic to the specific question, an implementation with the potential of delivering a result with quadratic speedup.

Keywords Quantum natural language processing · Grover's algorithm · Quantum search · Question answering · Syntactic ambiguities

✉ A. D. Correia
a.duartecorreia@uu.nl

1 Institute for Theoretical Physics, Utrecht University, Princetonplein 5, 3584 CC Utrecht, The Netherlands

2 Utrecht Institute of Linguistics OTS, Utrecht University, Trans 10, 3512 JK Utrecht, The Netherlands

3 Center for Complex Systems Studies, Utrecht University, Leuvenlaan 4, 3584 CE Utrecht, The Netherlands

1 Introduction

Recent developments in quantum computation have given rise to new and exciting applications in the field of natural language processing (NLP). Pioneering work in this direction is the DisCoCat framework [1, 2], which introduces a compositional mapping between types and derivations of Lambek's type-logical grammars [3, 4] and a distributional semantics [5] based on vector spaces, linear maps, and tensor products. In this framework, the interpretations of large text fragments are obtained by performing a tensor contraction between the tensor interpretations of individual words. To interpret text fragments taking into account their grammatical features, while staying in the vector space semantics, the dimension of the representation quickly scales, as it depends on the complexity of the syntactic type, which has been a limiting feature in vector-based semantics implementations [6]. This motivates a representation of words as quantum states, counting on the potential of quantum computers to outperform the limitations of classical computation both in terms of memory use [7] and in terms of processing efficiency [8]. In this setting, words are represented as multipartite quantum states, with the theory predicting that, when contracted with one another, the meaning of larger text fragments is encoded in the resulting quantum states.

The challenge is now in implementing these contractions on quantum circuits. Circumventing this issue, DisCoCirc [9] introduces a different way of representing the meaning of a sentence, where certain words are seen as quantum gates that act as operators on input states representing other words. The DisCoCirc approach uses quantum machine learning algorithms [10] for NLP [11, 12] where circuit parameters, related to word representations, are then learned by classical optimization and used to predict different binary labels statistically, such as the answers to *yes-no* questions [13], topics of phrases, or the distinction between subject and object relative clauses [14].

Although these implementations can play an important role in speeding up NLP tasks based on current machine-learning ideas and techniques, they do not go beyond the current paradigm in terms of classification tasks. Furthermore, a number of theoretical advances using the tensor contractions from DisCoCat cannot be directly reproduced, since the mapping from a phrase to a circuit requires extra steps that deviate from the original grammatical foundation, not treating every word as an input at the same level. We refer here to the work done in expanding the toolbox of word representations with density matrices [15], so as to achieve good results on discerning different word and phrase senses [16–18], and in entertaining simultaneously different possible interpretations of texts, either by looking at an incremental interpretation of the parsing process [19], or by considering a single representation for the multiple readings of syntactic ambiguities [20, 21]. This presents a strong incentive to find an alternative quantum circuit implementation that sticks to the original grammatical formulation, preserving the previous achievements, where all words are taken as input on an equal footing. In addition, it is our belief that a quantum framework can contribute a great deal to the reestablishment of rule-based NLP, as a desirable alternative to large-scale statistical approaches [22], since certain computations become more efficient if we use the appropriate quantum algorithms, as we will illustrate in the case of question answering where quadratic quantum speedup can be achieved.

The paper is structured as follows. In Sect. 2, we develop the grammatical framework and quantum state interpretation thereof, setting the stage for the types of linguistic problems we will deal with here. Here, we introduce the idea that words are represented as vectors, matrices, and higher-rank tensors, depending on their grammatical function, that contract with each other following grammatical rules, explaining how we can arrive at the interpretations of larger fragments of text. In Sect. 3, we put forward an approach where the words are interpreted as quantum states, and we show how the contractions between word representations can be implemented on a quantum computer as the measurement of a permutation operator. We elaborate on how this setting permits the simultaneous treatment of ambiguous phrases in English. In Sect. 4, we apply Grover's algorithm to question answering, using the framework developed in the previous section to turn the representation of the question and answers into the input of the algorithm, together with an oracle that identifies that correct answers. Finally, in Sect. 5 we give an overview of the developments introduced and discuss further work.

2 Syntax–semantics interface

In this section, we introduce the grammatical framework that we will be working with. It consists of a categorial grammar as the syntactic front end, together with a compositional mapping that sends the types and derivations of the syntax to a vector-based distributional interpretation. This is necessary to understand the type of linguistic problems that we can address and how they can be solved using a quantum circuit.

2.1 Type logic as syntax

The key idea of categorial grammar formalisms is to replace the parts of speech of traditional grammars (nouns, adjectives, (in)transitive verbs, etc.) by logical formulas or types; a deductive system for these type formulas then determines their valid combinations. The idea can be traced back to Ajdukiewicz [23], but Lambek's syntactic calculus [3] is the first full-fledged formulation of a categorial-type logic that provides an algorithm to effectively decide whether a phrase is syntactically well formed or not.

Let us briefly discuss types and their combinatorics. We start from a small set of primitive types, for example s for declarative sentences, n for noun phrases, w for open-ended interrogative sentences, etc. From these primitive types, compound types are then built with the aid of three operations: multiplication $\bullet$, left division $\backslash$, and right division $/$. Intuitively, a type $A \bullet B$ stands for the concatenation of a phrase of type A and a phrase of type B ("A and then B"). Concatenation is not commutative ("A and then B" $\neq$ "B and then A"). Hence, we have left *vs* right division matching the multiplication: $A \backslash B$ can be read as "give me a phrase A to the left, and I'll return a phrase B"; B/A is to be interpreted as "give me a phrase A to the right, and I'll return a phrase B." We can codify this informal interpretation in the rules below, where $A_1 \bullet \cdots \bullet A_n \vdash B$ means that from the concatenation of phrases of type $A_1, \ldots, A_n$ one can derive a phrase of type B. Hence,

$$B/A \bullet A \vdash B, \tag{1}$$
$$A \bullet A\backslash B \vdash B. \tag{2}$$

As examples of simple declarative sentences, consider *Alice talks*, or *Bob listens*. In the former case, we assign the type n to *Alice* and the type $n\backslash s$ to the intransitive verb *talks*. We start by multiplying the word types in the order by which the words appear, forming $n \bullet n\backslash s$. Then, it suffices to apply rule (2), with $A = n$ and $B = s$, to show that $n \bullet n\backslash s$ derives s, i.e., constitutes a well-formed sentence. Conversely, the lack of a derivation of s from $n\backslash s \bullet n$ (*talks Alice*) allows us to conclude that this is not a well-formed sentence. These, and the later examples, illustrate only the simplest ways of combining types, but these will suffice for the purposes of this paper. To obtain a deductive system that is sound and complete with respect to the intended interpretation of the type-forming operations, Lambek's syntactic calculus also includes rules that allow one to infer $A \vdash C/B$ and $B \vdash A\backslash C$ from $A \bullet B \vdash C$. Moreover, to deal with linguistic phenomena that go beyond simple concatenation, Lambek's type logic has been extended in a number of ways that keep the basic mathematical structure intact but provide extra type-forming operations for a finer control over the process of grammatical composition. See Ref. [24] for a survey and Ref. [21] for a quantum interpretation of such structural control operations.

2.1.1 Syntactic ambiguities

To see rule (1) in action, consider adjectives in English. An adjective is expecting a noun to its right, and, once it is composed with a noun, it must derive something that can be used, for instance, as the argument of an intransitive verb, which, as we have seen, is of type n. Thus, an adjective must be of type n/n, and we can use rule (1) to prove that, as an example, *rigorous mathematicians* is a well-formed phrase of type n.

