Language-independence of DisCoCirc's Text Circuits: English and Urdu

Muhammad Hamza Waseem
Clarendon Laboratory, Department of Physics
University of Oxford
Compositional Intelligence Team
Quantinuum
hamza.waseem@physics.ox.ac.uk

Jonathon Liu
Compositional Intelligence Team
Quantinuum
jonathon.liu@cambridgequantum.com

Vincent Wang-Maścianica
Quantum Group, Department of Computer Science
University Oxford
Compositional Intelligence Team
Quantinuum
vincent.wang@cambridgequantum.com

Bob Coecke
Compositional Intelligence Team
Quantinuum
bob.coecke@cambridgequantum.com

DisCoCirc [6] is a newly proposed framework for representing the grammar and semantics of texts using compositional, generative circuits. While it constitutes a development of the Categorical Distributional Compositional (DisCoCat) framework, it exposes radically new features. In particular, [14] suggested that DisCoCirc goes some way toward eliminating grammatical differences between languages.

In this paper we provide a sketch that this is indeed the case for restricted fragments of English and Urdu. We first develop DisCoCirc for a fragment of Urdu, as it was done for English in [14]. There is a simple translation from English grammar to Urdu grammar, and vice versa. We then show that differences in grammatical structure between English and Urdu - primarily relating to the ordering of words and phrases - vanish when passing to DisCoCirc circuits.

1 Introduction

In [6] Coecke introduced the Compositional Distributional Circuits (DisCoCirc) framework for modelling the structure of meaning in natural languages, building further on earlier work on the Categorical Distributional Compositional model for combining grammar and semantics [4, 11]. An important distinction between the two is that DisCoCirc is able to model not only the meaning of individual sentences, but also the interaction of sentences giving rise to the meaning of texts generally. The central idea is that the information associated with noun entities appearing in the text (encoded in circuit wires) are updated by sentences (modelled as gates) as the text progresses.

DisCoCirc admits a two-dimensional string diagrammatic formalism, inspired by quantum circuits/networks, and therefore provides prospects for quantum-computational natural language processing for texts comprising multiple sentences or paragraphs [2]. It also has been applied to a number of problems including spatio-temporal models of language meaning [15], logical and conversational negation in natural language [10, 13], and solving logical puzzles [7, 8]. The relationship between DisCoCirc and discourse-representation theory [9] is also discussed in [14].

Recently, Wang-Maścianica, Liu and Coecke proposed a method for generating DisCoCirc diagrams for a significant fragment of English [14]. They started by creating a hybrid grammar for English text,

M. Moortgat and G. Wijnholds: End-to-End Compositional Models of Vector-Based Semantics, 2022 (E2ECOMPVEC)
EPTCS 366, 2022, pp. 50–60, doi:10.4204/EPTCS.366.7

incorporating 'phrase structure', 'pronominal links', 'phrase regions', etc. These hybrid grammar representations of text were then translated into DisCoCirc text circuits, via an intermediate structure called 'text diagrams' that involving string diagrams. The entire translation process preserves the compositionality and connectedness of text meaning.

The same work describes how in the reverse direction, text circuits can be used as a generative grammar. For each freely generated text circuit, we can write some corresponding text. In this paper, we use 'text' to refer not only to a string of words forming sentences, but to this string of words further endowed with a hybrid grammar structure. The text corresponding to a given text circuit is not unique owing to the loss of grammatical 'bureaucracy' [14] in the passage from one-dimensional syntax to two-dimensional text circuits.[1] Here, grammatical bureaucracy is used in a broad sense to refer to all of the stylistic choices one must make when communicating some desired meaning in the form of a text. That is, whenever two different texts communicate what is essentially the same meaning, we attribute the differences in the structure of these texts to grammatical bureaucracy. Thus grammatical bureaucracy includes syntactic rules like those governing word order, but also choices like the use of pronouns, whether to use a single long sentence with multiple clausal constructions or multiple short sentences, etc.

It was in particular suggested that because of eliminating grammatical bureaucracy and stylistic choices embedded in a particular natural language, text circuits are to some degree language-independent. We take this suggestion seriously in this paper, and provide an outline of how DisCoCirc undoes the grammatical bureaucracy relating to word and phrase order for restricted fragments of English and Urdu. Our main argument is structured as follows.

