SemSpace 2021

Workshop on Semantic Spaces at the Intersection of NLP, Physics, and Cognitive Science

Proceedings of the 2021 Workshop

June 16th, 2021

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-954085-22-0

Preface

Semantic Spaces at the Intersection of NLP, Physics, and Cognitive Science (SemSpace2021) is the latest edition of a series of workshops that brings together research at the intersection of NLP, Physics, and Cognitive Science. Using the common ground of vector spaces, the workshop offers researchers in these areas an appropriate forum for presenting their uniquely motivated work and ideas. The interplay between the three disciplines fosters theoretically motivated approaches to understanding how meanings of words interact with each other in sentences and discourse via grammatical types, how they are determined by input from the world, and how word and sentence meanings interact logically.

We received 20 submissions of both extended abstracts and long papers, of which we accepted 7 extended abstracts and 8 long papers. Each paper was reviewed by at least two members of the programme committee. The submissions explored a range of topics, including the dynamics of language, grammar and parsing, quantum algorithms and phenomena in NLP, and compositional approaches for words and concepts.

Papers exploring the dynamics of language looked at how word meanings adapt in the context of a sentence (Aguirre-Celis and Miikkulainen) and at universalities in language (De las Cuevas and Sole). In the area of grammar and parsing, papers investigated various topics in pregroup grammar: efficient algorithms for pregroup parsing (Rizzo), a functorial passage from pregroup grammar to combinatory categorial grammar (Yeung and Kartsaklis), and equations within a pregroup grammar setting (Coecke and Wang).

Within the area of quantum algorithms and phenomena in NLP, papers looked at contextual phenomena in language use (Wang et al.), a functorial approach to language models (Toumi and Koziell-Pipe), quantum algorithms for wh- questions (Correia et al.), and a talk on quantum NLP - the first NLP algorithm to be run on a quantum computer (Meichanetzidis et al.)

A range of approaches to composing words and concepts were proposed. Phenomena such as conjunction (Duneau), conversational negation (Rodatz et al.), and noun phrase plausibility (McPheat et al.) have been investigated. The composition of fuzzy concepts was examined in Tull, and Hughes and Pavlovic look at the formal relation between syntax and semantics. Widdows et al provide a comprehensive survey of compositional approaches in vector space semantics, in particular looking at the link to quantum approaches.

As well as submitted talks, we have two invited talks from Ellie Pavlick (Brown University) and Haim Dubossarsky (Cambridge University).

We would like to thank everyone who submitted a paper or a talk to the workshop, all of the authors for their contributions, the programme committee for all their hard work, our invited speakers and (in advance) all the attendees. We hope for fruitful discussions and sharing of perspectives!

Workshop co-chairs

Martha Lewis, University of Bristol, UK

Mehrnoosh Sadrzadeh, University College London, UK

Organizing Committee

Martha Lewis, University of Bristol, UK

Mehrnoosh Sadrzadeh, University College London, UK

Lachlan McPheat, University College London, UK

Programme Committee

Tai-Danae Bradley, X, the Moonshot Factory, USA

Bob Coecke, Cambridge Quantum Computing, UK

Gemma De Las Cuevas, Institute for Theoretical Physics, University of Innsbruck, Austria

Stefano Gogioso, University of Oxford, UK

Peter Gärdenfors, Lund University, Sweden

Peter Hines, University of York, UK

Antonio Lieto, University of Turin, Italy

Dan Marsden, University of Oxford, UK

Michael Moortgat, Utrecht University, The Netherlands

Richard Moot, CNRS(LIRMM) & University of Montpellier, France

Dusko Pavlovic, University of Hawaii, USA

Emmanuel Pothos, City University of London, UK

Matthew Purver, Queen Mary University of London, UK

Pawel Sobocinski, Tallinn University of Technology, Estonia

Corina Stroessner, Ruhr University Bochum, Germany

Dominic Widdows, LivePerson Inc., USA

Gijs Wijnholds, Utrecht University, The Netherlands

Table of Contents

Conference Programme

16th June 2021

14:00–14:05 *Opening remarks*
Workshop Chairs

14:05–15:05 ***Invited talk: Haim Dubossarsky***

15:05–15:20 *Understanding the Semantic Space: How Word Meanings Dynamically Adapt in the Context of a Sentence*
Nora Aguirre-Celis and Risto Miikkulainen

15:20–15:30 *Universalities in natural languages*
Gemma De las Cuevas and Ricard Sole

15:30–15:40 ***Break***

15:40–15:55 *LinPP: a Python-friendly algorithm for Linear Pregroup Parsing*
Irene Rizzo

15:55–16:10 *A CCG-Based Version of the DisCoCat Framework*
Richie Yeung and Dimitri Kartsaklis

16:10–16:25 *Grammar equations*
Bob Coecke and Vincent Wang

16:25–16:35 *Functorial Language Models*
Alexis Toumi and Alex Koziell-Pipe

16:35–16:45 ***Break***

16:45–17:00 *On the Quantum-like Contextuality of Ambiguous Phrases*
Daphne Wang, Mehrnoosh Sadrzadeh, Samson Abramsky and Victor Cervantes

17:00–17:10 *QNLP: Compositional Models of Meaning on a Quantum Computer*
Konstantinos Meichanetzidis, Robin Lorenz, Anna Pearson, Alexis Toumi, Giovanni de Felice, Dimitri Kartsaklis and Bob Coecke

16th June 2021 (continued)

17:10–17:20 *Q-SAWh: a Quantum Search Algorithm to find out Whodunnit*
Adriana Correia, Michael Moortgat and H. Stoof

17:20–17:30 ***Break***

17:30–18:30 ***Invited Talk: Ellie Pavlick***

18:30–18:45 *Conversational Negation using Worldly Context in Compositional Distributional Semantics*
Benjamin Rodatz, Razin Shaikh and Lia Yeh

18:45–19:00 *Parsing conjunctions in DisCoCirc*
Tiffany Duneau

19:00–19:10 *Compositional Distributional Noun Phrase Plausibility (Abstract)*
Lachlan McPheat, Mehrnoosh Sadrzadeh and Hadi Wazni

19:10–19:20 ***Break***

19:20–19:35 *Should Semantic Vector Composition be Explicit? Can it be Linear?*
Dominic Widdows, Kristen Howell and Trevor Cohen

19:35–19:45 *Composing Fuzzy Concepts in Conceptual Spaces*
Sean Tull

19:45–19:55 *Sign as an adjunction*
Dominic Hughes and Dusko Pavlovic

19:55–20:00 *Closing remarks*
Workshop Chairs

Understanding the Semantic Space: How Word Meanings Dynamically Adapt in the Context of a Sentence

Nora Aguirre-Celis[1,2] and Risto Miikkulainen[2]

[1] ITESM, E. Garza Sada 2501, Monterrey, NL, 64840, Mexico
[2] The University of Texas in Austin, 2317 Speedway, Austin, TX, 78712 US
{naguirre,risto}@cs.utexas.edu

Abstract

How do people understand the meaning of the word *small* when used to describe a mosquito, a church, or a planet? While humans have a remarkable ability to form meanings by combining existing concepts, modeling this process is challenging. This paper addresses that challenge through CEREBRA (Context-dEpendent meaning REpresentations in the BRAin) neural network model. CEREBRA characterizes how word meanings dynamically adapt in the context of a sentence by decomposing sentence fMRI into words and words into embodied brain-based semantic features. It demonstrates that words in different contexts have different representations and the word meaning changes in a way that is meaningful to human subjects. CEREBRA's context-based representations can potentially be used to make NLP applications more human-like.

1 Introduction

The properties associated with a word such as *small* vary in context-dependent ways: It is necessary to know what the word means, but also the context in which is used, and how the words combine in order to construct the word meaning. Humans have a remarkable ability to form meanings by combining existing concepts. Modeling this process is difficult (Hampton, 1997; Janetzko 2001; Middleton et al, 2011; Murphy, 1988; Pecher et al., 2004; Sag et al., 2001, Wisniewski, 1997, 1998; Yee et al., 2016). How are concepts represented in the brain? How do word meanings change during concept combination or under the context of a sentence? What tools and approaches serve to quantify such changes?

Significant progress has been made in understanding how concepts and word meanings are represented in the brain. In particular, the first two issues are addressed by the Concept Attribute Representation theory (CAR; Binder et al., 2009, 2011, 2016a, 2016b). CAR theory represents concepts as a set of features that constitute the basic components of meaning in terms of known brain systems. It relates semantic content to systematic modulation in neuroimaging activity (fMRI patterns). It suggests that word meanings are instantiated by the weights given to different feature dimensions according to the context. The third issue is addressed by the CEREBRA or Context-dependent mEaning REpresentation in the BRAin neural network model (Aguirre-Celis & Miikkulainen, 2017, 2018, 2019, 2020a, 2020b). It is based on the CAR theory to characterize how the attribute weighting changes across contexts.

In this paper the CAR theory is first reviewed. Then, the CEREBRA model is introduced, followed by the data that provides the basis for the model. Later, experimental results are presented, showing an individual example on the concept combination effect on word meanings, how this effect applies to the entire corpus, and a behavioral analysis to evaluate the neural network model.

2 The CAR Theory

CARs (a.k.a. The Experiential attribute representation model), represent the basic components of meaning defined in terms of neural processes and brain systems. They are composed of a list of well-known modalities that correspond to specialized sensory, motor and affective brain processes, systems processing spatial, temporal, and casual information, and areas involved in social cognition. (Anderson et al., 2016, 2017,

Proceedings of the 2021 Workshop on Semantic Spaces at the Intersection of NLP, Physics, and Cognitive Science, pages 1–11
June 16th, 2021. ©2021 Association for Computational Linguistics

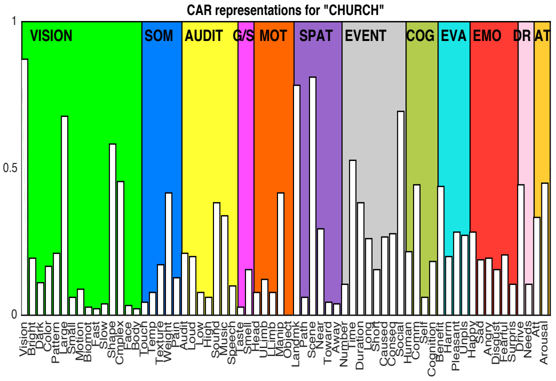


Figure 1: Bar plot of the 66 semantic features for the word *church* (Binder et al., 2009, 2011, 2016a). Given that *church* is an object, it has low weightings on animate attributes such as Face, Body, and Speech, and high weighting on attributes like Vision, Shape, and Weight. However, since it is a building for worship, it does include stronger weightings for spatial attributes such as Landmark and Scene, event attributes like Social, Time and Duration, as well as others such as Communication and Benefit. CAR weighted features for the word *church*.

2018, 2019; Binder et al. 2016a). It is supported by substantial evidence on how humans acquire and learn concepts (Binder et al., 2009, 2011, 2016a, 2016b). The central axiom of this theory is that concept knowledge is built from experience, as a result, knowledge representation in the brain is dynamic.

The features are weighted according to statistical regularities. The semantic content of a given concept is estimated from ratings provided by human participants. For example, concepts referring to things that make sounds (e.g., *explosion*, *thunder*) receive high ratings on a feature representing auditory experience, relative to things that do not make a sound (e.g., *milk*, *flower*).

Each word is modeled as a collection of 66 features that captures the strength of association between each neural attribute and word meaning. Specifically, the degree of activation of each attribute associated with the concept can be modified depending on the linguistic context, or combination of words in which the concept occurs. More detailed account of the attribute selection and definition is given by Binder, et al. (2009, 2011, 2016a, and 2016b).

Figure 1, shows an example of the weighted CARs for the concept *church*. The weight values represent average human ratings for each feature. Given that *church* is an object, it has low weightings on animate attributes such as Face, Body, and Speech, and high weighting on attributes like Vision, Size, Shape, and Weight. However, since it is a building and a place for worship, it does include strong weightings for Sound and Music, spatial attributes such as Landmark and Scene, event attributes like Social, Time and Duration, as well as others such as Communication and Benefit.