For certain phrases, there is more than one way of deriving the target type, with each derivation corresponding to a distinct interpretation. As an example, consider the noun phrase *rigorous mathematicians and physicists*, an ambiguous structure that has already been studied in the context of vector representations in Ref. [20]. Here, the conjunction *and* gets the type $(n\backslash n)/n$; for the complete phrase, we want to show that the following judgment holds:

$$n/n \bullet n \bullet (n\backslash n)/n \bullet n \vdash n. \tag{3}$$

There are two possible interpretations: a first one, where the adjective *rigorous* has scope over *mathematicians and physicists*, and a second one, where it only has scope over *mathematicians*. Each of these interpretations is connected to a different way of deriving the goal formula n. The first reading is obtained by applying the rules in the following order:

$$\underbrace{\underbrace{n/n \bullet n}_{(1)\vdash n} \bullet \underbrace{\underbrace{(n\backslash n)/n \bullet n}_{(1)\vdash n\backslash n}}_{(2)\vdash n}}_{(1)\vdash n}, \tag{4}$$

while for the second reading the rules apply in a different order as

$$\underbrace{\underbrace{n/n \bullet n}_{(1)\vdash n} \bullet \underbrace{(n\backslash n)/n \bullet n}_{(1)\vdash n\backslash n}}_{(2)\vdash n}. \tag{5}$$

Our goal is to treat both readings simultaneously until further information allows us to clarify which of the readings is the intended one.

2.1.2 Question answering

Question answering (Q&A) is one of the most common tasks in NLP [25]. Questions can be close-ended, having "yes" or "no" for an answer, or open-ended, starting by "who," "why," or "what," also referred to as *wh*-questions. For P possible answers, it is always possible to turn *wh*-questions into close-ended questions. If we know that either Alice, Bob, Carol, or Dave is talking, we can turn "Who talks?" into a series of four questions "Does `[name]` talk?". Thus, for P possible answers, there are P closed-ended questions that we need to check.[1] We would like to find the answer to the open-ended questions directly, without this mapping. Syntactically, *wh*-questions are open-ended interrogative sentences, and as such are assigned their own type w. For a subject question, the type of the word *who* is thus $w/(n\backslash s)$, since, when applied to an intransitive verb using rule (2), it derives the interrogative type w.

2.2 Vectors as semantics

In the context of automatic processing of text, the most widely used form of representing a word is by a unique array of values, referred to as a "word embedding." Seen as vectors, we can cluster or compare them using varied geometric tools [26–28]. Representing the meanings of words as such is widely known as "distributional semantics" [29]. In earlier work, vector entries were related to how often a word would appear next to other words [30], following the "distributional hypothesis" that states that words that appear in similar contexts are themselves similar [31]. Nowadays, word embeddings are extracted using language models, targeted on the prediction of the most likely next

[1] This is a common way of turning Q&A into a classification problem, where each close-ended question gets a binary label, depending on whether the answer is true or false. Binary classification problems are some of the most well-established applications of machine learning. After finding a way of representing the question statements, usually as single vectors, a number of these labeled statements is used to predict the labels of the holdout statements.

word [32, 33]. This presents a problem for the representation of larger fragments, since they are less likely to appear in a text, making their distributional array rather sparse and thus not particularly meaningful. Larger fragments can nevertheless receive an embedding, but a direct connection with grammatical composition is lost.

To tackle this problem, the authors in Ref. [1] propose that the arrays representing different words depend on their syntactic types, namely having a dimensionality that mirrors their type complexity. This introduces a way of composing the meanings of the individual words that is homomorphic to the syntactic derivations, generating a representation of larger fragments from the representation of smaller ones. For completeness, the mapping between the syntax and the semantics is done using the formalism of vector spaces. Each syntactic type A is mapped to its semantic type via $\lceil A \rceil$. Each semantic type is then interpreted as a vector space, where the particular words are represented. Let there be three basic semantic spaces $\{S, N, I\}$. The simple syntactic types n and s are mapped, respectively, to $\lceil n \rceil = N$ and $\lceil s \rceil = S$. Each individual word is an element of the semantic space that interprets its syntactic type. For instance, the interpretation of the word *physicists* is now seen as a vector in N, this being the vector space where the distributional information of nouns is stored. Similarly, *Alice talks* is represented by a vector in S, that has as basis elements two orthogonal states corresponding to "true" and "false." The interrogative type w is mapped to $\lceil w \rceil = I \otimes N \otimes I \otimes S$. The vector space I ("index") has basis elements that are in one-to-one correspondence to the nouns that can be used as answers to the interrogative sentence, providing an enumeration of the noun vectors of N. This will be useful later when we need to index the quantum states associated with each possible answer.

The vector spaces that translate the directional and multiplicative types are obtained recursively as

$$\lceil A \backslash B \rceil = \lceil A / B \rceil = \lceil A \bullet B \rceil = \lceil A \rceil \otimes \lceil B \rceil, \tag{6}$$

where $\otimes$ forms a tensor product space, inductively starting from $A, B, C \in \{n, s, w\}$. Note that the tensor is commutative, such that $\lceil A \rceil \otimes \lceil B \rceil \cong \lceil B \rceil \otimes \lceil A \rceil$. We perform tensor contractions as the interpretations of the rules in Eqs. (1) and (2). Thus, an intransitive verb is represented as a matrix in $N \otimes S$, that when acting on a vector of type N returns a vector in S. Using the notation $[\![.]\!]$ to represent the tensor interpretation of a word, and assuming an orthogonal basis $\{\hat{n}_i\}$ of N and an orthogonal basis $\{\hat{s}_i\}$ of S, the composition of the vectorial interpretations of *Alice* and *talks* leads to the interpretation of the entire sentence as a vector in S. The word meanings for this sentence are represented as

$$[\![\text{Alice}]\!] = \sum_p A_p \, \hat{n}_p \tag{7}$$

$$[\![\text{talks}]\!] = \sum_{qr} t_{qr} \, \hat{n}_q \otimes \hat{s}_r, \tag{8}$$

and the full sentence meaning as

$$[\![\text{Alice talks}]\!] = [\![\text{Alice}]\!] \cdot [\![\text{talks}]\!] = \sum_{pr} A_p t_{pr} \hat{s}_r. \tag{9}$$

A more refined treatment of the translation from the Lambek types to tensor spaces has been given in Ref. [20].

Similarly, the semantic space for an adjective can be seen as a matrix in $N \otimes N$. Note that here the analogy between a matrix modifying a vector and the adjective as a noun modifier is the clearest. Let us look at the meanings of *rigorous* and *mathematicians*, which can be represented as

$$[\![\text{rigorous}]\!] = \sum_{ij} r_{ij}\ \hat{n}_i \otimes \hat{n}_j \tag{10}$$

$$[\![\text{mathematicians}]\!] = \sum_{k} m_k\ \hat{n}_k. \tag{11}$$

The meaning of *rigorous mathematicians* will be given by the application of the translation of rule (1) to tensors. At the components level, it is the matrix multiplication between the *rigorous* matrix and the *mathematicians* vector, which gives, consistently with n being the syntactic type of this fragment, a vector in N, as

$$\begin{aligned}&[\![\text{rigorous mathematicians}]\!]\\ &= [\![\text{rigorous}]\!].[\![\text{mathematicians}]\!] = \sum_{ij} r_{ij}\, m_j\ \hat{n}_i.\end{aligned} \tag{12}$$

The different order of application of Lambek rules in Eqs. (4) and (5) translates into different vectors that represent the two readings of *rigorous mathematicians and physicists*. The words *and* and *physicists* are given the vector representations

$$[\![\text{and}]\!] = \sum_{lmn} a_{lmn}\ \hat{n}_l \otimes \hat{n}_m \otimes \hat{n}_n, \tag{13}$$

$$[\![\text{physicists}]\!] = \sum_{o} p_o\ \hat{n}_o. \tag{14}$$

The reading from Eq. (4) is represented by the vector

$$[\![\text{rigorous mathematicians and physicists}]\!]_1 = \sum_{ijln} r_{ij}\, m_l\, a_{ljn}\, p_n\ \hat{n}_i, \tag{15}$$

whereas the reading from Eq. (5) is encoded in the vector

$$[\![\text{rigorous mathematicians and physicists}]\!]_2 = \sum_{jlmn} r_{lj}\, m_j\, a_{lmn}\, p_n\ \hat{n}_m, \tag{16}$$

which are of the same form as the results in Ref. [20].