- In [14], a hybrid grammar was developed for English. A surjection from the set of all English text generated with the English hybrid grammar to the set of all English text circuits was demonstrated:

$$\text{English text} \twoheadrightarrow \text{English text circuits}$$

- In a similar vein, in this paper, we describe how the hybrid grammar can be adapted for Urdu. We then provide rules for its translation into text diagrams and text circuits, which is essentially the same as in [14]. We show that this gives a surjective map from the set of all Urdu text generated with Urdu hybrid grammar to the set of all Urdu text circuits:

$$\text{Urdu text} \twoheadrightarrow \text{Urdu text circuits}$$

- For these restricted fragments, there is a clear isomorphism between the between the hybrid grammars for English and Urdu:

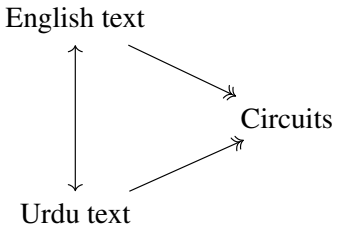
$$\text{English hybrid grammar} \simeq \text{Urdu hybrid grammar}$$

- Given the above correspondence, it turns out that text circuits for English and Urdu become the same, up to translating the labels on the gates. In other words, the following diagram commutes:

$$\begin{array}{ccc} \text{English text} & \twoheadrightarrow & \text{English Circuits} \\ \Big\updownarrow{\scriptstyle \textit{Grammar \& Dictionary}} & & \Big\updownarrow{\scriptstyle \textit{Dictionary}} \\ \text{Urdu text} & \twoheadrightarrow & \text{Urdu Circuits} \end{array}$$

[1] Some of this was already observed in [5].

Now formally speaking, our texts consist of a hybrid grammar structure with labels (words) provided by a dictionary. If we restrict to just the grammatical structures and forget the language-specific word labels, the circuits for English and Urdu become literally the same. In this case, the above diagram reduces to:



The paper is organised as follows. The hybrid grammar, text diagrams and text circuits for English, along with an example, are reviewed in Section 2. Hybrid grammar for Urdu is introduced in Section 3, followed by presentation of our sketch proof through an expository example in Section 4. Section 5 concludes the paper.

2 Grammar, diagrams and circuits for English text

In this section we quickly review the hybrid (generative) grammar for English developed in [14]. First, simple sentences were modelled, containing verbs, adverbs, adjectives and adpositions. Then, pronominal links were introduced to account for recurring nouns and pronoun-referent pairs. Rewrite rules were introduced that allowed for the fusion of simple sentences into more complex ones, and the introduction of relative pronouns. Rules for modeling verbs with sentential complement and 'phrase scope boundary' were introduced to accommodate compound sentences formed of components which are themselves sentences.

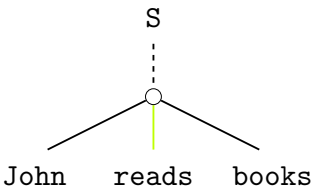
The hybrid grammar begins with a standard phrase structure grammar that generates our simple sentences, based on a string rewrite system with finitely many production rules of form $\alpha \mapsto \beta$. These are valid transformations of strings of symbols represented by α and β. Individual symbols may be phrase components or entire words of the language we are modelling. In the latter case, the symbol is terminal, meaning no more rewrite rules can be further applied. In a grammar such as ours, a particular language comprises all the strings of terminal symbols that can be generated by applying finitely many production rules (associated with that language) to a start symbol, S. For example, using the rules:

$$\begin{aligned} \mathtt{S} &\mapsto \mathtt{NP}_1 \cdot \mathtt{TVP} \cdot \mathtt{NP}_2 \\ \mathtt{NP}_1 &\mapsto \text{John} \\ \mathtt{TVP} &\mapsto \text{reads} \\ \mathtt{NP}_2 &\mapsto \text{books} \end{aligned}$$

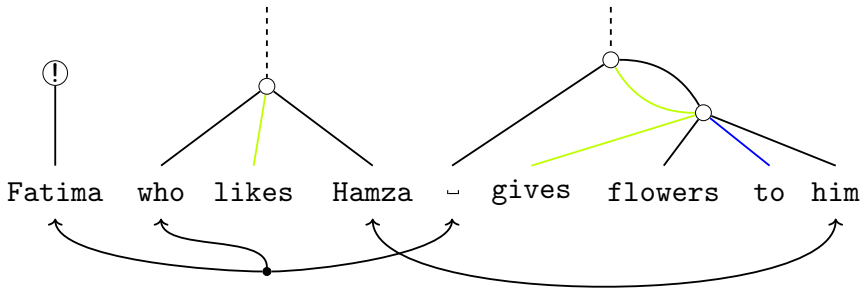
where NP and TVP represent noun phrase and transitive verb phrase respectively; we can generate sentences like 'John reads books':

$$\begin{aligned} \mathtt{S} &\mapsto \mathtt{NP}_1 \cdot \mathtt{TVP} \cdot \mathtt{NP}_2 \\ &\mapsto \text{John} \cdot \mathtt{TVP} \cdot \mathtt{NP}_2 \\ &\mapsto \text{John} \cdot \text{reads} \cdot \mathtt{NP}_2 \\ &\mapsto \text{John} \cdot \text{reads} \cdot \text{books} \end{aligned}$$