Terminology

CARWord: The neural network input. CARWords are formed based on ratings by human subjects (Section 3.3). They are the original brain-based semantic representations of words, i.e., word without context. Each CARWord is a vector of 66 attributes.

CARWordRevised: The input of the neural network after FGREP. CARWordsRevised are formed by FGREP modifying the original CARWords. They are the context-dependent meaning representations of words for each sentence where they occurred**.** Each CARWordRevised is a vector of 66 attributes.

ε : The error signal. The SynthSent is subtracted voxelwise from the fMRISent to produce an error signal. Each error is a vector of 396 changes.

fMRISent: The neural network target. They are the original brain data collected from human subjects using neuroimaging. Each fMRISent is a vector of 396 voxels.

SyntSent: The predicted fMRI sentence after training. The SynthWords in the sentence are averaged to form this prediction. Each SynthSent is a vector of 396 values.

SyntSentRevised: The modified SyntSent after applying the error signal changes. Each of these SynthSentRevised is a vector of 396 values.

SyntWord: The neural network target. They are derived by averaging the fMRISent. They are synthetic because individual fMRI data for words do not exist, thus they are obtained by averaging each fMRISent where the word occurred. Each SynthWord is a vector of 396 voxels.

SyntWordRevised: The target for the neural network after FGREP. They are derived from the SynthSentRevised using the error signal changes.

W1..W3: labels for each CARWord in a sentence.

W'1..W'3: labels for each SynthWord in a sentence.

Figure 2: Terminology for the abbreviated terms used in the CEREBRA model.

3 The CEREBRA Model

Building on the idea of grounded word representation in CAR theory, this work aims to understand how word meanings change depending on context. The following sections describe the computational model that characterizes such representations. The specific terms to the CEREBRA model are denoted by abbreviations throughout the paper (e.g., CARWord, fMRISent, SynthWord). For reference, they are described in Figure 2.

3.1 System Design

The overall design of CEREBRA is shown in Figure 3. It is a neural network model that performs two main tasks: Prediction and Interpretation. During the Prediction task, the model form a predicted fMRI for each sentence without the context effects. Each sentence is thus compared against the observed fMRI sentence to calculate an

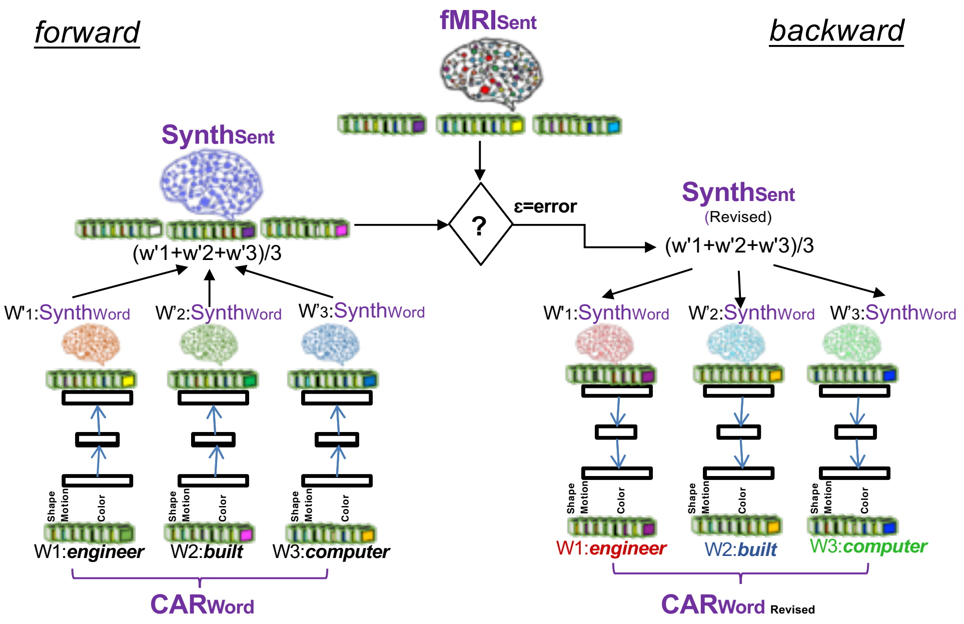


Figure 3: The CEREBRA model to account for context effects. (1) Propagate CARWords to SynthWords. (2) Construct SynthSent by averaging the SynthWords into a prediction of the sentence. (3) Compare SynthSent with the observed fMRI. (4) Backpropagate the error with FGREP for each sentence, freezing network weights and changing only CARWords. (5) Repeat until error reaches zero or CAR components reach their upper or lower limits. The modified CARs represent the word meanings in context. Thus, CEREBRA captures context effects by mapping brain-based semantic representations to fMRI sentence images.

error signal. This error signal is used repeatedly by the Interpretation task. During the Interpretation task, the model is used to determine how the CARs should adjust to eliminate the remaining error. The error is used to change the CARs themselves using the FGREP mechanism (Forming Global Representations with Extended BP, Miikkulainen & Dyer, 1991). The process iterates until the error goes to zero.

3.2 Mapping CARs to Synthetic Words

The CEREBRA model is first trained to map the CARWord representations in each sentence to SynthWords (The "forward" side of Figure 3). It uses a standard three-layer backpropagation neural network (BPNN). Gradient descent is performed for each word, changing the connection weights of the network to learn this task (Rumelharth, et al., 1986).

The BPNN was trained for each of the eleven fMRI subjects for a total of 20 repetitions each, using different random seeds. Complete training thus yields 20 different networks for each subject, resulting in 20 sets of 786 predicted SynthWord representations, that is, one word representation for each sentence where the word appears.

3.3 Sentence Prediction to Change CARs

For the Prediction task, the sentences are assembled using the predicted SynthWords by averaging all the words that occur in the sentence, yielding the prediction sentence called SynthSent. For the Interpretation task, in addition to the construction of the predicted sentence, further steps are required. First, the prediction error is calculated by subtracting the newly constructed predicted SynthSent from the original fMRISent. Then, the error is backpropagated to the inputs CARWords for each sentence (The "backward" side of Figure 3). However, following the FGREP method the weights of the network no longer change. Instead, the error is used to adjust the CARWords in order for the prediction to become accurate.

This process is performed until the prediction error is very small (near zero) or cannot be modified (CARWords already met their limits, i.e., 0 or 1), which is possible since FGREP is run separately for each sentence. These steps are repeated 20 times for each subject. At the end, the average of the 20 representations is used to represent each of the 786 context-based words (CARWord Revised), for each subject.

Eventually, the Revised CARWord represents the word meaning for the current sentence such that, when combined with other Revised CARWords in the sentence, the estimate of sentence fMRI becomes correct.

4 Data Collection and Processing

The CEREBRA model is based on the following sets of data: A sentence collection prepared by Glasgow et al. (2016), the semantic vectors (CAR ratings) for the words obtained via Mechanical Turk, and the fMRI images for the sentences, collected by the Medical College of Wisconsin (Anderson et al., 2016, 2017, 2018, 2019; Binder et al., 2016a, 2016b). Additionally, fMRI representations for individual words (called SynthWord) were synthesized by averaging the sentence fMRI.

4.1 Sentence Collection

A total of 240 sentences were composed of two to five content words from a set of 242 words (141 nouns, 39 adjectives and 62 verbs). The words were selected toward imaginable and concrete objects, actions, settings, roles, state and emotions, and events. Examples of words include *doctor*, *car*, *hospital*, *yellow*, *flood*, *damaged*, *drank*, *accident*, *summer*, *chicken*, and *family*. An example of a sentence containing some of these words is *The accident damaged the yellow car*.

4.2 Semantic Word Vectors

In a separate study Binder et al. (2016a, 2016b) collected CAR ratings for the original 242 words through Amazon Mechanical Turk. In a scale of 0-6, the participants were asked to assign the degree to which a given concept is associated with a specific type of neural component of experience (e.g. "To what degree do you think of a *church* as having a fixed location, as on a map?").

Approximately 30 ratings were collected for each word. After averaging all ratings and removing outliers, the final attributes were transformed to unit length yielding a 66-dimensional feature vector such as the one shown in Figure 1 for the word *church*. Note that this semantic feature approach builds its vector representations by mapping the conceptual content of a word (expressed in the questions) to the corresponding neural systems for which the CAR dimensions stand. This approach thus contrasts with systems where the features are extracted from text corpora and word co-occurrence with no direct association to perceptual grounding (Baroni et. al., 2010; Burgess, 1998; Harris, 1970; Landauer & Dumais, 1997; Mikolov et al., 2013).

4.3 Neural fMRI Sentence Representations

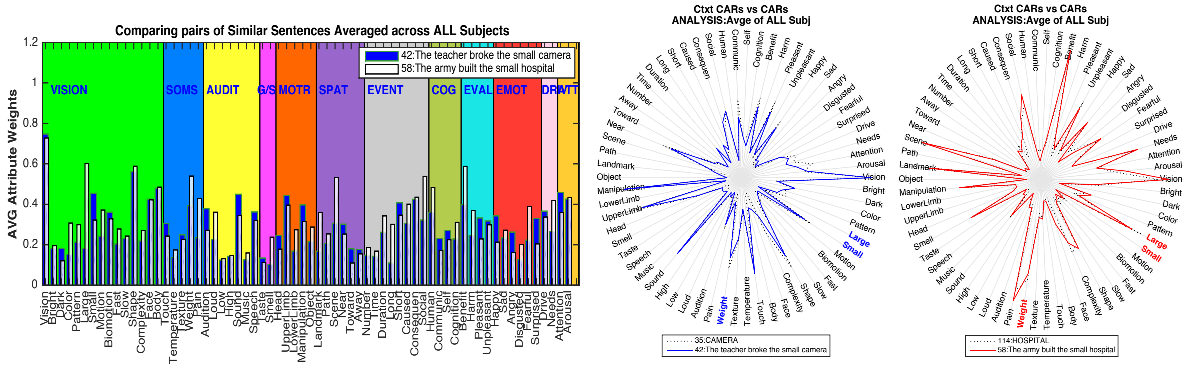
If indeed word meaning changes depending on context, it should be possible to see such changes by directly observing brain activity during word and sentence comprehension. Binder and his team collected twelve repetitions of brain imaging data from eleven subjects by recording visual, sensory, motor, affective, and other brain systems.

To obtain the neural correlates of the 240 sentences, subjects viewed each sentence on a computer screen while in the fMRI scanner. The fMRI patterns were acquired with a whole-body Three-Tesla GE 750 scanner at the Center for Imaging Research of the Medical College of Wisconsin (Anderson, et al., 2016). The sentences were presented word-by-word using a rapid serial visual presentation paradigm, with each content word exposed for 400ms followed by a 200ms inter-stimulus interval. Participants were instructed to read the sentences and think about their overall meaning.

The fMRI data were pre-processed using standard methods. The transformed brain activation patterns were converted into a single-sentence fMRI representation per participant by taking the voxel-wise mean of all repetitions (Anderson et al., 2016; Binder et al., 2016a, 2016b). To form the target for the neural network, the most significant 396 voxels per sentence were then chosen. The size selection mimics six case-role slots of content words consisting of 66 attributes each. The voxels were further scaled to [0.2..0.8].