For interrogative sentences, the word *who* will have the semantic function of "lifting" an intransitive verb with representation in space $N \otimes S$ to a representation in $I \otimes N \otimes I \otimes S$, since

$$\begin{aligned} \ulcorner w/(n\backslash s)\urcorner &= I \otimes N \otimes I \otimes S \otimes N \otimes S \\ &\cong I \otimes N \otimes I \otimes S \otimes S \otimes N. \end{aligned} \tag{17}$$

An element of this space contracts with an element of the representation space of intransitive verbs, $N \otimes S$, associating the index of every possible answer, in I, with both its representation in N and its truth value in S.

3 Implementation

In this section, we motivate a passage from vectors to quantum states and we introduce them as inputs of quantum circuits that calculate contractions between word representations.

3.1 Quantum states as inputs of a quantum circuit

We now switch to a representation of word embeddings as vectors in complex-valued inner product vector spaces, i.e., Hilbert spaces. Our atomic semantic spaces N, S, and I will now be replaced by their quantum counterparts as the interpretation spaces. We thus have the Hilbert spaces $\mathcal{H}^N$, $\mathcal{H}^S$, and $\mathcal{H}^{\otimes p}$, respectively, with $\mathcal{H}^{\otimes p}$ the p-qubit Hilbert space corresponding to the complex-valued realization of the semantic type I, where we assume that $P = 2^p$. For instance, with $\{|n_i\rangle\}$ the basis of $\mathcal{H}^N$, we now have

$$[\![\text{Alice}]\!] = |\text{Alice}\rangle = \sum_p A_p \, |n_p\rangle. \tag{18}$$

Note that this space allows us to expand our representations with complex-valued entries, and a proper contraction between the words will require the conjugation of some of the components, i.e.,

$$[\![\text{Alice}]\!]^* = \langle\text{Alice}| = \sum_p A_p^* \, \langle n_p|. \tag{19}$$

Let the input of a circuit be the product of the states that interpret each word in the language fragment in question. Our running example of a noun subject and an intransitive verb *Alice talks* is now represented as the input

$$|\text{Alice}\rangle|\text{talks}\rangle \in \mathcal{H}^N \otimes \mathcal{H}^N \otimes \mathcal{H}^S.$$

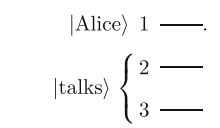


Fig. 1 Quantum circuit with intransitive sentence input

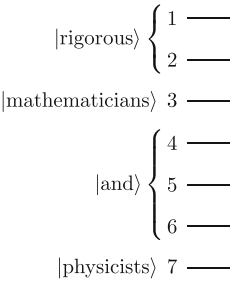


Fig. 2 Quantum circuit with syntactically ambiguous input

The basis of $\mathcal{H}^S$, $\{|s_i\rangle\}$, is the single-qubit spin states $|0\rangle$ and $|1\rangle$, where the former represents a sentence that is false and the latter one that is true. In this setting, it is also possible to establish a probability distribution over the truthfulness of a sentence.

Each of the elements of the interpreting spaces will be represented by a labeled quantum wire, thus rewriting the input state as

$$|\text{Alice}\rangle|\text{talks}\rangle \in \mathcal{H}_N^1 \otimes \mathcal{H}_N^2 \otimes \mathcal{H}_S^3, \tag{20}$$

used as the input of a quantum circuit, as shown in Fig. 1.
The ambiguous fragment *rigorous mathematicians and physicists* will be initially represented as a unique state in the tensor space, formed by numbered copies of the $\mathcal{H}^N$ space as

$$\begin{aligned}&|\text{rigorous}\rangle|\text{physicists}\rangle|\text{and}\rangle|\text{mathematicians}\rangle\\ &\in \mathcal{H}_1^N \otimes \mathcal{H}_2^N \otimes \mathcal{H}_3^N \otimes \mathcal{H}_4^N \otimes \mathcal{H}_5^N \otimes \mathcal{H}_6^N \otimes \mathcal{H}_7^N,\end{aligned}$$

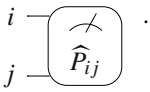
forming the input of a quantum circuit as in Fig. 2.
From here, different contractions, corresponding to the two possible readings, will be represented by different circuits acting on this same input, as we show in detail below.

3.2 Contraction as measurement of permutation operator

To compute the desired contraction using the quantum circuit, we calculate the expectation value of the permutation operator $\widehat{P}_{ij}$ on the input states/wires indexed by i, j. These correspond to the spaces with elements that we want to contract, following the syntactic rules. For two states $|\phi_1\rangle_i$ and $|\phi_2\rangle_j$ belonging to two numbered copies of a Hilbert space, respectively, $\mathcal{H}_i^{\lceil A \rceil}$ and $\mathcal{H}_j^{\lceil A \rceil}$, we refer to the following quantity as the *measurement of the expectation value of the permutation operator*:

$$\begin{aligned}\langle\phi_1|_i\langle\phi_2|_j\widehat{P}_{ij}|\phi_1\rangle_i|\phi_2\rangle_j &= \\ &= \langle\phi_1\rangle\phi_{2i}\langle\phi_2\rangle\phi_{1j} \equiv |\langle\phi_1\rangle\phi_2|^2 .\end{aligned} \tag{21}$$

In general, to obtain this quantity on a quantum circuit, one must perform repeated measurements of the input states $|\phi_1\rangle_i$ and $|\phi_2\rangle_j$, on a basis that diagonalizes the permutation operator, summing the frequency of outcomes, using the respective operator's eigenvalues as weights. We introduce the following circuit notation to indicate the measurement of the permutation operator:



If the measuring device is only capable of performing measurements in the standard basis, then we must additionally apply to the input states the inverse of the transformation that diagonalizes the permutation operator, before performing the repeated measurements. In Appendix 1, we show how this can be achieved using the inverse transformation to the Bell basis in the case of two-qubit inputs. In this case, the measurement of the expectation value can be understood as the map between the SWAP operator that represents the permutation operator in that case, and the projection operator on the maximally entangled state $|\beta_{00}\rangle$, which, although not a homomorphism, will be diagonal in the same basis, since both operators share an algebra.

We now show in two ways that the final representation of a simple sentence such as *Alice talks* is stored as an effective state $|\psi\rangle$ in $\mathcal{H}_3^S$, after measuring $\widehat{P}_{12}$ not normalizing for clarity, with input as given in Eq. (20).