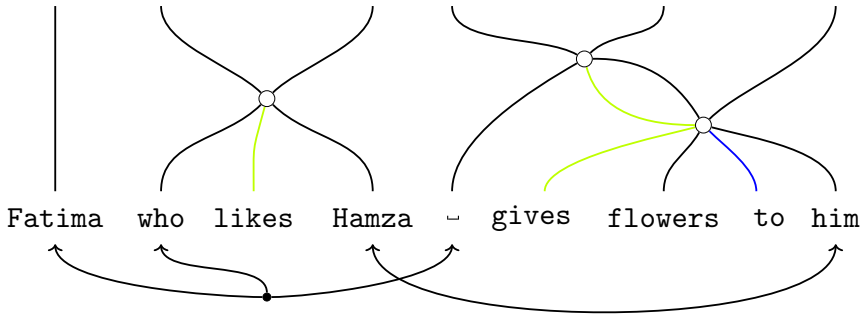
Rewriting of strings can be represented as two-dimensional tree diagrams, read from top to bottom. Our example sentence can be represented as



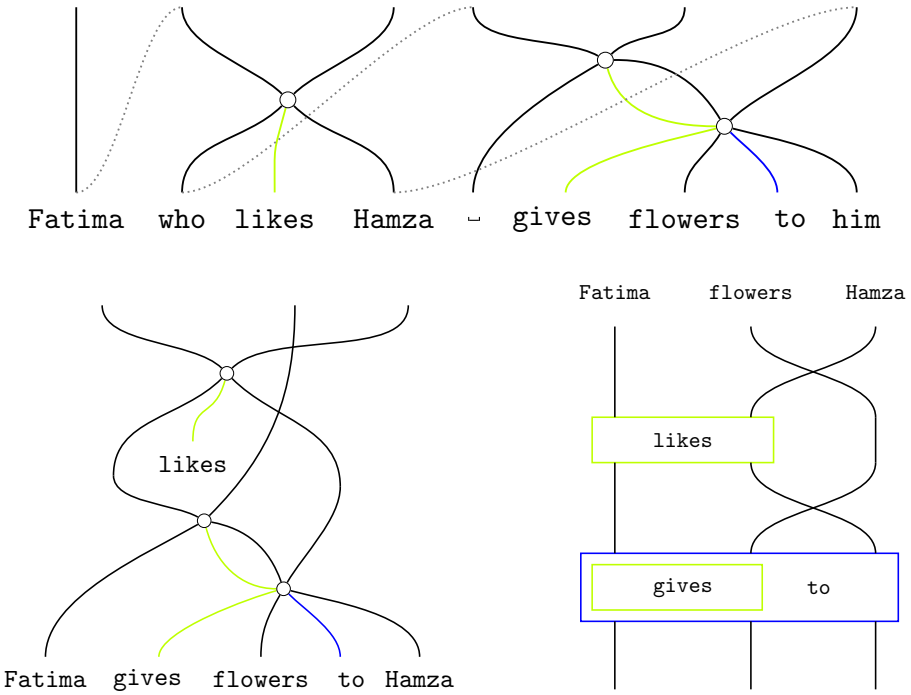
There may be multiple edges between parent and child nodes. Symbolic labels can be dropped for intermediate edges and, instead, a color coding can be used. In [14], production rules and the corresponding planar trees were defined for simple sentences with verbs, adjectives, adverbs and adpositions, and compound sentences formed with pronominal links and phrase scope structures.



To do away with artefacts such as those handling pronominal links, *text diagrams* are introduced. Based on string diagrams [1, 12, 3], which is a structural framework for boxes with inputs and outputs, it allows parallel and sequential composition. S-type is replaced by NP-types, the number of which depends on the sentence. This modification allows composition of tree fragments (while respecting the grammatical types). Moving to text diagrams, pronominal links and phrase scope constructions become part of one unified mathematical framework.



Finally, rewrite rules are introduced to map fragments of text diagrams to *text circuits*. Noun types are represented by wires. Adjectives and intransitive verbs are represented by single-input-single-output gates, and transitive verbs by double-input-double-output gates, acting on noun wires. Adverbs, adpositions and sentential complements, modifying verbs, are represented as boxes that contain the verb boxes/gates being modified. Conjunctions are taken to be boxes which contain two sentence circuits that are being connected.