4.4 Synthetic fMRI Word Representations

The Mapping CARs task in CEREBRA (described in Section 3.2) requires fMRI images for words in isolation. Unfortunately, the collected neural data set does not include such images. Therefore, a technique developed by Anderson et al. (2016) was adopted to approximate them. The voxel values for a word were obtained by averaging all fMRI images for the sentences where the word occurs. These vectors, called SynthWords, encode a combination of examples of that word along with other words that appear in the same sentence. Thus,



(a) Averaged sentences across subjects

(b) Averaged concepts across subjects

Figure 4: The effect of centrality on two contexts for the word *small*. (a) The average for all subjects for the two sentences. (b) The new *camera* and *hospital* representations averaged for all subjects. In the left side of the figure, the new CARs for Sentence 42 have salient activations for an object, denoting the *camera* properties like Dark, Small, Manipulation, Head, Upper Limb, Communication, and emotions such as Sad (e.g., broke the camera). The new CARs for Sentence 58, has high feature activations for large buildings describing a Large, and Heavy structure such as a *hospital*. In the right side of the figure, for each word the central attributes are highlighted to emphasize how same dimensions are more important to some concepts than others. The centrality effect correlation analysis (Medin & Shoben, 1988).

the SynthWord representation for *mouse* obtained from Sentence 56:*The mouse ran into the forest* and Sentence 60:*The man saw the dead mouse* includes aspects of running, forest, man, seeing, and dead, altogether. This process of combining contextual information is similar to several semantic models in computational linguistics (Baroni et al., 2010; Burgess, 1998; Landauer et al., 1997; Mitchell & Lapata, 2010). Additionally, in other studies, this approach has been used successfully to predict brain activation (Anderson et al., 2016, 2017, 2018, 2019; Binder, et al., 2016a, 2016b; Just, et al., 2017).

Due to the limited number of sentences, some of SynthWords became identical and were excluded from the dataset. The final collection includes 237 sentences and 236 words (138 nouns, 38 adjectives and 60 verbs). Similarly, due to noise inherent in the neural data, only eight subject fMRI patterns were used for this study.

5 Experiments

CEREBRA's context-based representations were evaluated through several computational experiments as well as through a behavioral analysis. The computational experiments quantify how the CAR representation of a word changes in different sentences for individual cases by correlating these changes to the CAR representations of the other words in the sentence (OWS). The behavioral study evaluates the CEREBRA context-based representations against human judgements.

5.1 Analysis of an Individual Example

Earlier work showed that (1) words in different contexts have different representations, and (2) these differences are determined by context. These effects were demonstrated by analyzing individual sentence cases across multiple fMRI subjects (Aguirre-Celis & Miikkulainen, 2017, 2018).

Particularly, in this example the attributes of the adjective-noun combinations are analyzed on the centrality effect for the word *small*, as expressed in Sentence 42: *The teacher broke the small camera*, and Sentence 58: *The army built the small hospital*. Centrality expresses the idea that some attributes are true to many different concepts but they are more important to some concepts than others (Medin & Shoben, 1988). For example, it is more important for boomerangs to be curved than for bananas.

Figure 4 shows the differences for *small* in these two contexts. The left side displays all 66 attributes for the two sentence representations averaged across subjects, and the right side displays the context-based representations averaged across all subjects for *camera* and *hospital*.

The size dimensions (e.g., Small and Large), demonstrated the centrality principle for these specific contexts. The left side of Figure 4 shows Sentence 42 (e.g., *small camera*) with salient activation for the central attribute Small and low

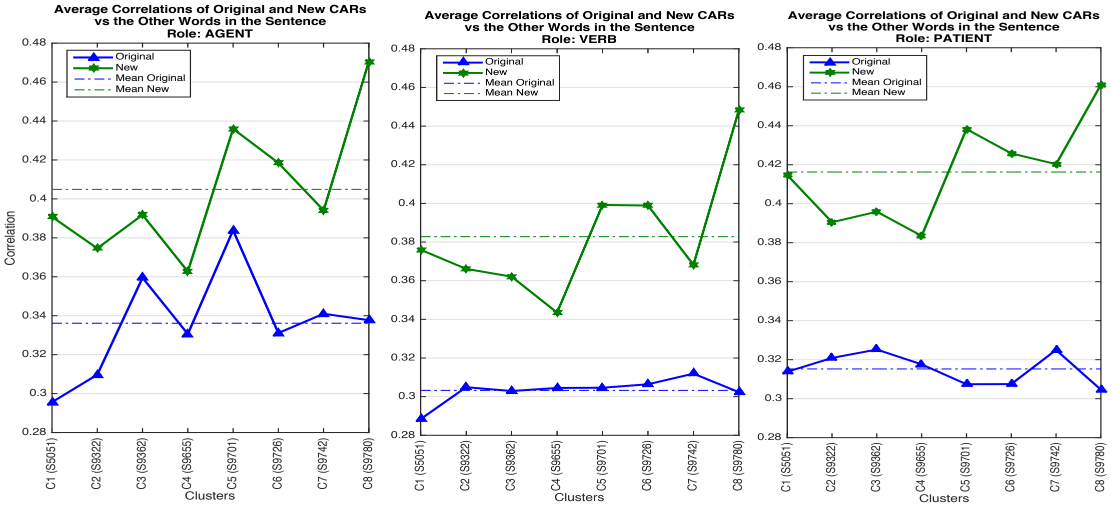


Figure 5: Correlation results per subject cluster and word roles. Average correlations analyzed by word class for eight subjects comparing original and new CARs vs. the average of the OWS respectively. A moderate to strong positive correlation was found between new CARs and the OWS, suggesting that features of one word are transferred to OWS during conceptual combination. Interestingly, the original and new patterns are most similar in the AGENT panel, suggesting that this role encodes much of the context. The results show that the effect occurs consistently across subjects and sentences.

activation for the non-central attribute Large. In contrast, Sentence 57 (e.g., *small hospital*) presents low activation on the non-central attribute Small but high activation on the central attribute Large.

These findings suggest that these attributes are essential to small objects and big structures, respectively. However, the size dimension alone cannot represent the centrality effect completely.

Additionally, given that both *camera* and *hospital* are inanimate objects, the right side of Figure 4 shows that they share low weightings on human-related attributes like Biomotion, Face, Body, and Speech. However, they also differ in expected ways, including salient activations on Darkness, Color, Small and Large size, and Weight. As part of the sentence context, the activations include human-like attributes such as Social, Human, Communication, Pleasant, Happy, Sad and Fearful. Overall, each sentence representation moves towards their respective sentence context (e.g., *camera* or *hospital*).

5.2 Aggregation Analysis

Further work verified the above conclusions in the aggregate through a statistical analysis across an entire corpus of sentences. The goal was to measure how the CARs of a word changes in different sentences, and to correlate these changes to the CARs of the other words in the sentence (OWS). In other words, the conceptual combination effect was quantified statistically across sentences and subjects (Aguirre-Celis & Miikkulainen, 2019, 2020b).

The hypothesis is based on the idea that similar sentences have a similar effect, and this effect is consistent across all words in the sentence. In order to test this hypothesis it is necessary to (1) form clusters of similar sentences for the entire collection, and (2) calculate the average changes on the words identified by the role they play for the same cluster of sentences. Through correlations, it is possible to demonstrate how similar sentences cause analogous changes in words that play identical roles in those sentences.

The results are shown in Figure 5. The correlations are significantly higher for new CARs than for the original CARs across all subjects and all roles. Furthermore, the AGENT role represents a large part of the context in both analyses (i.e., modified and original CARs). Thus, the results confirm that the conceptual combination effect occurs reliably across subjects and sentences, and it is possible to quantify it by analyzing the fMRI images using the CEREBRA model on CARs. As a summary, the average correlation was 0.3201 (stdev 0.020) for original CAR representations and 0.3918 (stdev 0.034) for new CAR representations.

Thus, this process demonstrated that changes in a target word CAR originate from the OWS. For instance, if the OWS have high values in the CAR

HUMAN RESPONSES DISTRIBUTION						
Resp/Part	P1	P2	P3	P4	AVG	%
-1	2065	995	645	1185	1223	34.0%
0	149	1120	1895	1270	1109	30.8%
1	1386	1485	1060	1145	1269	35.3%
TOT	3600	3600	3600	3600	3600	100%
PARTICIPANT AGREEMENT ANALYSIS						
	P1	P2	P3	P4	AVERAGE	%
P1	0	1726	1308	1650	1561	43%
P2	1726	0	1944	1758	1809	50%
P3	1308	1944	0	1741	1664	46%
P4	1650	1758	1741	0	1716	48%
				TOTAL	6751	
				AVG xPAR	1688	
		AVERAGE	Particip match each other			47%

(a) Human Responses

PARTICIPANTS AVERAGE AGREEMENT			
RATINGS	HUMAN	CEREBRA	CHANCE
-1	618	463	8
0	456	3	0
1	892	587	886
TOTAL	1966	1053	894
	AVERAGE	54%	45%

(b) Matching Predictions

SUBJECTS	CEREBRA		CHANCE		p-value
	MEAN	VAR	MEAN	VAR	
S5051	1033	707.25	894	6.01	3.92E-24
S9322	1035	233.91	894	7.21	6.10E-33
S9362	1063	224.41	894	11.52	5.22E-36
S9655	1077	94.79	894	7.21	3.89E-44
S9701	1048	252.79	895	12.03	1.83E-33
S9726	1048	205.82	894	4.62	1.73E-35
S9742	1075	216.77	895	7.21	1.65E-37
S9780	1039	366.06	894	2.52	6.10E-30

(c) Statistical Significance

Table 1: Comparing CEREBRA predictions with human judgements. (a) Distribution analysis and inter-rater agreement. The top table shows human judgement distribution for the three responses "less" (-1), "neutral" (0), and "more" (1). The bottom table shows percentage agreement for the four participants. Humans agree 47% of the time. (b) Matching CEREBRA predictions with human data, compared to chance baseline. The table shows the average agreement of the 20 repetitions across all subjects. CEREBRA agrees with human responses 54% while baseline is 45% - which is equivalent to always guessing "more", i.e., the largest category of human responses. (c) Statistical analysis for CEREBRA and baseline. The table shows the means and variances of CEREBRA and chance models for each subject and the p-values of the t-test, showing that the differences are highly significant. Thus, the context-dependent changes are actionable knowledge that can be used to predict human judgements.

spatial dimension for Path, then that dimension in the modified CAR should be higher than in the original CAR, for such target word. The CEREBRA model encodes this effect into the CARs where it can be measured.

5.3 Behavioral Study

While Sections 5.1 and 5.2 showed that differences in the fMRI patterns in sentence reading can be explained by context-dependent changes in the semantic feature representations of the words. The goal of this section is to show that these changes are meaningful to humans. Therefore, human judgements were compared against CEREBRA predictions (Aguirre-Celis & Miikkulainen, 2020a, 2020b).

Measuring Human Judgements: A survey was designed to characterize context-dependent changes by asking the subject directly: In this context, how does this attribute change? Human judgements were crowdsourced using Google Forms. The complete survey was an array of 24 questionnaires that included 15 sentences each. For each sentence, the survey measured 10 attribute changes for each target word. Only the top 10 statistically most significant attribute changes for each target words (roles) were used. Overall, each questionnaire thus contained 150 evaluations. The 24 questionnaires can be found at: `https://drive.google.com/drive/folders/1jDCqKMuH-SyTxcJ7oJRbr7mYV6WNNEWH?usp=sharing`

Human responses were first characterized through data distribution analysis. Table 1 (a) shows the number of answers "less" (-1), "neutral" (0), and "more" (1) for each participant. Columns labeled P1, P2, P3, and P4 show the answers of the participants. The top part of Table 1 (a) shows the distribution of the raters' responses and the bottom part shows the level of agreement among them. As can be seen from the table, the participants agreed only 47% of the time. Since the inter-rater reliability is too low, only questions that were the most reliable were included, i.e., where three out of four participants agreed. There were 1966 such questions, or 55% of the total set of questions.

Measuring CEREBRA's Predictions: The survey directly asks for the direction of change of a specific word attribute in a particular sentence, compared to the word's generic meaning. Since the changes in the CEREBRA model range within (-1,1), in principle that is exactly what the model produces. However, during the experiments it was found that some word attributes always increase, and do so more in some contexts than others. This effect is well known in conceptual combination (Hampton, 1997; Wisniewsky, 1998), contextual modulation (Barclay, 1974, Barsalou et al., 1987, 1993), and attribute centrality (Medin & Shoben, 1988). The direction of change is therefore not a good predictor of human responses.

These changes need to be measured relative to changes in the OWS. Thus, the approach was based on asking: What is the effect of CARs used in context as opposed to CARs used in isolation? This effect was measured by computing the average of the CEREBRA changes (i.e., new minus original) of the different representations of the same word in several contexts, and subtracting that average change from the change of the target word.