Using operators Assume that an operator $\widehat{O}_3$ is being measured in space $\mathcal{H}_3^S$. Its expectation value is given by $\langle\psi|\widehat{O}_3|\psi\rangle$, after measuring $\widehat{P}_{12}$, with

$$\begin{aligned}&\langle\text{Alice}|\langle\text{talks}|\widehat{P}_{12}\otimes\widehat{O}_3|\text{Alice}\rangle|\text{talks}\rangle \\ &\quad\equiv \langle\psi|\widehat{O}_3|\psi\rangle.\end{aligned} \tag{22}$$

The left-hand side unfolds as follows:

$$\begin{aligned}&\langle\text{Alice}|\langle\text{talks}|\ \widehat{P}_{12}\otimes\widehat{O}_3\ |\text{Alice}\rangle|\text{talks}\rangle = \\ &= \sum_{pqr,p'q'r'} A_p^* t_{qr}^* \langle n_p n_q s_r|\ \widehat{P}_{12}\otimes\widehat{O}_3 \\ &= A_{p'}t_{q'r'}|n_{p'}n_{q'}s_{r'}\rangle \\ &= \sum_{pqr,p'q'r'} A_p^* t_{qr}^* \langle n_p n_q s_r|\ \widehat{O}_3\ A_{p'}t_{q'r'}|n_{q'}n_{p'}s_{r'}\rangle \\ &= \sum_{pqr,p'q'r'} A_p^* t_{qr}^* \langle s_r|\ \widehat{O}_3\ A_{p'}t_{q'r'}|s_{r'}\rangle\delta_{pq'}\delta_{qp'} \\ &= \sum_{qr,q'r'} A_q t_{qr}^* \langle s_r|\ \widehat{O}_3\ A_{q'}^* t_{q'r'}|s_{r'}\rangle.\end{aligned} \tag{23}$$

To uniquely determine $|\psi\rangle$, we need to solve $m = d(d-1)/2$ independent equations, where d is the dimension of $\mathcal{H}^3$ of the form below:

$$\sum_{qr,q'r'} A_q t^*_{qr} \langle s_r| \ \widehat{O}_3 \ A^*_{q'} t_{q'r'} |s_{r'}\rangle = \langle\psi|\widehat{O}_3|\psi\rangle. \tag{24}$$

Any operator $\widehat{O}_3$ can be decomposed as a sum of m linearly independent operators $\widehat{O}^a_3$, with $1 < a < m$. Since Eq. (24) holds for any operator, it holds for each $\widehat{O}^a_3$, thus generating m independent equations, necessary and sufficient to solve for $|\psi\rangle$. In particular, if $|\psi\rangle$ is expressed in the basis $|s_{r'}\rangle$, the components of $|\psi\rangle$ are given precisely by the respective components of the left-hand side of Eq. (24), from which we can immediately conclude that the effective state in $\mathcal{H}^3_S$ is

$$|\psi\rangle = \sum_{q'r'} A^*_{q'} t_{q'r'} |s_{r'}\rangle \equiv [\![\text{Alice talks}]\!]. \quad \blacksquare \tag{25}$$

Similarly, density matrices can be used to confirm not only that the state in $\mathcal{H}^S_3$ corresponds to Eq. (25) after the measurement, using partial tracing, but also that the outcome of this operation is a pure state.

Using density matrices Assume that the sentence *Alice talks* is being represented by the pure state density matrix

$$\widehat{\rho} = |\text{Alice}\rangle|\text{talks}\rangle\langle\text{Alice}|\langle\text{talks}|. \tag{26}$$

We want to show that the density matrix $\widehat{\rho}_3$ that we obtain in space $\mathcal{H}^S_3$ after the measurement of $\widehat{P}_{12}$ is in fact a pure state. We do that by taking the partial trace in spaces 1 and 2 of $\widehat{P}_{12}\widehat{\rho}$:

$$\begin{aligned}
\widehat{\rho}_3 &= \text{Tr}_{12}\left(\widehat{P}_{12}\widehat{\rho}\right) = \\
&= \sum_{ab} \sum_{pqr,p'q'r'} \langle n_a n_b|\widehat{P}_{12} A_{p'} t_{q'r'} |n_{p'} n_{q'} s_{r'}\rangle \langle n_p n_q s_r| A^*_p t^*_{qr} |n_a n_b\rangle \\
&= \sum_{ab} \sum_{pqr,p'q'r'} \langle n_a n_b| A_{p'} t_{q'r'} |n_{q'} n_{p'} s_{r'}\rangle \langle n_p n_q s_r| A^*_p t^*_{qr} |n_a n_b\rangle \\
&= \sum_{r,p'q'r'} A_{p'} t_{q'r'} |s_{r'}\rangle\langle s_r| A^*_{q'} t^*_{p'r} \\
&= \sum_{q'r'} A^*_{q'} t_{q'r'} |s_{r'}\rangle \sum_{p'r} \langle s_r| A_{p'} t^*_{p'r} = |\psi\rangle\langle\psi|.
\end{aligned} \tag{27}$$

This thus proves that the resulting state in space $\mathcal{H}^S_3$ is pure and equal to Eq. 25. It also proves that $\langle\psi\rangle = \langle\widehat{P}_{12}\rangle$, as expected from $\langle\psi|\widehat{O}|\psi\rangle = \langle\widehat{O}\rangle\langle\widehat{P}_{12}\rangle$. $\blacksquare$

Note that here the index contraction is equivalent to that of Eq. (9), enhanced with the conjugation of some components, which remain as an informative feature from the directionality of language, which would be lost otherwise. The circuit that calculates Eq. (25) is shown in Fig. 3.

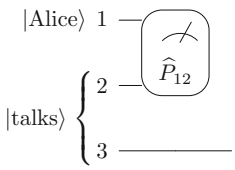


Fig. 3 Quantum circuit that measures the permutation operator $\widehat{P}_{12}$ on an intransitive sentence input

An alternative way of contracting the two-qubit spaces has been proposed in Ref. [12], where the Bell effect $\langle\beta_{00}| \equiv (\langle 00| + \langle 11|)/\sqrt{2} \in \mathcal{H}_N^1 \otimes \mathcal{H}_N^2$ is measured instead as

$$\begin{aligned}[\![\text{Alice talks}]\!] &= \langle\beta_{00}||\text{Alice}\rangle|\text{talks}\rangle \\ &= \frac{1}{\sqrt{2}} \sum_{p'q'r'} (\langle 00| + \langle 11|)\, A_{p'} t_{q'r'} |n_{p'} n_{q'} s_{r'}\rangle \\ &= \frac{1}{\sqrt{2}} \sum_{r'} (A_0 t_{0r'} + A_1 t_{1r'})\, |s_{r'}\rangle. \end{aligned} \tag{28}$$

Measuring the permutation operator as we do in Eq. (23) is manifestly a more general way of contracting the representations of words than what is done in Eq. (28). On the one hand, it allows each interpretation space to have more than two basis states, that is, each quantum wire can represent something more general than one qubit. On the other hand, it accommodates correctly the existence of complex numbers in the quantum mechanical representations. Importantly, it has also one more important feature that we will make use of now: It allows us to integrate a quantum superposition of conflicting readings.

3.3 Ambiguous readings on a quantum circuit

The quantum states of the two readings in Eqs. (15) and (16), that result from contracting the individual word states, can be written as

$$[\![\text{rigorous mathematicians and physicists}]\!]_1 = \sum_{ijln} r_{ij}\, m_l\, a^*_{ljn}\, p^*_n\, |n_i\rangle \tag{29}$$

and

$$[\![\text{rigorous mathematicians and physicists}]\!]_2 = \sum_{jlmn} r^*_{lj}\, m_j\, a_{lmn}\, p^*_n\, |n_m\rangle. \tag{30}$$

These can be represented by the two different circuits in Figs. 4 and 5, respectively, coming from the two different contraction schemes, as obtained in Ref. [20]. Also in this reference, an analysis of how to express the two readings syntactic ambiguities simultaneously is developed, which we here implement. We can go from the first

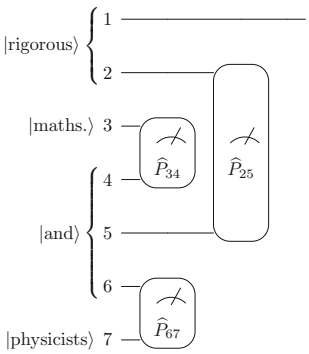


Fig. 4 Quantum circuit for the first interpretation of the syntactically ambiguous phrase