It was shown in [14] that English text generated from the aforementioned hybrid grammar is surjective to text circuits. In this paper, we demonstrate a similar result for Urdu.

3 A hybrid grammar for Urdu

Since a language is specified by set of production rules, different production rules lead to different languages. Translating to Urdu, 'John reads books' can be transliterated[2] in English as

John *kitabein* *parhta hai* (Urdu)
John books reads

Using the production rules

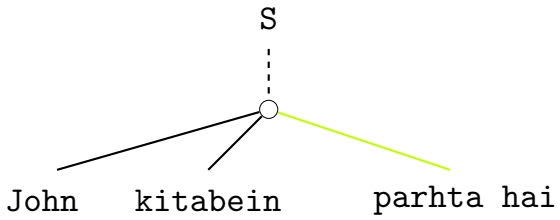
$$\begin{aligned} \mathtt{S} &\mapsto \mathtt{NP}_1 \cdot \mathtt{NP}_2 \cdot \mathtt{TVP} \\ \mathtt{NP}_1 &\mapsto \text{John} \\ \mathtt{NP}_2 &\mapsto \text{kitabein} \\ \mathtt{TVP} &\mapsto \text{parhta hai} \end{aligned}$$

we can generate the sentence under discussion

$$\begin{aligned} \mathtt{S} &\mapsto \mathtt{NP}_1 \cdot \mathtt{NP}_2 \cdot \mathtt{TVP} \\ &\mapsto \text{John} \cdot \mathtt{NP}_2 \cdot \mathtt{TVP} \\ &\mapsto \text{John} \cdot \text{kitabein} \cdot \mathtt{TVP} \\ &\mapsto \text{John} \cdot \text{kitabein} \cdot \text{parhta hai} \end{aligned}$$

[2]Urdu script is written from right to left, opposite to English. Throughout this paper, for ease of readability and linguistic analysis, we shall use English transliteration of Urdu text and hence use left-to-right script.

the tree diagram of which is given by



From this example already, we can spot an obvious difference between English and Urdu: the order of subject, verb and object. More particularly, the verb is usually placed at the end of the sentence in Urdu, unlike in English. In fact, as we shall see in this paper, this contrast plays a significant role in differentiating Urdu and English grammars.

We develop production rules and tree diagrams for the fragment of Urdu text (including verbs, adjectives, adverbs, adpositions, pronominal links, phrase scope) corresponding to that of English in [14]. Doing this, we realise that many of the rules and tree fragments are in fact the same. The ones that are different differ mainly in the relative placement of the verb. See Table 1.

4 Text diagrams and circuits for Urdu

4.1 Main result

Let $\mathtt{E}$ denote the set of generators of the hybrid English grammar, summarised in Table 1. Note we have restricted to a version of hybrid grammar generated by the explicitly context-free rules (i.e., we keep all generators except for the rule involving ADP and TVP, and the rules that allow noun wires to leave phrase scope). Let $\mathtt{T}_\mathtt{E}$ denote the set of all English text constructed with grammar $\mathtt{E}$, and let $\mathtt{C}_\mathtt{E}$ denote the set of all text circuits for English. Then, there exists a surjection $\mathtt{T}_\mathtt{E} \twoheadrightarrow \mathtt{C}_\mathtt{E}$ [14].

We create a set of generators for Urdu grammar $\mathtt{U}$ which closely correspond with the English generators $\mathtt{E}$. Let $\mathtt{T}_\mathtt{U}$ denote the set of all Urdu text generated with $\mathtt{U}$. Let $\mathtt{C}_\mathtt{U}$ denote the set of all text circuits for Urdu. Then,

- there exists a surjection $\mathtt{T}_\mathtt{U} \twoheadrightarrow \mathtt{C}_\mathtt{U}$, and
- $\mathtt{C}_\mathtt{U}$ is isomorphic to $\mathtt{C}_\mathtt{E}$, i.e., $\mathtt{C}_\mathtt{U} \xrightarrow{\simeq} \mathtt{C}_\mathtt{E}$ (up to word translations at the gate level).

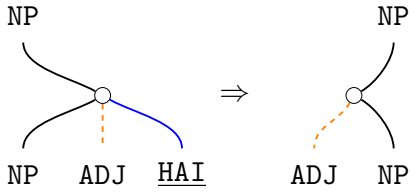
4.2 Urdu text surjects onto circuits

The method of turning hybrid grammar trees into text diagrams is the same in English and Urdu: each component of a hybrid grammar tree is modified so that the number of NP wires for inputs and outputs is made equal, and sentence types S are eliminated.