Matching CEREBRA's Predictions with Human Judgements: In order to demonstrate that the CEREBRA model has captured human performance, the agreements of the CEREBRA changes and human surveys need to be at least above chance. Therefore a baseline model that generated random responses from the distribution of human responses was created. The results are reported in Table 1 (b), and the statistical significance of the comparisons in Table 1 (c).

The CEREBRA model matches human responses in 54% of the questions when the baseline is 45% - which is equivalent to always guessing "more", i.e., the largest category of human responses. The differences shown in Table 1 (c) are highly statistically significant for the eight subjects. These results show that the changes in word meanings (i.e., due to sentence context observed in the fMRI and interpreted by CEREBRA) are real and meaningful to humans (Aguirre-Celis & Miikkulainen, 2020a, 2020b).

6 Discussion and Future Work

This paper described how the CAR theory, the fMRI images, and the CEREBRA model form the groundwork to characterize dynamic word meanings. The CEREBRA model generates good interpretations of word meanings considering that the dataset was limited and was not originally designed to address the dynamic effects in meaning. In future work, it would be interesting to replicate the studies on a more extensive data set. A fully balanced stimuli including sentences with identical contexts (e.g., *The yellow bird flew over the field* vs. *The yellow plane flew over the field*) and contrasting contexts (e.g., *The aggressive dog chased the boy* vs. *The friendly dog chased the boy*), could help characterize the effects in more detail. The context-based changes should be even stronger, and it should be possible to uncover more refined effects. Such data should also improve the survey design, since it would be possible to identify questions where the effects can be expected to be more reliable.

Similarly, it would be desirable to extend the fMRI data with images for individual words. The CEREBRA process of mapping semantic CARs to SynthWords and further to sentence fMRI refines the synthetic representations by removing noise. However, such representations blend together the meanings of many words in many sentences. Thus, by acquiring actual word fMRI, the observed effects should become even more clear.

One disadvantage on CEREBRA is that it is expensive to collect fMRI patterns and human ratings at a massive scale compared to running a statistical algorithm on a data repository. Furthermore, any changes to the model (e.g., adding features) would require new data to be collected. On the other hand, such data provides a grounding to neural processes and behavior that does not exist with statistical approaches.

Concept representation in the CAR approach can be compared to other methods such as Conceptual Spaces (CS; Gardenfors, 2004; Bechberger & Kuhnberger, 2019), and distributional semantic models (DSMs; Anderson et. al., 2013; Bruni et al., 2014; Burgess, 1998; Landauer & Dumais, 1997; Mikolov et al., 2013; Mitchell & Lapata, 2010; Silberer & Lapata, 2014). The CAR theory and CS characterize concepts with a list of features or dimensions as the building blocks. The CAR theory provides a set of primitive features for the analysis of conceptual content in terms of neural processes (grounded in perception and action). The CS framework suggests a set of "quality" dimensions as relations that represent cognitive similarities between stimuli (observations or instances of concepts). CS is also considered a grounding mechanism that connects abstract symbols to the real world. The CAR and CS approaches include similar dimensions (i.e., weight, temperature, brightness) and some of those dimensions are part of a larger domain (e.g., color) or a process (e.g., visual system). Whereas CAR theory is a brain-based semantic representation where people weigh concept dimensions differently based in context, DSMs are not grounded on perception and motor mechanisms. Instead, DSM representations reflect semantic knowledge acquired through a lifetime of linguistic experience based on statistical co-occurrence. DSMs do not provide precise information about the experienced features of the concept itself (Anderson et al., 2016). In CEREBRA, this grounding provides a multimodal approach where features directly relate semantic content to neural activity.

7 Conclusions

The CEREBRA model was constructed to test the hypothesis that word meanings change dynamically based on context. The results suggest three significant conclusions: (1) context-dependent meaning representations are embedded in the fMRI sentences, (2) they can be characterized using CARs together with the CEREBRA model, and (3) the attribute weighting changes are real and meaningful to human subjects. Thus, CEREBRA opens the door for cognitive scientists to achieve better understanding and form new hypotheses about how semantic knowledge is represented in the brain. Additionally, the context-based representations produced by the model could be used for a broad range of artificial natural language processing systems, where grounding concepts as well as understanding novel combinations of concepts is critical.

Acknowledgments

We would like to thank J. Binder (Wisconsin), R. Raizada and A. Anderson (Rochester), M. Aguilar and P. Connolly (Teledyne) for providing this data and for their valuable help regarding this research. This work was supported in part by IARPA-FA8650-14-C-7357 and by NIH 1U01DC014922 grants.

References

Nora Aguirre-Celis & Risto Miikkulainen. (2017). From Words to Sentences & Back: Characterizing Context-dependent Meaning Representations in the Brain. *Proceedings of the 39th Annual Meeting of the Cognitive Science Society*, London, UK, pp. 1513-1518.

Nora Aguirre-Celis & Risto Miikkulainen. (2018) Combining fMRI Data and Neural Networks to Quantify Contextual Effects in the Brain. In: Wang S. et al. (Eds.). *Brain Informatics*. BI 2018. Lecture Notes in Computer Science. 11309, pp. 129-140. Springer, Cham.

Nora Aguirre-Celis & Risto Miikkulainen. (2019). Quantifying the Conceptual Combination Effect on Words Meanings. *Proceedings of the 41th Annual Conference of the Cognitive Science Society*, Montreal, CA. 1324-1331.

Nora Aguirre-Celis & Risto Miikkulainen. (2020a). Characterizing the Effect of Sentence Context on Word Meanings: Mapping Brain to Behavior. *Computation and Language*. arXiv:2007.13840.

Nora Aguirre-Celis & Risto Miikkulainen. (2020b). Characterizing Dynamic Word Meaning Representations in the Brain. *In Proceedings of the 6th Workshop on Cognitive Aspects of the Lexicon (CogALex-VI)*, Barcelona, ES, December 2020.

Andrew J. Anderson, Elia Bruni, Ulisse Bordignon, Massimo Poesio, and Marco Baroni. 2013. Of words, eyes and brains: Correlating image-based distributional semantic models with neural representations of concepts. *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (EMNLP 2013); Seattle, WA: Association for Computational Linguistics. pp. 1960–1970.

Andrew J. Anderson, Jeffrey R. Binder, Leonardo Fernandino, Colin J. Humphries, Lisa L. Conant, Mario Aguilar, Xixi Wang, Donias Doko, Rajeev D S Raizada. 2016. Perdicting Neural activity patterns associated with sentences using neurobiologically motivated model of semantic representation. *Cerebral Cortex*, pp. 1-17. DOI:10.1093/cercor/bhw240

Andrew J. Anderson, Douwe Kiela, Stephen Clark, and Massimo Poesio. 2017. Visually Grounded and Textual Semantic Models Differentially Decode Brain Activity Associated with Concrete and Abstract Nouns. *Transaction of the Association for Computational Linguistics* 5: 17-30.

Andrew J. Anderson, Edmund C. Lalor, Feng Lin, Jeffrey R. Binder, Leonardo Fernandino, Colin J. Humphries, Lisa L. Conant, Rajeev D.S. Raizada, Scott Grimm, and Xixi Wang. 2018. Multiple Regions of a Cortical Network Commonly Encode the Meaning of Words in Multiple Grammatical Positions of Read Sentences. *Cerebral Cortex*, pp. 1-16. DOI:10.1093/cercor/bhy110.

Andrew J. Anderson, Jeffrey R. Binder, Leonardo Fernandino, Colin J. Humphries, Lisa L. Conant, Rajeev D.S. Raizada, Feng Lin, and Edmund C. Lalor. 2019. An integrated neural decoder of linguistic and experiential meaning. *The Journal of neuroscience: the official journal of the Society for Neuroscience.*

Richard Barclay, John D. Bransford, Jeffery J. Franks, Nancy S. McCarrell, & Kathy Nitsch. 1974.

Comprehension and semantic flexibility. *Journal of Verbal Learning and Verbal Behavior, 13*:471–481.

Marco Baroni, Brian Murphy, Eduard Barbu, and Massimo Poesio. 2010. Strudel: A Corpus-Based Semantic Model Based on Properties and Types. *Cognitive Science,* 34(2):222-254.

Lawrence W. Barsalou. 1987. The instability of graded structure: Implications for the nature of concepts. In U. Neisser (Ed.), *Concepts and conceptual development: Ecological and intellectual factors in categorization.* Cambridge, England: Cambridge University Press.

Lawrence W. Barsalou, Wenchi Yeh, Barbara J. Luka, Karen L. Olseth, Kelly S. Mix, Ling-Ling Wu. 1993. Concepts and Meaning. *Chicago Linguistic Society 29: Papers From the Parasession on Conceptual Representations*, 23-61. University of Chicago.

Lucas Bechberger, Kai-Uwe Kuhnberger. 2019. A Thorough Formalization of Conceptual Spaces. In: Kern-Isberner, G., Furnkranz, J., Thimm, M. (eds.) KI 2017: *Advances in Artificial Intelligence: 40th Annual German Conference on AI*, Dortmund, Germany.

Jeffrey R. Binder and Rutvik H. Desai, William W. Graves, Lisa L. Conant. 2009. Where is the semantic system? A critical review of 120 neuroimaging studies. *Cerebral Cortex*, 19:2767-2769.

Jeffrey R. Binder and Rutvik H. Desai. 2011. The neurobiology of semantic memory. *Trends Cognitive Sciences*, 15(11):527-536.

Jeffrey R. Binder. 2016a. In defense of abstract conceptual representations. *Psychonomic Bulletin & Review*, 23. doi:10.3758/s13423-015-0909-1

Jeffrey R. Binder, Lisa L. Conant, Colin J. Humpries, Leonardo Fernandino, Stephen B. Simons, Mario Aguilar, Rutvik H. Desai. 2016b. Toward a brain-based Componential Semantic Representation. *Cognitive Neuropsychology*, 33(3-4):130-174.

Elia Bruni, Nam Khanh Tran, Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research (JAIR),* 49:1-47.

Curt Burgess. 1998. From simple associations to the building blocks of language: Modeling meaning with HAL. *Behavior Research Methods, Instruments, & Computers*, 30:188–198.

Peter Gardenfors. 2004. Conceptual spaces: The geometry of thought, The MIT Press.

Kimberly Glasgow, Matthew Roos, Amy J. Haufler, Mark Chevillet, Michael Wolmetz. 2016. Evaluating semantic models with word-sentence relatedness. *Computing Research Repository*, arXiv:1603.07253.

James Hampton. 1997. Conceptual combination. In K. Lamberts & D. R. Shanks (Eds.), *Studies in cognition. Knowledge, concepts and categories*, 133–159. MIT Press.

Zellig Harris. 1970. Distributional Structure. *In Papers in Structure and Transformational Linguistics,* 775-794.

Dietmar Janetzko. 2001. Conceptual Combination as Theory Formation. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 23.

Marcel A. Just, Jing Wang, Vladimir L. Cherkassky. 2017. Neural representations of the concepts in simple sentences: concept activation prediction and context effect. Neuroimage, 157:511–520.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory. *Psychological Review*, 104:211-240.

Douglas L. Medin and Edward J. Shoben. 1988. Context and structure in conceptual combination. *Cognitive Psychology*, 20:158-190.

Erica L. Middleton, Katherine A. Rawson, and Edward J. Wisniewski. 2011. "How do we process novel conceptual combinations in context?". *Quarterly Journal of Experimental Psychology.* **64** (4): 807–822.

Risto Miikkulainen and Michael Dyer. 1991. Natural Language Processing with Modular PDP Networks and Distributed Lexicon. Cognitive Science, 15: 343-399.

Thomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 3111–3119.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 38(8):1388–1439. DOI: 10.1111/j.1551-6709.2010.01106.x

Gregory Murphy. 1988. Comprehending complex concepts. *Cognitive Science*, 12: 529-562.