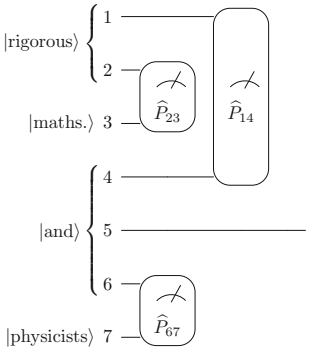


Fig. 5 Quantum circuit for the second reading of the syntactically ambiguous phrase

reading to the second by applying two wire swappings. First, we swap wires 3 and 5 on the second reading, which turns the measurement of $\widehat{P}_{23}$ into the measurement of $\widehat{P}_{25}$. Next, by swapping wires 5 (which now contains the information from wire 3) and 1, we effectively turn the measurement of $\widehat{P}_{14}$ into the measurement of $\widehat{P}_{34}$. In this way, the circuit in Fig. 6 is equivalent to that of the first reading. If we control the application of this set of swap gates on an extra qubit $|c\rangle = c_1|1\rangle + c_2|0\rangle$, we entangle the states of this qubit with the two possible readings. The first reading is stored in the quantum wire 5 with probability $|c_1|^2$, while the second reading is stored in that same quantum wire with probability $|c_2|^2$. In total, we have what is represented in the circuit of Fig. 7.

The innovation that this implementation brings is that we are now able to deal with both interpretations simultaneously, that later contractions, with the representations of other words, have the potential to disambiguate.

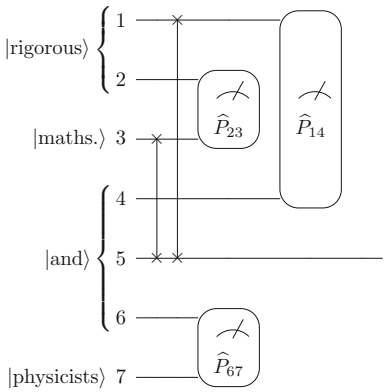


Fig. 6 Quantum circuit that computes the first reading from the contractions of the second and is therefore equivalent to Fig. 4

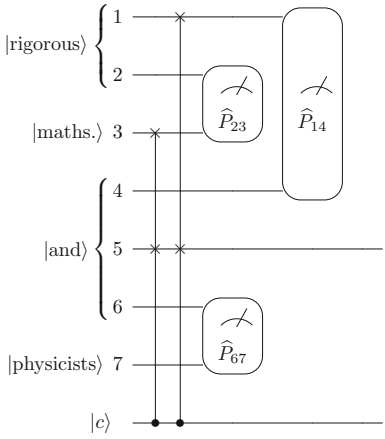


Fig. 7 Quantum circuit that computes simultaneously the two readings from the ambiguous input

4 Application

In this section, we apply Grover's quantum search algorithm to obtain the answer to a *wh*-question with quantum speedup, using the quantum circuits for sentence representation developed in the previous section.

4.1 Grover's quantum search algorithm

Grover's quantum search algorithm aims at finding the correct answer to a query, by taking a state with an equal superposition of orthogonal states representing the answers as the input, and outputting a state in which the only basis states that have any probability of being measured correspond to correct answers. For $P = 2^p$ possible solutions, the first step is to generate a linear superposition of P unique states, with its index x corresponding to one of the possible solutions. In the original proposal [34],

this input state is obtained by acting on $|0\rangle^{\otimes p}$ qubit states with the $H^{\otimes p}$ gate, where H is the one-qubit Hadamard gate, which generates

$$|\Psi\rangle = \frac{1}{\sqrt{P}} \sum_{x=0}^{P-1} |x\rangle. \tag{31}$$

Then, a sequence of gates, the *Grover iteration*, is repeatedly applied to this input state, until a correct answer is the guaranteed outcome of the measurement of the initial qubits. For Q correct solutions, only $O(\sqrt{P/Q})$ iterations are necessary, representing a quadratic speedup compared to a classical search algorithm, which requires checking all P possible answers. Each Grover iteration G has two main components: first, an *oracle* operation O, and then an *inversion about the mean* operation, formed by applying the unitary transformation that generates $|\Psi\rangle$ to $2|0\rangle\langle 0| - 1$, in this case

$$H^{\otimes p}(2|0\rangle\langle 0| - 1)H^{\otimes p} \equiv 2|\Psi\rangle\langle\Psi| - 1, \tag{32}$$

which can easily be shown to be unitary. The heart of the algorithm is the oracle, as it is able to distinguish the answers that are correct from those that are not. It is a unitary operation that works by flipping the sign of the answer state if $|x\rangle$ is correct, that is,

$$\begin{cases} O(|x\rangle) = -|x\rangle \text{ if} |x\rangle \text{is a correct answer,} \\ O(|x\rangle) = |x\rangle \qquad\quad \text{otherwise.} \end{cases}$$

To achieve this, more qubits might be necessary, and those constitute the "oracle workspace." In the original setup, it is the oracle that depends on the query at hand, as well as the form of the inputs, while the inverse operation has a universal form. By representing our *wh*-question query as quantum states and contractions therein, we will see that we can use Grover's algorithm with the problem dependence of its parts reversed: Instead, it is the oracle that is universal, and the rotation is query-dependent, obtained from a unitary transformation on $|0\rangle$.

4.2 Input-state preparation for question answering

The question statement and possible answers hold the key for the search algorithm to identify the correct answers in our application. This will happen as a consequence of the contractions of the possible solutions with the question predicate. We will use our previous construction as the input of the first Grover iteration. To this end, suppose that we want to know the answer to the question *Who talks?* and that we have P possible answers, of which Q are absolutely correct and $P - Q$ are, on the contrary, definitely wrong. For the oracle to identify the correct answers, they must be produced from the contraction with the verb and they must be in a superposition equivalent to Eq. (31).

The more complex mapping of w to the semantics, when compared with the syntactic types s and n, can be attributed to the particular semantics of questions and our application. In standard terms, the meaning of a question is taken as the map that sends answers, which belong to the interpretation space of nouns, to truth values, which are

elements of the interpretation space of declarative sentences. We want to keep track of which word provides a correct answer in our quantum circuit, and a map like the latter, upon performing contractions, would only give us a count of how many correct and wrong answers there are. To see this, suppose that the word *who* is represented in the space

$$\lceil w/(n\backslash s)\rceil = \mathcal{H}_1^N \otimes \mathcal{H}_2^S \otimes \mathcal{H}_3^S \otimes \mathcal{H}_4^N,$$

and semantic representation as

$$\begin{aligned} |\text{who}\rangle &= \sum_{ab,ij,kl} |n_i\rangle_1 |s_j\rangle_2 |s_k\rangle_3 |n_l\rangle_4 \delta_{il}\delta_{jk} \\ &= \sum_{i,l} |n_i\rangle_1 |s_j\rangle_2 |s_j\rangle_3 |n_i\rangle_4. \end{aligned} \tag{33}$$

and the intransitive verb *talks*

$$|\text{talks}\rangle = \sum_{mn} t_{mn}\ |n_m\rangle_5 |s_n\rangle_6. \tag{34}$$

Following the contraction schemes presented in Sect. 3.2, the contraction between these two words states results in the state

$$|\text{who}\rangle|\text{talks}\rangle = \sum_{ij} t_{ij}\ |n_i\rangle_1 |s_j\rangle_2, \tag{35}$$

and representing the answers as

$$|\text{answers}\rangle = \sum_{p} W_p |n_p\rangle_7, \tag{36}$$

their final contraction results in

$$|\text{who}\rangle|\text{talks}\rangle|\text{answers}\rangle = \sum_{ij} W_i^* t_{ij}^*\ |s_j\rangle_2. \tag{37}$$

This shows that a contraction just on the S and N spaces gives only a count of how many correct or incorrect answers there are, but not which ones are which.