Table 1 illustrates the similarities and differences between Urdu and English grammar, as reflected in the hybrid grammar trees and text diagrams. The changes are as follows:

- Urdu has Subject-Object-Verb order, in contrast to Subject-Verb-Object as in English.
- The placement of adpositions differs from English; in Urdu, the verb simply comes last.
- Sentential complements precede verbs in Urdu.
- In Urdu, the copula HAI in the postpositional adjectival construction appears to the right of adjective. On the other hand, in English, the copula IS) appears to the left of the adjective.

Everything else remains the same. For instance, the same reductions of text diagrams hold in Urdu as in English. Following is the reduction of a postpositional adjectival construction using a copula HAI($\simeq$ IS) to a prepositional adjective that does not require a copula.



The aforementioned modifications translate the formal claims of [14] for Urdu; any hybrid grammar text for Urdu surjects onto a text circuit.

4.3 English and Urdu give the same circuits

In the case that we only consider context-free generators, the desired isomorphism between English and Urdu circuits essentially follows from the isomorphism between the trees generated by the English and Urdu hybrid grammar.

As a running example, we choose the English sentence 'the young student who sees the honest teacher passionately teach smiles at him', which we translate into the Urdu sentence transliterated as:

nojawan talib-e-ilm jo imandar ustad (ko) shauq se parhate huwe (Urdu)
(the) young student who (the) honest teacher passionately teach

dekhta hai us ki taraf muskurata hai (Urdu)
sees him at smiles

Lemma 1. *Let T_E denote the set of texts generated by the English production rules E, and T_U denote the set of texts generated by the Urdu production rules U. Viewing the syntax trees as rooted labelled trees (vertices are labels like NP or HAI, and the root is the initial sentence type S), there is an isomorphism $T_E \simeq T_U$ in the graph-theoretic sense.*

Essentially, our Urdu generators U correspond exactly to the English generators E, except some of them have the order of their outputs switched around. So, given an English syntax tree generated by some sequence of applications of production rules in E, to translate this to an Urdu syntax tree we simply apply the corresponding rules in U to the appropriate symbols. The exact proof of this is a simple induction. Note that in order to 'apply the right rule to the right symbol', we must track the identities of different symbols - e.g. distinguishing between different instances of NP in our string. This tracking can be done by numbering the symbols with indices. Note also the tracking of the identities of NP's across wires is important later when we move to text diagrams, since we want to identify each 'noun wire' in our diagram with a specific noun entity in our text.

Now we have an isomorphism $T_E \simeq T_U$ between individual syntax trees. Next we introduce pronominal links and the rewrite rules that allow us to adjoin pronominally linked trees into single sentences, thus attaining the full-blown 'hybrid grammar' of [14]. The isomorphism between trees lifts to a kind of