Diane Pecher, Rene Zeelenberg, and Lawrence Barsalou. 2004. Sensorimotor simulations underlie conceptual representations Modality-specific effects of prior activation. *Psychonomic Bulletin & Review*, 11: 164-167.

David E. Rumelhart, James L. McClelland, and PDP Research Group (1986) *Parallel Distributed Processing. Explorations in the Microstructure of Cognition*, Volume 1: Foundations. Cambridge, MA: MIT Press.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann Copestake, Dan Flickinger. 2001. Multiword expressions: A pain in the neck for NLP. *In*

International conference on intelligent text processing and computational linguistics, 1-15. Springer, Berlin, Heidelberg.

Carina Silberer and Mirella Lapata. 2014. Learning Grounded Meaning Representations with Autoencoders. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 721-732.

Edward J. Wisniewski. 1997. When concepts combine. Psychonomic Bulletin & Review, 4, 167–183.

Edward J. Wisniewski. 1998. Property Instantiation in Conceptual Combination. *Memory & Cognition*, 26, 1330-1347.

Eiling Yee, & Sharon L. Thompson-Schill. 2016. Putting concepts into context. *Psychonomic Bulletin & Review*, 23, 1015–1027.

LinPP: a Python-friendly algorithm for Linear Pregroup Parsing

Irene Rizzo
University of Oxford / Oxford
irene.rizzo@cs.ox.ac.uk

Abstract

We define a linear pregroup parser, by applying some key modifications to the *minimal parser* defined in (Preller, 2007a). These include handling words as separate blocks, and thus respecting their syntactic role in the sentence. We prove correctness of our algorithm with respect to parsing sentences in a subclass of pregroup grammars. The algorithm was specifically designed for a seamless implementation in Python. This facilitates its integration within the *DisCopy* module for QNLP and vastly increases the applicability of pregroup grammars to parsing real-world text data.

1 Introduction

Pregroup grammars (PG), firstly introduced by J. Lambek in (Lambek, 1997), are becoming popular tools for modelling syntactic structures of natural language. In compositional models of meaning, such as *DisCoCat* (Coecke et al., 2010) and *DisCoCirc* (Coecke, 2019), grammatical composition is used to build sentence meanings from words meanings. Pregroup types mediate this composition by indicating how words connect to each other, according to their grammatical role in the sentence. In DisCoCat compositional sentence embeddings are represented diagrammatically; these are used as a language model for QNLP, by translating diagrams into quantum circuits via the Z-X formalism (Zeng and Coecke, 2016; Coecke et al., 2020a,b; Meichanetzidis et al., 2020b,a). *DisCopy*, a Python implementation of most elements of DisCoCat, is due to Giovanni Defelice, Alexis Toumi and Bob Coecke (Defelice et al., 2020).

An essential ingredient for a full implementation of the DisCoCat model, as well as any syntactic model based on pregroups, is a correct and efficient pregroup parser. Pregroup grammars are weakly equivalent to context-free grammars (Buszkowski, 2009). Thus, general pregroup parsers based on this equivalence are poly-time, see e.g. (Earley, 1970). Examples of cubic pregroup parsers exist by Preller (Degeilh and Preller, 2005) and Moroz (Moroz, 2009b) (Moroz, 2009a). The latter have been implemented in Python and Java. A faster *Minimal Parsing* algorithm, with linear computational time, was theorised by Anne Preller in (Preller, 2007a). This parser is correct for the subclass of pregroup grammars characterised by *guarded dictionaries*. The notion of *guarded* is defined by Preller to identify dictionaries, whose *criticalities* satisfy certain properties (Preller, 2007a). In this paper we define *LinPP*, a new linear pregroup parser, obtained generalising Preller's definition of guards and applying some key modifications to the Minimal Parsing algorithm. *LinPP* was specifically designed with the aim of a Python implementation. Such implementation is currently being integrated in the DisCopy package (github:*oxford-quantum-group/discopy*).

The need for a linear pregroup parser originated from the goal of constructing a grammar inference machine learning model for pregroup types, i.e. a Pregroup Tagger. training and evaluation of such model is likely to involve parsing of several thousand sentences. Thus, LinPP will positively affect the overall efficiency and performance of the Tagger. The Tagger will enable us to process real world data and test the DisCoCat pregroup model against the state-of-the-art with respect to extensive tasks involving real-world language data.

2 Pregroup Grammars

We recall the concepts of *monoid*, *preordered monoid* and *pregroup*.

Definition 2.1. A monoid $\langle P, \bullet, 1\rangle$ is a set P together with binary operation $\bullet$ and an element 1, such that

$$(x \bullet y) \bullet z = x \bullet (y \bullet z) \tag{1}$$

$$x \bullet 1 = x = 1 \bullet x \tag{2}$$

for any $x, y \in P$. We refer to $\bullet$ as *monoidal product*, and we often omit it, by simply writing xy in place of of $x \bullet y$.

Definition 2.2. A preordered monoid is a monoid together with a *reflexive transitive* relation $P \to P$ such that:

$$x \to y \implies uxv \to uyv \tag{3}$$

Definition 2.3. A **pregroup** is a preordered monoid $\langle P, \bullet, 1 \rangle$, in which every object x has a left and a right *adjoint*, respectively written as x^l and x^r, such that:

$$\text{contraction rules} \quad x^l x \to 1; \quad xx^r \to 1$$
$$\text{expansion rules} \quad 1 \to x^r x; \quad 1 \to xx^l$$

Adjoints are unique for each object.

In the context of natural language, pregroups are used to model grammatical types. This approach was pioneered by J. Lambek, who introduced the notion of *Pregroup Grammars* (Lambek, 1997). These grammars are constructed over a set of *basic types*, which represent basic grammatical roles. For example, $\{n, s\}$ is a set consisting of the *noun* type and the *sentence* type.

Definition 2.4. Let B be a set of *basic* types. The free pregroup over B, written P_B, is the free pregroup generated by the set $B \cup \Sigma$, where Σ is the set of iterated adjonts of the types in B.

In order to easily write iterated adjoints, we define the following notation.

Definition 2.5. Given a basic type t, we write t^{l^n} to indicate its n-fold left adjoint, and t^{r^n} for its n-fold right adjoint.E.g. we write t^{l^2} to indicate $(t^l)^l$.

Thanks to the uniqueness of pregroup adjoints we can mix the right and left notation. E.g. $(t^{r^2})^l$ is simply t^r. We write $t^{l^0} = t = t^{r^0}$. We now define pregroup grammars, following the notation of (Shiebler et al., 2020).

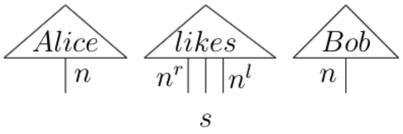
Definition 2.6. A **pregroup grammar** is a tuple $PG = \{V, B, \mathbb{D}, P_B, s\}$ where:

1. V is the *vocabulary*, i.e. a set of words.
2. B is a set of basic grammatical types.
3. P_B is the free pregroup over B.
4. $\mathbb{D} \subset V \times P_B$ is the *dictionary*, i.e it contains correspondences between words and their assigned grammatical types.
5. $s \in P_B$ is a basic type indicating the *sentence type*.

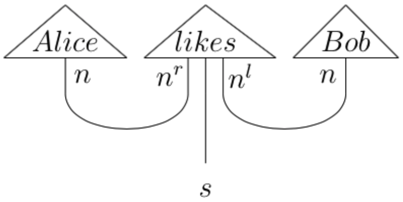
Example 2.7. Consider the grammar given by $V = \{Alice, loves, Bob\}$, $B = \{n, s\}$ and a dictionary with the following type assignments:

$$\mathbb{D} = \{(Alice, n), (Bob, n), (loves, n^r s n^l)\}$$

Note that the grammatical types for *Alice* and *Bob* are so-called *simple* types, i.e basic types or their adjoints. On the other hand, the type of the transitive verb is a monoidal product. The type of this verb encodes the recipe for creating a sentence: it says *give me a noun type on the left and a noun type on the right and I will output a sentence type*. In other words, by applying iterated contraction rules on $nn^r sn^l n$ we obtain the type s. Diagrammatically we represent the string as



Then, after applying the contraction rules, we obtain a sentence diagram:



This diagram is used to embed the sentence meaning. This framework - introduced by Coecke et al. in 2010 - is referred to as *DisCoCat* and provides a mean to equip distributional semantics with compositionality. The composition is mediated by the sentence's pregroup contractions, as seen in the example above. (Coecke et al., 2010).

The iterated application of contraction rules yields a *reduction*.

Definition 2.8. Let $S := t_1....t_n$ be a string of simple types, and let $T_S := t_{j_1}....t_{j_p}$ with $j_i \in [1, n]$ for all i. We say that $R : S \to T_S$ is a

reduction if R is obtained by iterating contraction rules only. We say that T_S is a reduced form of S. If T_S cannot be contracted any further, we say that it is **irreducible** and we often write $R : S \implies T_S$. Note that neither reductions nor irreducible forms are unique, as often we are presented with different options on where to apply contraction rules.

In the context of pregroup grammars, we are interested in reducing strings to the sentence type s, whenever this is possible. Thus, we give such reduction a special name (Shiebler et al., 2020):

Definition 2.9. a reduction $R : S \implies T_S$ is called a **parsing** of S, if T_s is the simple type s. A string S is a *sentence* if there exists a parsing.

Often, we want to keep track of the types as they get parsed:

Definition 2.10. The **set of reductions** of $R : S \to T_S$ is a set containing index pairs $\{i, j\}$ such that $t_i t_j$ is the domain of a contraction in R. These pairs are referred to as *underlinks*, or *links* (Preller, 2007a).

3 Linear vs critical

We now discuss critical and linear types in a pregroup grammar. We first need to introduce the notion of *complexity* (Preller, 2007a, Definition 5) [Preller].

Definition 3.1. A pregoup grammar with dictionary $\mathbb{D}$ has **complexity** k if, for every type $t \in \mathbb{D}$, any left (right) adjoint t^{l^n} (t^{r^n}) in $\mathbb{D}$ is such that $n < k$.

Complexity 1 indicates a trivial grammar that contains only basic types (no adjoints). Complexity 2 allows for dictionaries containing at most basic types and their 1-fold left and right adjoints, e.g. n^l and n^r. As proven in (Preller, 2007b), every pregroup grammar is *strongly equivalent* to a pregroup grammar with complexity 2. This means that the subclass of complexity 2 pregroup grammars has the same expressive power of the whole class of pregroup grammars.

We now introduce critical types (Preller, 2007a).

Definition 3.2. A type c is **critical**, if there exists types $a, b \in \mathbb{D}$ such that $ab \to 1$ and $bc \to 1$. A type is *linear* if it is not critical.

We say that a grammar is linear if all types in the dictionary are linear types. Given a string from a linear grammar, its reduction links are unique (Preller, 2007a, Lemma 7). In fact, a very simple algorithm can be used to determine whether a linear string is a sentence or not.

3.1 Lazy Parsing

The Lazy Parsing algorithm produces parsing for all linear sentences.

Definition 3.3. Consider a linear string S. Let St be an initially empty stack, and R an initially empty set of reductions. The Lazy Parsing algorithm reduces the string as follows:

1. The first type in S is read and added to St.

2. Any following type t_n is read. Letting t_i indicate the top of the stack St up until then, if $t_i t_n \to 1$ then St is popped and the link is added to R. Otherwise t_n is added to St and R remains unchanged.

By (Preller, 2007a, Lemma8, Lemma9) Lazy Parsing reduces a linear string to its *unique* irreducible form, thus a linear string is a sentence if and only if the Lazy Parsing reduces it to s. Unfortunately linear pregroup grammars do not hold a lot of expressive power, and criticalities are immediately encountered when processing slightly more complex sentences than 'subject + verb + object'. Thus, defining parsing algorithms that can parse a larger class of pregroup grammars becomes essential.

3.2 Guards

In order to discuss new parsing algorithms in the next sections, we introduce some useful notions.

Definition 3.4. Given a reduction R, a subset of *nested* links is a called a **fan** if the right endpoints of the links form a segment in the string. A fan is critical if the right endpoints are critical types (Preller, 2007a).