As such, the map of the *wh*-word needs to be furthermore tensored with elements of $\mathcal{H}^{\otimes p}$, of which each of the basis elements corresponds to the unique indexing of the possible answers. This provides an entanglement between the distributional representation of a noun, its corresponding truth value and an enumerable representation in the quantum circuit. The word *who* thus belongs to the following semantic space, in the image of Eq. (17),

$$\lceil w/(n\backslash s)\rceil = \mathcal{H}_1^{\otimes p} \otimes \mathcal{H}_2^N \otimes \mathcal{H}_3^{\otimes p} \otimes \mathcal{H}_4^S \otimes \mathcal{H}_5^S \otimes \mathcal{H}_6^N,$$

with semantic representation given as

$$\begin{aligned} |\text{who}\rangle &= \sum_{ab,ij,kl} |a\rangle_1 |n_i\rangle_2 |b\rangle_3 |s_j\rangle_4 |s_k\rangle_5 |n_l\rangle_6 \delta_{ab}\delta_{il}\delta_{jk} \\ &= \sum_{a,i,l} |a\rangle_1 |n_i\rangle_2 |a\rangle_3 |s_j\rangle_4 |s_j\rangle_5 |n_i\rangle_6. \end{aligned} \tag{38}$$

The intransitive verb *talks* has the same representation as before, now with the adapted labeling of wires

$$|\text{talks}\rangle = \sum_{mn} t_{mn}\ |n_m\rangle_7 |s_n\rangle_8. \tag{39}$$

For clarity, we flesh out the computation of the contractions that involve the extra index space. The semantic contraction of *who* with *talks*, following the interpretation of the syntactic contraction, results in

$$\begin{aligned} &\langle \text{who}|\langle \text{talks}|\ \widehat{P}_{67} \otimes \widehat{P}_{58} \otimes \widehat{O}_{1234}\ |\text{who}\rangle|\text{talks}\rangle \\ &= \sum_{a'i'j'm'n'} \langle a'|_1 \langle n_{i'}|_2 \langle a'|_3 \langle s_{j'}|_4 \langle s_{j'}|_5 \langle n_{i'}|_6 t^*_{m'n'} \langle n_{m'}|_7 \langle s_{n'}|_8 \\ &\quad \cdot \widehat{O}_{1234} \sum_{aijmn} |a\rangle_1 |n_i\rangle_2 |a\rangle_3 |s_j\rangle_4 |s_n\rangle_5 |n_m\rangle_6 t_{mn} |n_i\rangle_7 |s_j\rangle_8 \\ &= \sum_{a'i'j'm'n'} \langle a'|_1 \langle n_{i'}|_2 \langle a'|_3 \langle s_{j'}|_4 t^*_{m'n'} \widehat{O}_{1234}\ |a\rangle_1 |n_{m'}\rangle_2 |a\rangle_3 |s_{n'}\rangle_4 t_{i'j'} \\ &= \sum_{a'i'j'} t_{i'j'} \langle a'|_1 \langle n_{i'}|_2 \langle a'|_3 \langle s_{j'}|_4 \widehat{O}_{1234}\ \sum_{ail} t^*_{ij} |a\rangle_1 |n_i\rangle_2 |a\rangle_3 |s_j\rangle_4. \end{aligned} \tag{40}$$

From this, we read off the question representation, rewriting the indices:

$$[\![\text{who talks}]\!] = \sum_{aij} t^*_{ij} |a\rangle_1 |n_i\rangle_2 |a\rangle_3 |s_j\rangle_4. \tag{41}$$

To be the input of the quantum search algorithm, the full input needs to correspond to an equal superposition of all possible answers. This can be achieved by entangling the distributional representation of the answers with the corresponding index

$$|\text{answers}\rangle = \sum_{bp} W^b_p |n_p\rangle_9 |b\rangle_{10}, \tag{42}$$

followed by a contraction with the question representation, which happens strictly at the semantic level. Hence,

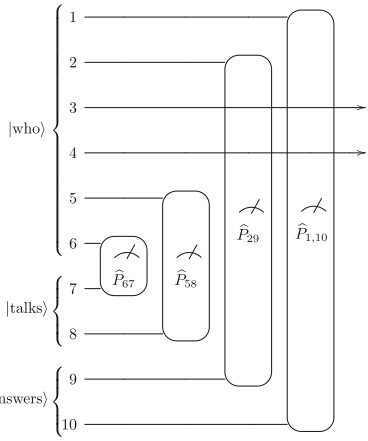


Fig. 8 Quantum circuit that generates the input of the first Grover iteration for question-answering

$$
\begin{aligned}
&\langle \text{who talks}|\langle \text{answers}|\widehat{P}_{29}\otimes\widehat{P}_{1,10}\otimes\widehat{O}_{34}|\text{who talks}\rangle|\text{answers}\rangle = \\
&\sum_{a'i'l'} t_{i'j'}\langle a'|_1\langle n_{i'}|_2\langle a'|_3\langle s_{j'}|_4 \sum_{b'p'} W_{p'}^{*b'}\langle n_{p'}|_9\langle b'|_{10} \\
&\quad\cdot\widehat{O}_{34}\sum_{ail} t_{ij}^{*}|b\rangle_1|n_p\rangle_2|a\rangle_3|s_j\rangle_4 \sum_{bp} W_p^{b}|n_i\rangle_9|a\rangle_{10} \\
&= \sum_{a'i'j'} W_{i'}^{a'} t_{i'j'}\langle a'|_3\langle s_{j'}|_4\widehat{O}_{34}\sum_{amn} W_i^{*a} t_{ij}^{*}|a\rangle_3|s_j\rangle_4,
\end{aligned}
\tag{43}
$$

such that the effective input state to the Grover's algorithm is

$$
|\Psi_{initial}\rangle = \sum_{aij} W_i^{*a} t_{ij}^{*}|a\rangle_3|s_j\rangle_4, \tag{44}
$$

which represents an entanglement between the indices of possible answers and truth values. This process is represented by the circuit in Fig. 8.

4.3 Oracle and inversion

Grover's algorithm requires a normalized state as initial input. Since the amplitudes in the state in Eq. (44) have information about whether a certain answer is correct, they uniquely associate each word indexed by a with one of the basis states $|s_j\rangle$, in such a way that $\sum_i W_i^{*a} t_{ij}^{*}$ is null if the combination between word index a and truth value j is not correct, and otherwise equal to one. Since every word should be either true or false, that leaves us with precisely P independent and equally summed states. Therefore, if the indices aj are abbreviated by one index x, the normalized state is given as

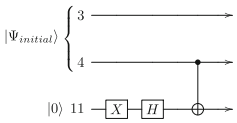


Fig. 9 Oracle for question-answering

$$|\Psi_{initial}\rangle = \frac{1}{\sqrt{P}} \sum_{x=0}^{P-1} |x\rangle, \tag{45}$$

with

$$|x\rangle = \sum_i W_i^{*a} t_{ij}^* |a\rangle_3 |s_j\rangle_4. \tag{46}$$

The oracle applied to this input state takes the form of the circuit in Fig. 9.