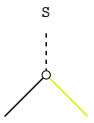
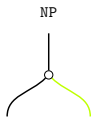
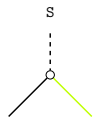
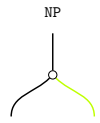
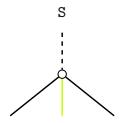
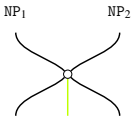
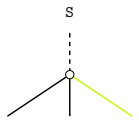
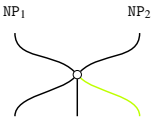
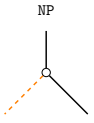
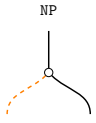
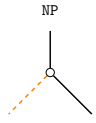
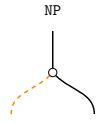
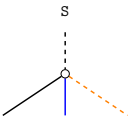
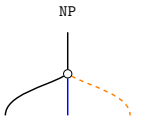
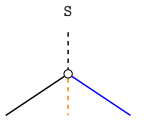
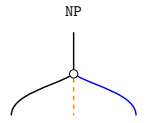
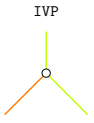
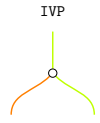
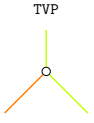
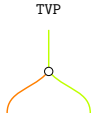
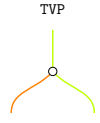
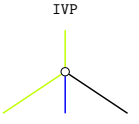
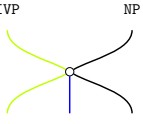
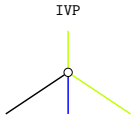
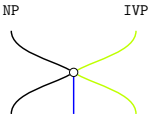
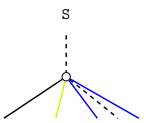
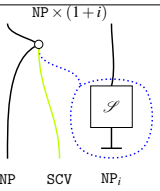
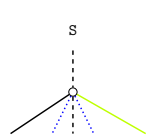
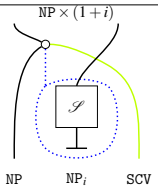
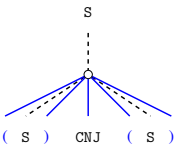
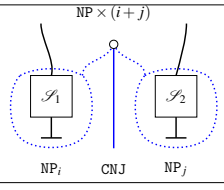
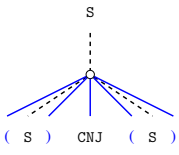
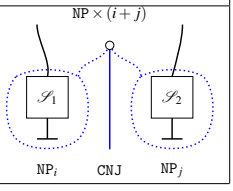
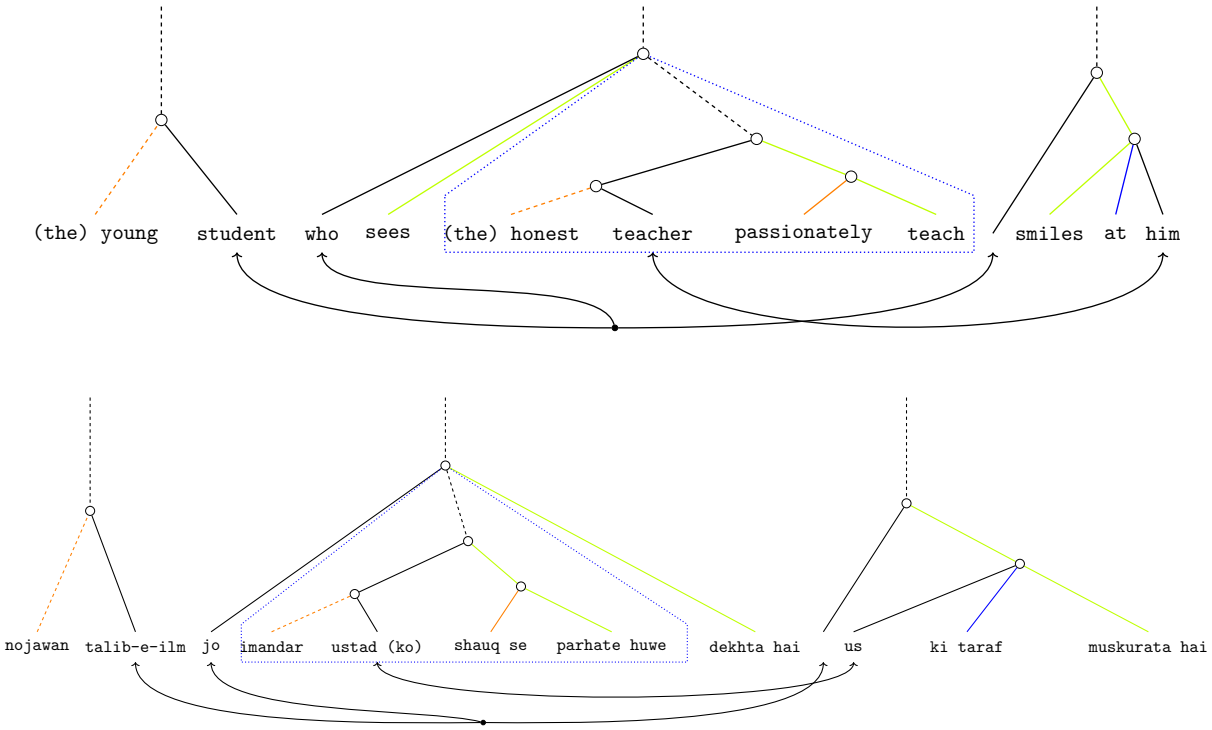
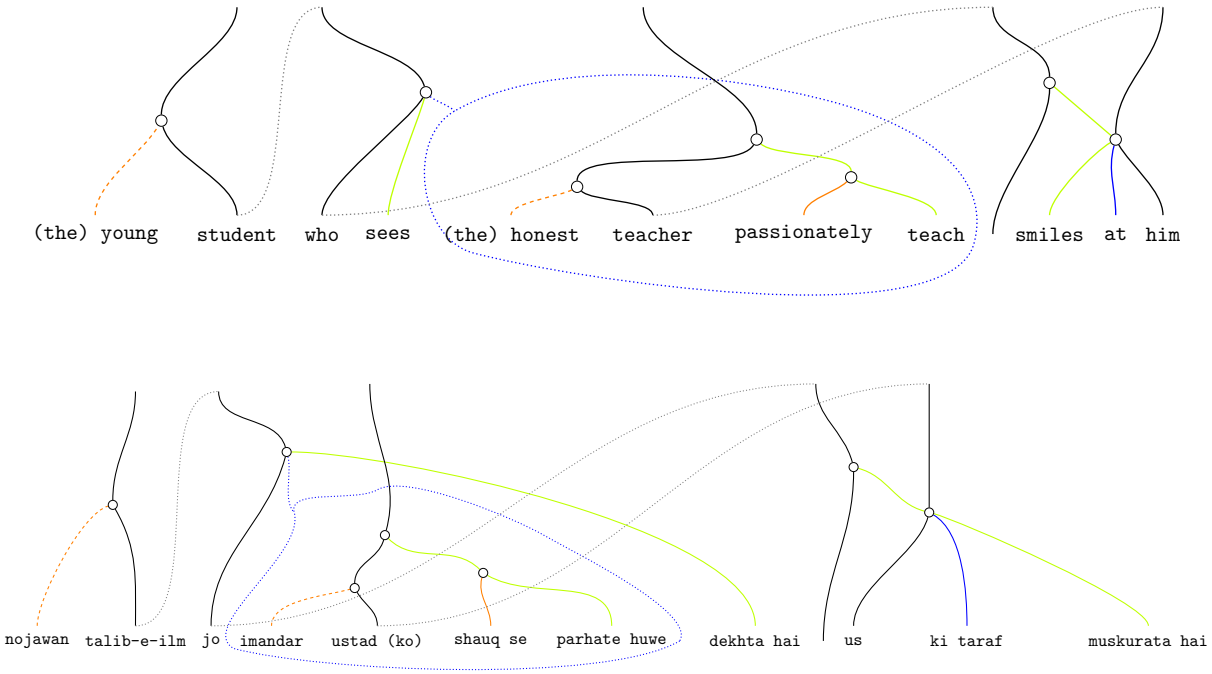
Rule	English grammar	English diagram	Urdu grammar	Urdu diagram
Intrans.Verb				
Trans.Verb				
Adjective(Pre.)				
Adjective(Post.)				
Adverb(IV)				
Adverb(TV)				
Adposition(IV)				
Sent.Comp.Verb				
Conjunction				