Below, we define *guards*, reformulating the notion introduced by Preller in (Preller, 2007a).

Definition 3.5. Let us consider a string $S := t_1 .... t_b = X t_p Y$, containing a critical type t_p. Let S reduce to 1. We say that t_b is a **guard** of t_p in S if the following conditions are satisfied:

1. X contains only linear types and there exists a reduction $R : X \implies 1$.

2. There exists a link $\{j, k\}$ of R such that $t_k t_p \to 1$ and $t_j t_b \to 1$ are contractions.

3. There exist subreductions $R1, R_2 \subset R$ such that $R_1 : t_{k+1}..t_{p-1} \implies 1$ and $R_2 : t_1...t_{j-1} \implies 1$.

4. There exists a reduction $R_y : Y \implies t_b$.

If such guard exists, we say that the critical type is *guarded* and we say that $\{j, b\}$ is a *guarding link* for the critical type.

Let us adapt this definition to critical fans.

Definition 3.6. Let us consider the segment $S := t_1.....t_n = XT_cY$. Let us assume there exists a reduction $S \implies 1$, that contains a critical fan with right end points $t_p....t_{p+q} =: T_c$. We say that the fan is guarded in S if:

1. X is linear and there exists a reduction $R : X \implies 1$.

2. There exist links $\{j_i, k_i\} \in R$, for $i \in [p, p+q]$, with $k_p > ... > k_{p+q}$, $j_p < ... < j_{p+q}$ and $t_{k_{p+q}}...t_{k_p}T_c \implies 1$.

3. The segments $t_1...t_{j_p}$ and $t_{k_p+1}..t_{p-1}$, as well as the ones in between each t_k or t_j and the next ones, have reductions to 1.

4. There exists a reduction $R_y : Y \implies T_c^l$.

3.3 Critical types in complexity 2 grammars

Critical types are particularly well behaved in dictionaries of complexity 2, as they are exactly the right adjoints t^r of basic types t. We recall the following results from (Preller, 2007a, Lemma 17 & 18). We assume complexity 2 throughout.

Lemma 3.7. Let $R : t_1...t_m \implies 1$. Let t_p be the leftmost critical type in the string and let R link $\{k, p\}$. Let t_i be the top of the stack produced by Lazy Parsing, then $i \leq k$. Moreover, if $k > i$, there are j, b with $i < j < k$ and $b > p$, such that Lazy Parsing links $\{j, k\}$ and R links $\{j, b\}$.

Corollary 3.8. *Let t_p be the leftmost critical type of a sentence S. With i as above, if t_it_p reduce to the empty type, then all reductions to type s will link $\{i, p\}$.*

We prove the following result.

Lemma 3.9. Let $S := s_1....s_n$ be a string with $m \geq 2$ critical types. Let them all be guarded. Let s_p be a critical type, and let s_q be the next one. Let s_{b_p} and s_{b_q} be their guards respectively. Assume the notation of the previous definitions. Then, either $j_q > p$ and $b_q < b_p$, or $j_q > b_p$.

Proof. By assumption, s_p is guarded, and by definition of guard, the segment $s_{p+1}....s_{b_p-1}$ reduces to the empty type. For the sake of contradiction, assume $j_q < p$. Then, because crossings are not allowed, we must have $j_q < k_p$. Since j_q is a left adjoint of a basic type, it can only reduce on its right, and we have $k_q < k_p$. However, the segment Y_p does contain s_q, and does not contain its reduction s_{k_q}, thus Y_p cannot reduce to type s_{b_p}, which is a contradiction. Thus $j_q > p$, and to avoid crossings, it is either $j_q > p$ and $b_q < b_p$ or $j_q > b_p$. □

The lemmas above also hold for guarded critical fans.

4 $MinPP$: Minimal Parsing Algorithm

In this section we define $MiniPP$, a minimal parsing algorithm complexity 2 pregroup grammars.

MinPP pseudo-code. Let $sentence : t_1....t_m$ be a string of types from a dictionary with complexity 2. We associate each processing step of the algorithm with a stage S_n. Let S_0 be the initial stage, and $S_n := \{a, n\}$ with $n \geq 1$ be the stage processing the type a in position n. Let R_n and St_n be respectively the set of reductions and the reduced stack at stage S_n. Let us write $\top(St_n)$ for the function returning the top element of the stack at stage n and $pop(St_n)$ for the function popping the stack. The steps of the algorithms are defined as follows. At stage S_0, we have $R_0 = \emptyset$ and $St_0 = \emptyset$. At stage S_1, $R_1 = \emptyset$ and $St_1 = t_1$. At stages S_n, $n > 1$, let $t_i = \top(St_{n-1})$. We define the following cases.

- If $t_it_n \to 1$:

$$St_n = pop(St_{n-1})$$
$$R_n = R_{n-1} \cup \{i, n\}$$

- Elif t_n is linear:

$$St_n = St_{n-1} + t_n$$
$$R_n = R_{n-1}$$

- Else (t_n is **critical**):

  1. **while** types are critical read $sentence$ forward starting from t_n and store read types. Let $T^r := t_n...t_{n+v}$, $v \geq 0$, be the segment of stored types.

2. Create a new empty stack St_{back}. Process *sentence* backward, starting from T^r and not reading further than t_{i+1}.
3. If St_{back} is never found empty, set $St_n = St_{n-1} + T^r$, $R_n = R_{n-1}$ and move to stage S_{n+v+1} i.e. the first type after the critical fan. If instead St_{back} becomes empty, proceed as follows.
4. St_{back} being empty means that T^r was reduced with some types T. By construction, T had been initially reduced with some types T^l by the forward process. Set $St_n = St_{n-1} + T^l$. Write $R_{T_{prec}}$ for the set of links that originally reduced $T^l T$. Write R_T for the set of links for the TT^r reduction, as found by the backward process. Set $R_n = (R_{n-1} \cup R_T)/R_{T_{prec}}$. Move to the next stage.

4.1 Formal Verification

In this section we prove the correctness of $MinPP$ with respect to reducing strings to an irreducible form, given some restrictions on the grammar. First we prove that $MinPP$ is a **sound** and **terminating** parsing algorithm for complexity 2 pregroup grammars. Then, we prove that it is also **correct** with respect to a subclass of complexity 2 pregroup grammars identified bt certain restrictions.

Theorem 4.1. *Let str be a string of types from a complexity 2 pregroup grammar. If we feed str to $MinPP$, then:*

1. ***Termination**: $MinPP$ eventually halts.*
2. ***Completeness**: If str is a sentence, $MinPP$ reduces str to sentence type s.*
3. ***Soundness**: If str is not a sentence, then $MinPP$ will reduce it to an irreducible form different from s.*

Proof. Let t_i always indicate the top of the stack.
Termination. Let us consider strings of finitely many types. We prove that at each stage the computation is finite, and that there are a finite number of stages. A stage S_n is completed once its corresponding stack St_n and set of reductions R_n is computed. If t_n is linear, updating St_n and R_n only involves two finite computations: checking whether $t_i t_n \to 1$ (done via a terminating truth value function), and popping or adding t_n to the stack. In the case of t_n being critical, if $t_i t_n \to 1$, this is handled like in the linear case. Else, the following computations are involved: first, the algorithm will read forward to identify a critical fan. This will halt when either reading the last critical type of the fan, or the last type in the string. Then the string is processed backward. This computation involves finite steps as in the forward case, and halts when reading t_i or the first type in the string, or when the stack is empty. The next computations involve updating the stack and reduction sets via finite functions. This proves that each step of the process is finite and that $MinPP$ terminates.

Soundness. We prove it by induction on the number of critical fans.
Base Case
Consider a string with one critical fan with right endpoints T^r, and assume it is not a sentence. The case in which the fan reduces with the stack is trivial, so we assume otherwise. We have two cases:
# 1: Let T^r have a left reduction T. Assuming the notation above, consider segments $\theta_{prec}, \theta, \theta_{post}$. θ reduces to the empty type. So we must have $\theta_{prec}\theta_{post} \to C$, with $C \neq s$. Since this string is linear $MinPP$ will reduce the full string to C.
# 2: Assume T^r doesn't have a left reduction. Then the backward stack will not become empty, and once the backward parsing will reach t_i, $MinPP$ will add T^r to the forward stack. At this stage, the remaining string will be $CT^r D$ with C possibly empty. D is linear and cannot contain right reductions for T^r since the complexity is 2. Thus $MinPP$ will reduce it by Lazy Parsing to its unique irreducible form $T^r U \neq s$.
Inductive Hypothesis
Assume that $MinPP$ reduces any non-sentences to to an irreducible form different from s, given that the string has no more than m critical fans.
Inductive Step
We consider a string with $m + 1$ critical fans, and no reduction to the sentence type.
# 1: Assume the notation above and let T^r have left reductions. Then, we remove the segment θ. $MinPP : \theta \implies 1$. The remaining string has m critical fans and no reduction to sentence type, so by induction hypothesis, $MinPP$ won't reduce it to the sentence type.
# 2: Assume that T^r has no left adjoints in the string. Then, $MinPP$ will add T^r to the to the top of the forward stack. The remaining string

to process is CT^rD, with C linear, irreducible and possibly empty, and D containing m critical fans. Thus, $MinPP$ will correctly parse D to its irreducible form, by inductive hypothesis or by proof of completeness (depending on whether D has a reduction to s or not). Therefore $MinPP$ will reduce CT^rD to an irreducible form, that must be different from s since T^r cannot contain s and cannot reduce further. □

In order to prove completeness we need to restrict our grammars further.

Theorem 4.2. *Let str be a string of types from a complexity 2 pregroup grammar. Let also assume that all critical fans are guarded or their critical types contract with the top of the stack of the corresponding stages. If we feed str to $MinPP$, then:* ***(Completeness)*** *If str is a sentence, $MinPP$ reduces str to sentence type s.*

Proof. We prove it by induction on the number of critical fans.

Base case

Let us consider a sentence with one critical fan, with right-end points $T^r := t_p....t_{p+n}$. At stage S_p, we have two cases: **# 1:** Let $t_it_p \to 1$. Then, by 3.8 all reductions of the string to the sentence type will link i, p. Since links cannot cross, we have $k_q < i$ for all q. Thus all critical types are linked to types in the stack. Thus, their links are unique and will be reduced by Lazy Parsing. By assumption, all types other than the critical fan are linear, thus their links are unique. Thus, Lazy Parsing will correctly reduce this sentence, and, by construction, so will $MinPP$.

# 2: Let $t_it_p \not\to 1$, let R be an arbitrary reduction of the string to sentence type. Then, by 3.7, R links each critical type t_q on the left with some t_{k_q}, such that $i < k_q < p$. Moreover, since the fan is guarded, the backward stack will become empty when the type $t_{k_{p+n}}$ is read. At this point, the segment $T^l := t_{j_p}....t_{j_{p+n}}$ is added to the forward stack. The remaining reductions are linear and T^l will be linearly reduced by Lazy parsing, since the fan is guarded. Thus, $MinPP$ will correctly reduce this string to the sentence type.

Inductive Hypothesis

Assume $MinPP$ parses any sentence with at most m guarded critical fans.

Inductive Step

Consider a string with $m+1$ guarded critical types. Consider the leftmost critical fan, and write $T^r := t_p...t_{p+n}$ for the segment given by its right end points. Let R be a reduction of the string to the sentence type. We have again two cases:

# 1: Let R reduce T^r with T in the top of the stack computed by Lazy Parsing. $MinPP$ will reduce $TT^r \to 1$ by lazy parsing. After this stage, consider the string P obtained by appending the remaining unprocessed string to St. P contains m critical fans and reduces to sentence type, thus, by inductive hypothesis, $MinPP$ will parse it.