The states $|a\rangle$ on the first p qubits, being in one-to-one correspondence with each of the possible solutions, are the complete set of states that build up the equal superposition $|\Psi\rangle = H^{\otimes p}|0\rangle^{\otimes p}$, as in Eq. (31). In terms of the states that correspond to words that are correct or incorrect, we can rewrite $|\Psi\rangle$ as

$$|\Psi\rangle = \cos\left(\frac{\theta}{2}\right)|\alpha\rangle_3 + \sin\left(\frac{\theta}{2}\right)|\beta\rangle_3, \tag{47}$$

with $|\alpha\rangle$ the normalized sum of all states that correspond to words that are not solutions, and $|\beta\rangle$ to the normalized sum of those that correspond to words that are solutions. Using this notation, the $|\Psi_{initial}\rangle$ state that we obtain using the contractions can be expressed as

$$|\Psi_{initial}\rangle = \cos\left(\frac{\theta}{2}\right)|\alpha\rangle_3|0\rangle_4 + \sin\left(\frac{\theta}{2}\right)|\beta\rangle_3|1\rangle_4. \tag{48}$$

Though there is entanglement between the first p qubits and the last one, this is a pure state in the $p+1$ qubit space, as it results from the measurement of the permutation operators, as shown in Sect. 3.2. As such, there is a unitary transformation that generates it from $|0\rangle^{\otimes p+1}$ as

$$|\Psi_{initial}\rangle = U|0\rangle^{p+1}. \tag{49}$$

Using this, we can construct the rotation part of the Grover algorithm as

$$U(2|0\rangle\langle 0| - 1)U^\dagger = 2|\Psi_{initial}\rangle\langle\Psi_{initial}| - 1. \tag{50}$$

It follows that the Grover iteration applied on the input state in Eq. (48) using the oracle and the rotation in Eq. (50) gives the desired outcome of

$$(2|\Psi_{initial}\rangle\langle\Psi_{initial}| - 1)O(|\Psi_{initial}\rangle) = \cos(\frac{3\theta}{2})|\alpha\rangle_3|0\rangle_4 + \sin(\frac{3\theta}{2})|\beta\rangle_3|1\rangle_4. \quad (51)$$

This is precisely what we expect to obtain. For the second iteration, the oracle acts as desired, and so does the rotation. After a number of iterations, only the states associated with $|1\rangle_4$ have positive amplitude, which means that we are certain to measure the index of a word that corresponds to a correct answer when we make a measurement on the first p qubits. Thus, we have obtained a correct answer with quadratic speedup due to the quantum search algorithm.

5 Conclusion and outlook

In this paper, we introduced two main developments in the application of quantum computation to natural language processing. The first one is a tensor contraction scheme on quantum circuits. Taking quantum states as the representations of all input words, contractions are then identified with the expectation value of an appropriate permutation operator. Doing this, we are not only able to reproduce previous analytical results, but we also allow for complex values and create quantum circuits that are equipped to deal with the syntactic ambiguities in Ref. [20]. With this setup, each reading of an ambiguous phrase corresponds to a particular circuit, and different readings are interchangeable upon the application of a number of swap gates. Controlling on these swap gates, we can obtain a true quantum superposition of the multiple readings. This covers the problem of how to deal with multiple readings in real time, without the need to assume any contextualization. While this addresses the question of syntactic ambiguities by making use of the quantum superposition principle, ambiguities at the word level can be immediately accommodated for by using density matrices [15–17, 35], instead of the pure states we use here for simplicity. A generalization to other sentence-level ambiguities constitutes further work, in the expectation that the use of different controls allows for different readings simultaneously in the output state. Note that, in terms of a concrete implementation, the permutation between two qubits used to generate an equal superposition of readings from an ambiguous input takes the form of a Fredkin gate, or a CSWAP gate, which might add considerable circuit complexity, but this is expected to be compensated by the fact that the number of two-qubit operations only scales linearly with an increasing number of readings, since for these types of ambiguities all permutations can be generated via sets of SWAP operations. We leave a more robust exploration of these technical constraints to future work.

The second development builds on this quantum framework and consists of a quantum search algorithm that is able to find the answer to a *wh*-question with quantum speedup. As input, the algorithm takes a multipartite state in quantum superposition, representing a *wh*-question and its possible answers, and performs a series of tensor contractions as established previously. A series of gates then acts repeatedly on the post-contraction state, guaranteeing that a correct answer to the question is obtained upon a single final measurement. Our algorithm takes advantage of intrinsic quantum features to identify and deliver a correct answer with quantum speedup of quadratic

order, when compared to the classical alternative of checking every possible answer. We are thus able to provide a correct answer using information given directly by the tensor contractions of representations of words as proposed by DisCoCat, and without needing to hand-feed any labels nor to learn the answers to other questions. Our approach thus shows how quantum circuit implementations can break with the widely accepted "learning paradigm" of current NLP approaches to question answering and other tasks used in Ref. [13], providing a scalable approach to open-ended questions. Our approach differs from that of Ref. [9] also in the sense that we keep all words as input states, instead of representing words from complex types as gates that modify circuit inputs, remaining closer to the compositional spirit of the syntax and therefore being more easily extensible to larger language fragments. Further work includes finding an effective implementation of the measurement of the permutation operator for an arbitrary number of qubits, possibly making use of the Hadamard test [36], and understanding how to find a universal form of the inversion operator that does not depend on $|\alpha\rangle$ and $|\beta\rangle$ separately. An extension of the present formalism can furthermore account for a better understanding of the temporal evolution of the meanings of sentences.

Acknowledgements We would like to thank the anonymous referees for the helpful comments. This work is supported by the Utrecht University's Complex Systems Fund, with special thanks to Peter Koeze, and the D-ITP consortium, a program of the Netherlands Organization for Scientific Research (NWO) that is funded by the Dutch Ministry of Education, Culture and Science (OCW).

Data Availability Statement All data generated or analyzed during this study are included in this published article.

Appendix A

Measuring the permutation operator on two qubits

In this appendix, we show that for the measurement of the permutation operator applied to two qubits it suffices to measure the input states in the Bell basis. The two input qubits have the four possible joint states in the standard basis, given by

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, |10\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, |01\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, |11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \tag{52}$$

The permutation operator applied to two qubits is equivalent to the SWAP gate S. In this basis, this operator has the matrix representation

$$S = \begin{pmatrix} 1\,0\,0\,0 \\ 0\,0\,1\,0 \\ 0\,1\,0\,0 \\ 0\,0\,0\,1 \end{pmatrix}. \tag{53}$$

The eigenstates of this operator are the well-known singlet and triplet states that represent the joint spin of two spin-1/2 particles. With eigenvalue -1, we have the singlet state

$$|0,0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ -1 \\ 1 \\ 0 \end{pmatrix}, \tag{54}$$

and with eigenvalue 1 we have the three triplet states

$$|1,-1\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, |1,0\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \text{ and } |1,1\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \tag{55}$$

expressed in the standard basis as

$$|0,0\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle), \tag{56}$$

$$|1,-1\rangle = |00\rangle, \tag{57}$$

$$|1,0\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \tag{58}$$

$$|1,1\rangle = |11\rangle. \tag{59}$$

In its turn, the Bell basis can be expressed in terms of the standard basis in the following way:

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{60}$$

$$|\beta_{01}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \tag{61}$$

$$|\beta_{10}\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \tag{62}$$

$$|\beta_{11}\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle). \tag{63}$$