Table 1: Generators for hybrid Urdu and English grammar and the corresponding diagrams. Note the different constructions for English and Urdu in the rules: Trans.Verb, Adjective(Post.), Adposition(IV) and Sent.Comp.Verb.

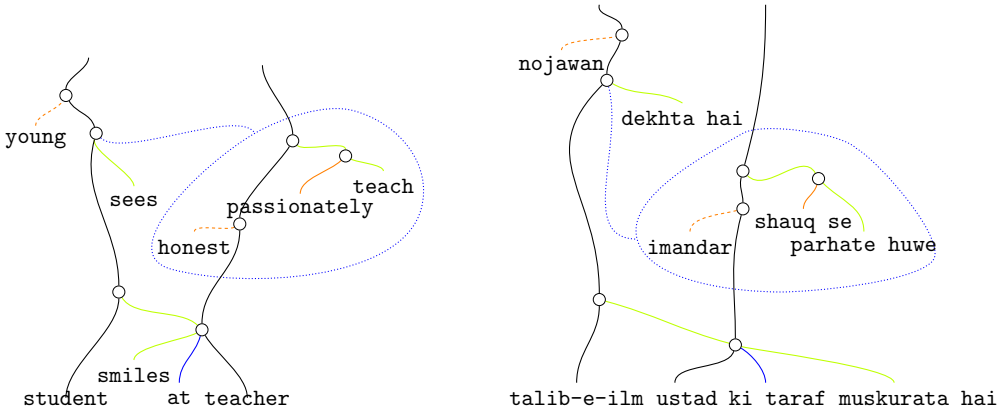
isomorphism between these full hybrid grammar structures.



Next we convert hybrid grammar to text diagrams. We apply the rules in Table 1 for resolving the `S` type into constituent `N` wires and converting phrase scope into bubbles. Then we turn the pronominal links into dashed wires in preparation for composition.

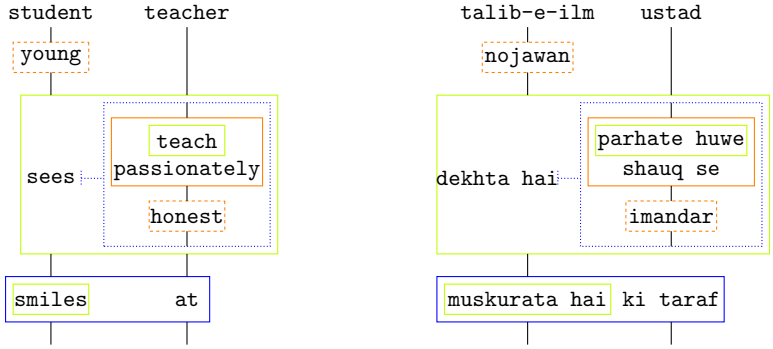


After performing the composition, we recover text diagrams.