# 2: Assume T^r does not reduce with types in the stack. Let $T := t^l_{p+n}...t^l_p$ be the types in the string which are reduced with T^r. Their index must be larger than i. Write $\theta := t^l_{p+n}...T^r$. Write θ_{prec} for the segment preceding θ, and θ_{post} for the segment following θ. θ_{prec} is linear, so its irreducible form D is unique. Moreover, by construction, we must have $D = CT^l$. Then $MinPP : \theta_{prec} \implies CT^l$ by Lazy Parsing. Since T^r is guarded, the backward stack will eventually be empty and $MinPP : \theta_{prec}\theta \implies CT^l$. The remaining string $CT^l\theta_{post}$ contains m guarded critical types and, since T^r is guarded, it has a reduction to sentence type. By inductive hypothesis, $MinPP : C\theta_{post} \implies s$.

□

Note that this proves that $MinPP$ is **correct** for the class of complexity 2 pregroup grammars identified by the above restrictions on the critical fans. We recall that complexity 2 grammars hold the same expressive power of the whole class of pregroup grammars. We now verify that $MinPP$ parses string in quadratic computational time.

Lemma 4.3. $MinPP$ parses a string in time proportional to the square of the length of the string.

Proof. Let N be the number of simple types in the processed string. $MinPP$ sees each type exactly once in forward processing. This includes either attempting reductions with the top of the stack or searching for a critical fan. In both cases these processes are obtained via functions with constant time d. Thus the forward processing happens overall in time dN. Then, for each critical fan, we read the string backward. This process is done in time dN^2 at most. Finally, when backward critical reductions are found, we correct the stack and set of reductions. The correction functions have constant time c, so all corrections happen in time cN at most. Summing these terms we obtain:

$$time = dN^2 + (d + c)N.$$

□

5 $LinPP$: Linear Pregroup Parsing algorithm

Certain words are typically assigned compound types by the dictionary, e.g. $T := n^r s n^l$ for *transitive verbs*. It might be the case that a compound type T_W of a word W, is not irreducible. Both $MinPP$ and the parsers mentioned in the Introduction will read types in T_W and reduce T_W to an irreducible form. However, the main purpose of grammatical pregroup types is to tell us how to connect different words. Reducing words internally defeats this purpose. We want to overcome this limitation and construct an algorithm that ignores intra-word reductions. Given a word W_1 let T_1 be its corresponding type (simple or compound). In $MinPP$ we defined stages S_n corresponding to each simple type t_n being read. Let us write Z_1 for the *super stage* corresponding to word W_1 being read. Z_1 contains one or more S_n corresponding to each simple type in T_1. We modify $MinPP$ as follows.

- At stage Z_1, we add $T_1 = t_1 ... t_j$ to the stack. We immediately jump to super stage Z_2 and stage S_{j+1}.
- When each new word W_m, with $m > 1$ and $T_m := t_{m_1} ... t_{m_k}$ is processed, We try to contract $t_i t_{m_1}$. While types contracts we keep reducing the types t_{m_j} with the top of the stack. We stop when either a pair $t_i t_{m_j}$ does not contract or when we reach the end of the word.
- If $t_i t_{m_j} \nrightarrow 1$ and t_{m_j} is linear, we add $t_{m_j} ... t_{m_k}$ to the stack and jump to stages Z_{m+1}, S_{m_k+1}. If t_{m_j} is critical, we handle it as in $MinPP$: if a backward reduction is found, the stack and reduction set are updated and we move to S_{m_j+1}; if the backward reduction is not found, we add $t_{m_j} ... t_{m_k}$ to the stack and move to the next word as above.

In other words, $LinPP$ follows the same computational steps of $MinPP$, while only checking reductions between types of separate words. By assuming dictionaries whose sentences do not involve intra-word reductions, the above proof of correctness can be adapted to hold for $LinPP$. Modifications are trivial. We previously highlighted the importance of a linear parser; up to this point $LinPP$ computes parsing in quadratic time. Below we impose some further restrictions on the input data, which enable linear computational time.

Definition 5.1. We say that a dictionary of complexity 2 is **critically bounded** if, given a constant $K \in \mathbb{N}$, for each critical type t_c in a string, exactly one of the following is true:

- t_c reduces when processing the substring $t_{c-K} ... t_c$ backwards;
- t_c does not not reduce in the string.

In other words, critical underlinks cannot exceed lenght K.

Lemma 5.2. Assume the restrictions of section 4.1, no-intra word reductions, and critically bounded dictionaries. Then $LinPP$ parses strings in linear computational time.

Proof. Assume a string of length N. $LinPP$ forward processing involves reading each type at most once. Thus it happens at most in time dN, with d as in section 4.1. Moreover, when a critical fan is read, the string is parsed backward, reading at most K types. This process takes dK time per critical fan. Thus it takes overall times dKN. Finally there is an extra linear term, cN, given by the time spent to correct the stack and reduction set. Summing up those terms, we obtain overall computational time CN, with $C = d(1 + K) + c$ being a constant specific to each bounded dictionary. □

6 Conclusion

In this paper we first defined a quadratic pregroup parser, $MinPP$, inspired by Preller's minimal parser. We proved its correctness with respect to reducing strings to irreducible forms, and in particular to parse sentences to the sentence type, in the class of pregroup grammar charactorised by complexity 2 and guarded critical types. Note that our definition of guards differs from the one given in (Preller, 2007a). We then modified $MinPP$ in order to remove intra-words links. We proved that the obtained algorithm, $LinPP$, is linear, given that the dictionaries are critically bounded. $LinPP$ was implemented in Python and it's soon to be integrated in the *DisCopy* package. The reader can find it at github:*oxford-quantum-group/discopy*. $LinPP$ is an important step towards the implementation of a supervised pregroup tagger, which will enable extensive testing of the DisCoCat model on task involving larger data-sets. Future theoretical work and implementations will involve researching a probabilistic pregroup parser based on $LinPP$. Future work might also involve investigation the connection between pregroup parsers and compositional dynamical networks.

Acknowledgments

The author thanks Giovanni Defelice and Alexis Toumi for the constructive discussions and feedback on the parser and its Python implementation. The author thanks their supervisors, Bob Coecke and Stefano Gogioso, for directions and feedback. Many Thanks to Antonin Delpeuch for insights on cubic pregroup parsers. Last but not least, the author thanks Anne Preller for the precious input in reformulating the definition of *guards* and for the insightful conversation on the topic of this paper.

References

W. Buszkowski. 2009. Lambek grammars based on pregroups. *Logical Aspects of Computa- tional Linguistics, LNAI 2099.*

B. Coecke. 2019. The mathematics of text structure. *arXiv: 1904.03478 [cs.CL].*

B. Coecke, M. Sadrzadeh, and S. Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *arXiv:1003.4394v1 [cs.CL]*, pages 1–34.

Bob Coecke, Giovanni de Felice, Konstantinos Meichanetzidis, and Alexis Toumi. 2020a. Foundations for Near-Term Quantum Natural Language Processing. *arXiv:2012.03755 [quant-ph].*

Bob Coecke, Giovanni de Felice, Konstantinos Meichanetzidis, Alexis Toumi, Stefano Gogioso, and Nicolo Chiappori. 2020b. Quantum natural language processing.

G. Defelice, A. Toumi, and B. Coecke. 2020. Discopy: Monoidal categories in python. *arXiv:2005.02975 [math.CT].*

S. Degeilh and A. Preller. 2005. Efficiency of pregroups and the french noun phrase. *Journal of Language, Logic and Information*, 4:423–444.

J. Earley. 1970. An efficient context-free parsing algorithm. *Communications of the AMC*, 13:94–102.

J. Lambek. 1997. Type grammars revisited. *LACL 1997*, pages 1–27.

Konstantinos Meichanetzidis, Stefano Gogioso, Giovanni De Felice, Nicolò Chiappori, Alexis Toumi, and Bob Coecke. 2020a. Quantum Natural Language Processing on Near-Term Quantum Computers. *arXiv:2005.04147 [quant-ph].*

Konstantinos Meichanetzidis, Alexis Toumi, Giovanni de Felice, and Bob Coecke. 2020b. Grammar-Aware Question-Answering on Quantum Computers. *arXiv:2012.03756 [quant-ph].*

K. Moroz. 2009a. Parsing pregroup grammars in polynomial time. *International Multiconference on Computer Science and Information Technology.*

K. Moroz. 2009b. A savateev-style parsing algorithm for pregroup grammars. *International Conference on Formal Grammar*, pages 133–149.

A. Preller. 2007a. Linear processing with pregoups. *Studia Logica.*

A. Preller. 2007b. Towards discourse prepresentation via pregroup grammars. *JoLLI*, 16:173–194.

D. Shiebler, A. Toumi, and M. Sadrzadeh. 2020. Incremental monoidal grammars. *arXiv:2001.02296v2 [cs.FL].*

William Zeng and Bob Coecke. 2016. Quantum Algorithms for Compositional Natural Language Processing. *Electronic Proceedings in Theoretical Computer Science*, 221:67–75.

A CCG-Based Version of the DisCoCat Framework

Richie Yeung and **Dimitri Kartsaklis**
Cambridge Quantum Computing Ltd.
17 Beaumont Street, Oxford, OX1 2NA, UK
{richie.yeung;dimitri.kartsaklis}@cambridgequantum.com

Abstract

While the DisCoCat model (Coecke et al., 2010) has been proved a valuable tool for studying compositional aspects of language at the level of semantics, its strong dependency on pregroup grammars poses important restrictions: first, it prevents large-scale experimentation due to the absence of a pregroup parser; and second, it limits the expressibility of the model to context-free grammars. In this paper we solve these problems by reformulating DisCoCat as a passage from Combinatory Categorial Grammar (CCG) to a category of semantics. We start by showing that standard categorial grammars can be expressed as a biclosed category, where all rules emerge as currying/uncurrying the identity; we then proceed to model permutation-inducing rules by exploiting the symmetry of the compact closed category encoding the word meaning. We provide a proof of concept for our method, converting "Alice in Wonderland" into DisCoCat form, a corpus that we make available to the community.

1 Introduction

The compositional model of meaning by Coecke, Sadrzadeh and Clark (Coecke et al., 2010) (from now on DisCoCat[1]) provides a conceptual way of modelling the interactions between the words in a sentence at the level of semantics. At the core of the model lies a passage from a grammatical derivation to a mathematical expression that computes a representation of the meaning of a sentence from the meanings of its words. In its most common form, this passage is expressed as a functor from a *pregroup grammar* (Lambek, 2008) to the category of finite-dimensional vector spaces and linear maps, **FdVect**, where the meanings of words live in the form of vectors and tensors (Kartsaklis et al., 2016). The job of the functor is to take a grammatical derivation and translate it into a linear-algebraic operation between tensors of various orders, while the composition function that returns the meaning of the sentence is tensor contraction.

The particular choice of using a pregroup grammar as the domain of this functor is based on the fact that a pregroup, just like the semantics category on the right-hand side, has a compact-closed structure, which simplifies the transition considerably. However, while this link between pregroup grammars and DisCoCat is well-motivated, it has also been proved stronger than desired, imposing some important restrictions on the framework. As a motivating example for this paper we mention the absence of any robust statistical pregroup parser (at the time of writing) that would provide the derivations for any large-scale DisCoCat experiment on sentences of arbitrary grammatical forms. As up to the time of writing (11 years after the publication of the paper that introduced DisCoCat), all experimental work related to the model is restricted to small datasets with sentences of simple fixed grammatical structures (e.g. subject-verb-object) that are provided to the system manually.

Furthermore, pregroup grammars have been proved to be weakly equivalent to context-free grammars (Buszkowski, 2001), a degree of expressiveness that it is known to be not adequate for natural language; for example Bresnan et al. (1982) and Shieber (1985) have shown that certain syntactical constructions in Dutch and Swiss-German give rise to *cross-serial dependencies* and are beyond context-freeness. While in practice these cases are quite limited, it would still be linguistically interesting to have a version of DisCoCat that is free of any restrictions with regard to its generative power.

In this paper we overcome the above problems by detailing a version of DisCoCat whose domain is *Combinatory Categorial Grammar* (CCG) (Steed-

[1] DIStributional COmpositional CATegorical.