The Bell states can thus be rewritten using the total spin eigenstates of S, given in (56) to (59), as:

$$|\beta_{00}\rangle = \frac{1}{\sqrt{2}}(|1,-1\rangle + |1,1\rangle) \tag{64}$$

$$|\beta_{01}\rangle = |1,0\rangle \tag{65}$$

$$|\beta_{10}\rangle = \frac{1}{\sqrt{2}}(|1,-1\rangle - |1,1\rangle) \tag{66}$$

$$|\beta_{11}\rangle = |0,0\rangle. \tag{67}$$

Because any linear combination of degenerate eigenstates is also an eigenstate of that operator with the same eigenvalue [proof: $A\mathbf{v} = \lambda\mathbf{v}$, $A\mathbf{w} = \lambda\mathbf{w} \Rightarrow A(a\mathbf{v} + b\mathbf{w}) = \lambda(a\mathbf{v} + b\mathbf{w})$], we see that $|\beta_{00}\rangle$, $|\beta_{01}\rangle$, and $|\beta_{10}\rangle$ are eigenstates of S with eigenvalue 1, and $|\beta_{11}\rangle$ is an eigenstate with eigenvalue -1. Therefore, we can conclude that the Bell basis also diagonalizes the permutation operator, and as such repeated measurements of the qubits in this basis allow us to directly compute the expectation value of the operator in the input states. So, for a two-qubit input state

$$|\Phi\rangle = \sum_{ij} a_i b_j |ij\rangle, \tag{68}$$

with $i, j \in \{0, 1\}$, the expectation value of the S operator is given by

$$\langle S\rangle_\Phi = |\langle\beta_{00}||\Phi\rangle|^2 + |\langle\beta_{01}||\Phi\rangle|^2 + |\langle\beta_{10}||\Phi\rangle|^2 - |\langle\beta_{11}||\Phi\rangle|^2. \tag{69}$$

If we are in the possession of a measuring device that can only measure in the standard basis, we must transform our input states with the inverse transformation that generates the Bell states. This serves to guarantee that an outcome $|ij\rangle$ is in fact as likely as the measurement of $|\beta_{ij}\rangle$ if the input states were measured directly in the Bell basis.

References

1. Coecke, B., Sadrzadeh, M., Clark, S.: Mathematical foundations for a compositional distributional model of meaning. Lambek Festschr., Linguist. Anal. **36**(1–4), 345–384 (2010)
2. Coecke, B., Grefenstette, E., Sadrzadeh, M.: Lambek vs. lambek: functorial vector space semantics and string diagrams for lambek calculus. Ann. Pure Appl. Logic **164**(11), 1079–1100 (2013)
3. Lambek, J.: The mathematics of sentence structure. Am. Math. Mon. **65**(3), 154–170 (1958)
4. Lambek, J.: Type grammar revisited. In: Lecomte, A., Lamarche, F., Perrier, G. (eds.) Logical Aspects of Computational Linguistics, Second International Conference, LACL '97. Lecture Notes in Computer Science, vol. 1582, pp. 1–27 (1997)
5. Turney, P.D., Pantel, P.: From frequency to meaning: vector space models of semantics. J. Artif. Intell. Res. **37**, 141–188 (2010)
6. Wijnholds, G., Sadrzadeh, M., Clark, S.: Representation learning for type-driven composition. In: Proceedings of the 24th Conference on Computational Natural Language Learning (CoNLL '20), pp. 313–324 (2020)
7. Giovannetti, V., Lloyd, S., Maccone, L.: Quantum random access memory. Phys. Rev. Lett. **100**(16), 160501 (2008)

8. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J.C., Barends, R., Biswas, R., Boixo, S., Brandao, F.G., Buell, D.A., et al.: Quantum supremacy using a programmable superconducting processor. Nature **574**(7779), 505–510 (2019)
9. Coecke, B.: The mathematics of text structure. arXiv:1904.03478 (2019)
10. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. Nature **549**(7671), 195–202 (2017)
11. Meichanetzidis, K., Gogioso, S., De Felice, G., Chiappori, N., Toumi, A., Coecke, B.: Quantum natural language processing on near-term quantum computers. arXiv:2005.04147 (2020)
12. Coecke, B., de Felice, G., Meichanetzidis, K., Toumi, A.: Foundations for near-term quantum natural language processing. arXiv:2012.03755 (2020)
13. Meichanetzidis, K., Toumi, A., de Felice, G., Coecke, B.: Grammar-aware question-answering on quantum computers. arXiv:2012.03756 (2020)
14. Lorenz, R., Pearson, A., Meichanetzidis, K., Kartsaklis, D., Coecke, B.: QNLP in practice: Running compositional models of meaning on a quantum computer. arXiv:2102.12846 (2021)
15. Piedeleu, R., Kartsaklis, D., Coecke, B., Sadrzadeh, M.: Open system categorical quantum semantics in natural language processing. In: Moss, L., Sobociński, P. (eds.) 6th International Conference on Algebra and Coalgebra in Computer Science (CALCO'15), pp. 267–286 (2015)
16. Sadrzadeh, M., Kartsaklis, D., Balkir, E.: Sentence entailment in compositional distributional semantics. Ann. Math. Artif. Intell. **82**(4), 189–218 (2018)
17. Bankova, D., Coecke, B., Lewis, M., Marsden, D.: Graded hyponymy for compositional distributional semantics. J. Lang. Modell. **6**(2), 225–260 (2019)
18. Meyer, F., Lewis, M.: Modelling lexical ambiguity with density matrices. In: Proceedings of the 24th Conference on Computational Natural Language Learning, pp. 276–290. Association for Computational Linguistics, ??? (2020)
19. Shiebler, D., Toumi, A., Sadrzadeh, M.: Incremental monoidal grammars. arXiv:2001.02296 (2020)
20. Correia, A.D., Moortgat, M., Stoof, H.T.C.: Density matrices with metric for derivational ambiguity. J. Appl. Logics **7**(5), 795–822 (2020)
21. Correia, A.D., Stoof, H.T.C., Moortgat, M.: Putting a spin on language: A quantum interpretation of unary connectives for linguistic applications. In: 17th International Conference on Quantum Physics and Logic (QPL) '20. Electronic Proceedings of Theoretical Computer Science, vol. 340, pp. 114–140 (2021)
22. Bender, E.M., Gebru, T., McMillan-Major, A., Shmitchell, S.: On the dangers of stochastic parrots: Can language models be too big? In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, pp. 610–623 (2021)
23. Ajdukiewicz, K.: Die syntaktische Konnexität. Studia philosophica, 1–27 (1935)
24. Moortgat, M.: Categorial type logics. In: Handbook of Logic and Language, pp. 93–177 (1997)
25. Soares, M.A.C., Parreiras, F.S.: A literature review on question answering techniques, paradigms and systems. J. King Saud Univ.-Comput. Inf. Sci. **32**(6), 635–646 (2020)
26. Lin, D.: Automatic retrieval and clustering of similar words. In: 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Vol. 2, pp. 768–774 (1998)
27. Navigli, R., Crisafulli, G.: Inducing word senses to improve web search result clustering. In: Proceedings of the 3rd Conference on Empirical Methods in Natural Language Processing (EMNLP '10), pp. 116–126 (2010)
28. Nasiruddin, M.: A state of the art of word sense induction: A way towards word sense disambiguation for under-resourced languages. In: Proceedings of RECITAL 2013, pp. 192–205 (2013)
29. Boleda, G.: Distributional semantics and linguistic theory. Annu. Rev. Linguist. **6**, 213–234 (2020)
30. Rieger, B.B.: On Distributed Representation in Word Semantics. International Computer Science Institute, Berkeley, CA (1991)
31. Harris, Z.S.: Distributional structure. Word **10**(2–3), 146–162 (1954)
32. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. arXiv preprint arXiv:1310.4546 (2013)
33. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
34. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the 28th Annual ACM Symposium on Theory of Computing, pp. 212–219 (1996)

35. Meyer, F., Lewis, M.: Modelling lexical ambiguity with density matrices. In: Proceedings of the 24th Conference on Computational Natural Language Learning (CoNLL '20), pp. 276–290 (2020)
36. Aharonov, D., Jones, V., Landau, Z.: A polynomial quantum algorithm for approximating the Jones polynomial. Algorithmica **55**(3), 395–421 (2009)