We see that the isomorphism of hybrid grammar structures simply lifts to an isomorphism of the structures at each intermediate step. At the step where we reach the level of text diagrams, the isomorphism becomes an exact equality (modulo the different word labels, and allowing a certain degree of topological deformation as we usually do in string diagrams).

With the (topologically) identical structure of the English and Urdu diagrams, we simply apply the same conversion map from text diagrams to text circuits to obtain the same text circuit up to word-translations at the level of individual gates.



This process from text-to-circuits can be (nondeterministically) reversed, which make text circuits generative formalisms for English and Urdu text.

5 Conclusion

A major difference between grammars in different natural languages arises from different word orderings for, for example, subject, verb and object. For instance, in English the usual ordering is subject-verb-object, whereas in Urdu, it is subject-object-verb. These differences, in turn, exist because human verbal communication is restricted to one dimension and different cultures and demographics made different stylistic choices as languages evolved [14]. But there is no such restriction on machines. Two-dimensional grammars such as ours may be a suitable abstraction of text for computers (particularly quantum computers) and may prove advantageous for natural language processing tasks, such as machine translation. This paper moves a step forward in this direction. We emphasise that this paper represents a first step, and there is work to be done to expand the fragments of natural language that we can handle.

Acknowledgments

Muhammad Hamza Waseem acknowledges the Rhodes Trust for funding his graduate studies at Oxford.

References

[1] J. C. Baez & M. Stay (2011): *Physics, topology, logic and computation: a Rosetta Stone*. In B. Coecke, editor: *New Structures for Physics*, Lecture Notes in Physics, Springer, pp. 95–172, doi:10.1007/978-3-642-12821-9_2.

[2] B. Coecke, G. de Felice, K. Meichanetzidis & A. Toumi (2020): *Foundations for Near-Term Quantum Natural Language Processing*. arXiv:2012.03755.

[3] B. Coecke & A. Kissinger (2017): *Picturing Quantum Processes. A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, doi:10.1017/9781316219317.

[4] B. Coecke, M. Sadrzadeh & S. Clark (2010): *Mathematical foundations for a compositional distributional model of meaning*. In J. van Benthem, M. Moortgat & W. Buszkowski, editors: *A Festschrift for Jim Lambek*, *Linguistic Analysis* 36, pp. 345–384. arXiv:1003.4394.

[5] B. Coecke & V. Wang (2021): *Grammar Equations*. arXiv:2106.07485.

[6] Bob Coecke (2021): *The mathematics of text structure*. In: *Joachim Lambek: The Interplay of Mathematics, Logic, and Linguistics*, Springer, pp. 181–217, doi:10.1007/978-3-030-66545-6_6.

[7] T. Duneau (2020): *Solving logical puzzles in DisCoCirc*. Available at `https://www.youtube.com/watch?v=7Mri3OXzJq4`.

[8] T. Duneau (2021): *Parsing Conjunctions in DisCoCirc*. *SemSpace 2021*, p. 66.

[9] H. Kamp & U. Reyle (2013): *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*. 42, Springer Science & Business Media, doi:10.1007/978-94-017-1616-1.

[10] B. Rodatz, R. A. Shaikh & L. Yeh (2021): *Conversational Negation using Worldly Context in Compositional Distributional Semantics*. arXiv:2105.05748.

[11] M. Sadrzadeh, S. Clark & B. Coecke (2013): *The Frobenius anatomy of word meanings I: subject and object relative pronouns*. *Journal of Logic and Computation* 23, pp. 1293–1317, doi:10.1093/logcom/ext044. arXiv:1404.5278.

[12] P. Selinger (2011): *A survey of graphical languages for monoidal categories*. In B. Coecke, editor: *New Structures for Physics*, Lecture Notes in Physics, Springer-Verlag, pp. 275–337, doi:10.1007/978-3-642-12821-9_4. arXiv:0908.3347.

[13] R. A. Shaikh, L. Yeh, B. Rodatz & B. Coecke (2021): *Composing Conversational Negation*. arXiv:2107.06820.

[14] Vincent Wang, Jonathon Liu & Bob Coecke (2022): *Distilling Text into Circuits*. *draft*.

[15] V. Wang-Mascianica & B. Coecke (2021): *Talking Space: inference from spatial linguistic meanings*. arXiv:2109.06554.