June 16th, 2021. ©2021 Association for Computational Linguistics

man, 1987, 1996). We achieve this by encoding CCG as a biclosed category, where all standard order-preserving rules of the grammar find a natural translation into biclosed diagrams. CCG rules whose purpose is to relax word ordering and allow cross-serial dependencies are encoded as special morphisms. We then define a closed monoidal functor from the biclosed category freely generated over a set of atomic types, a set of words, and the set of arrows encoding the special rules of the grammar to a compact-closed category. We show that since the category of the semantics is symmetric, the special rules that allow word permutation can be encoded efficiently using the mechanism of "swapping the wires". As we will see in Section 3, while in the past there were other attempts to represent CCG in DisCoCat using similar methods (Grefenstette, 2013), this is the first time that a complete and theoretically sound treatment is provided and implemented in practice.

By presenting a version of DisCoCat which is no longer bound to pregroups, we achieve two important outcomes. First, since CCG is shown to be a *mildly context-sensitive* grammar (Vijay-Shanker and Weir, 1994), we increase the generative power of DisCoCat accordingly; and second, due to the availability of many robust CCG parsers that can be used for obtaining the derivations of sentences in large datasets – see, for example (Clark and Curran, 2007) – we make large-scale DisCoCat experiments on sentences of arbitrary grammatical structures possible for the first time.

In fact, we demonstrate the applicability of the proposed method by using a standard CCG parser (Yoshikawa et al., 2017) to obtain derivations for all sentences in the book "Alice's Adventures in Wonderland", which we then convert to DisCoCat diagrams based on the theory described in this paper. This resource – the first in its kind – is now available to the DisCoCat community for facilitating research and experiments. Furthermore, a web-based tool that allows the conversion of any sentence into a DisCoCat diagram is available at CQC's QNLP website.[2]

2 Introduction to DisCoCat

Based on the mathematical framework of compact-closed categories and inspired by the category-theoretic formulation of quantum mechanics (Abramsky and Coecke, 2004), the compositional distributional model of Coecke et al. (2010) computes semantic representations of sentences by composing the semantic representations of the individual words. This computation is guided by the grammatical structure of the sentence, therefore at a higher level the model can be summarised as the following transition:

$$\textbf{Grammar} \Rightarrow \textbf{Meaning}$$

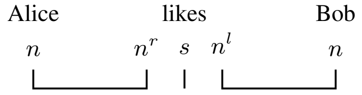
Up until now, at the left-hand side of this mapping lies a *pregroup grammar* (Lambek, 2008), that is, a partially-ordered monoid whose each element p has a left (p^l) and a right (p^r) adjoint such that:

$$p \cdot p^r \leq 1 \leq p^r \cdot p \qquad p^l \cdot p \leq 1 \leq p \cdot p^l \tag{1}$$

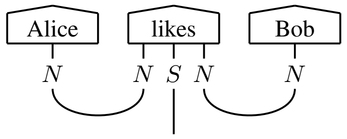
The inequalities above form the production rules of the grammar. As an example, assume a set of atomic types $\{n, s\}$ where n is a noun or a noun phrase and s a well-formed sentence, and type-assignments (Alice, n), (Bob, n), and (likes, $n^r \cdot s \cdot n^l$); based on Eq. 1, the pregroup derivation for the sentence "Alice likes Bob" becomes:

$$n \cdot n^r \cdot s \cdot n^l \cdot n \leq 1 \cdot s \cdot 1 \leq s \tag{2}$$

showing that the sentence is grammatical. Note that the transitive verb "likes" gets the compound type $n^r \cdot s \cdot n^l$, indicating that such a verb is something that expects an n on the left and another one on the right in order to return an s. In diagrammatic form, the derivation is shown as below:



where the "brackets" (⊔) correspond to the grammatical reductions. Kartsaklis et al. (2016) showed how a structure-preserving passage can be defined between a pregroup grammar and the category of finite-dimensional vector spaces and linear maps (**FdVect**), by sending each atomic type to a vector space, composite types to tensor products of spaces and cups to inner products. The DisCoCat diagram (also referred to as a *string diagram*) for the above derivation in **FdVect** becomes:



where N, S are vector spaces, "Alice" and "Bob" are represented by vectors in N, while "likes" is a tensor of order 3 in $N \otimes S \otimes N$. Here the "cups" (∪)

[2] Links for the the corpus and the web demo are provided in Section 10.

correspond to *tensor contractions*, so that the vector for the whole sentence lives in S. The preference for using a pregroup grammar in the DisCoCat model becomes clear when we notice the structural similarity between the two diagrams above, and how closely the pregroup derivation dictates the shapes of the tensors and the contractions.

3 Related work

Implementations of the DisCoCat model have been provided by Grefenstette and Sadrzadeh (2011) and Kartsaklis et al. (2012), while Piedeleu et al. (2015) detail a version of the model based on density matrices for handling lexical ambiguity. DisCoCat has been used extensively in conceptual tasks such as textual entailment at the level of sentences, see for example (Bankova et al., 2019; Lewis, 2019). Further, exploiting the formal similarity of the model with quantum mechanics, Meichanetzidis et al. (2020) and Lorenz et al. (2021) have used it recently with success for the first implementations of NLP models on NISQ computers.

The connection between categorial grammars and biclosed categories is long well-known (Lambek, 1988), and discussed by Dougherty (1993). More related to DisCoCat, and in an attempt to detach the model from pregroups, Coecke et al. (2013) detail a passage from the original Lambek calculus, formed as a biclosed category, to vector spaces. In (Grefenstette, 2013) can be found a first attempt to explicitly provide categorical semantics for CCG, in the context of a functor from a closed category augmented with swaps to **FdVect**. In that work, though, the addition of swaps introduces an infinite family of morphisms that collapse the category and lead to an overgenerating grammar. Further, the actual mapping of crossed composition rules to the monoidal diagrams has flaws, as given in diagrammatic and symbolic forms – see footnote 5. We close this section by mentioning the work by Maillard et al. (2014), which describes how CCG derivations can be expressed directly as tensor operations in the context of DisCoCat, building on (Grefenstette, 2013).

4 Categorial grammars

We start our exposition by providing a short introduction to categorial grammars. A *categorial grammar* (Ajdukiewicz, 1935) is a grammatical formalism based on the assumption that certain syntactic constituents are functions applied on lower-order arguments. For example, an intransitive verb gets the type *S**NP*, denoting that this kind of verb is a function that expects a noun phrase on the left in order to return a sentence, while a determiner has type *NP*/*N* – a function that expects a noun on the right to return a noun phrase. The direction of the slash determines the exact position of the argument with regard to the word that represents the function. In the following derivation for the sentence "Alice likes Bob", the noun phrases and the transitive verb are assigned types *NP* and *(S**NP)*/*NP* respectively.

$$\dfrac{\dfrac{\text{Alice}}{NP} \qquad \dfrac{\dfrac{\text{likes}}{(S\backslash NP)/NP} \quad \dfrac{\text{Bob}}{NP}}{S\backslash NP}{>}}{S}{<}$$

As the diagram shows, a term with type X/Y takes a term of type Y on the right in order to return a term of type X. Similarly, a term with type $X\backslash Y$ takes a term of type Y on the left, in order to return a term of type X. In this paper we adopt a slightly different and hopefully more intuitive notation for categorial types: X/Y becomes $X \leftarrowtail Y$ while for $X\backslash Y$ we will use $Y \rightarrowtail X$. Using the new notation, the above diagram takes the form shown in Figure 1.

$$\dfrac{\dfrac{\text{Alice}}{NP} \qquad \dfrac{\dfrac{\text{likes}}{(NP \rightarrowtail S) \leftarrowtail NP} \quad \dfrac{\text{Bob}}{NP}}{NP \rightarrowtail S}{>}}{S}{<}$$

Figure 1

The two rules described above are called *forward* and *backward application*, respectively, and formally can be defined as below:

$$\text{FA } (>)\ \frac{\alpha : X \leftarrowtail Y \qquad \beta : Y}{\alpha\beta : X}$$

$$\text{BA } (<)\ \frac{\alpha : Y \qquad \beta : Y \rightarrowtail X}{\alpha\beta : X}$$

Categorial grammars restricted to application rules are known as *basic categorial grammars* (BCG) (Bar-Hillel, 1953), and have been proved to be equivalent to context-free grammars (Bar-Hillel et al., 1960) and pregroup grammars (Buszkowski, 2001). Interestingly, although all grammars mentioned above are equivalent in terms of theoretical expressiveness, BCGs are restrictive on the order of the reductions in a sentence. In the derivation of Figure 1, we see for example that a transitive verb must always first compose with its object, and then with the subject.

To address this problem, some categorial grammars (including CCG) contain *type-raising* and *composition* rules which, although they do not affect grammar's theoretical power, allow some additional flexibility in the order of composition. These rules can be seen of as a form of *currying*, and are discussed in more depth in Section 6.

$$\text{FC } (B_>)\ \frac{\alpha : X \leftarrowtail Y \qquad \beta : Y \leftarrowtail Z}{\alpha\beta : X \leftarrowtail Z}$$
$$\text{BC } (B_<)\ \frac{\alpha : Z \rightarrowtail Y \qquad \beta : Y \rightarrowtail X}{\alpha\beta : Z \rightarrowtail X}$$

$$\text{FTR } (T_>)\ \frac{\alpha : X}{\alpha : T \leftarrowtail (X \rightarrowtail T)}$$
$$\text{BTR } (T_<)\ \frac{\alpha : X}{\alpha : (T \leftarrowtail X) \rightarrowtail T}$$

In Figure 2 we see how type-raising (T) and composition (B) can be used to change the order of reductions in our example sentence, in a version that the verb is first composed with the subject and then with the object.

$$\frac{\dfrac{\dfrac{\dfrac{\text{Alice}}{NP}}{S \leftarrowtail (NP \rightarrowtail S)}{>}\mathrm{T} \quad \dfrac{\text{likes}}{(NP \rightarrowtail S) \leftarrowtail NP}}{S \leftarrowtail NP}{>}\mathrm{B} \quad \dfrac{\text{Bob}}{NP}}{S}{>}$$

Figure 2

Finally, in CCG composition has also a generalized version, where additional arguments (denoted below as $\$_1$) are allowed to the right of the Z category.

$$\text{GFC } (B^n_>)\ \frac{\alpha : X \leftarrowtail Y \qquad \beta : (Y \leftarrowtail Z) \leftarrowtail \$_1}{\alpha\beta : (X \leftarrowtail Z) \leftarrowtail \$_1}$$
$$\text{GBC } (B^n_<)\ \frac{\alpha : X \rightarrowtail Y \qquad \beta : (Y \rightarrowtail Z) \rightarrowtail \$_1}{\alpha\beta : (X \rightarrowtail Z) \rightarrowtail \$_1}$$

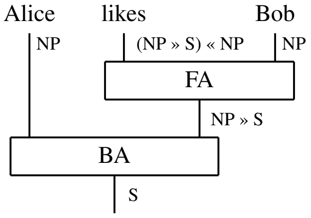
The rule can be seen as "ignoring the brackets" in the right-hand type:

$$\frac{\dfrac{\text{might}}{(NP \rightarrowtail S) \leftarrowtail VP} \quad \dfrac{\text{give}}{(VP \leftarrowtail NP) \leftarrowtail NP}}{((NP \rightarrowtail S) \leftarrowtail NP) \leftarrowtail NP}{>}\mathrm{B}^2$$

The generalized composition rules have special significance, since it is argued to be the reason for the beyond context-free generative capacity of CCG – see for example (Kuhlmann et al., 2015).

5 Categorial grammars as biclosed categories

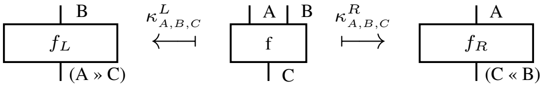
Categorial grammars can be seen as a proof system, and form a *biclosed category* $\mathcal{B}$ whose objects are the categorial types while the arrows $X \to Y$ correspond to proofs with assumption X and conclusion Y. A word with categorial type X lives in this category as an axiom, that is, as an arrow of type $I \to X$ where the monoidal unit I is the empty assumption. Below we show the biclosed diagram for the CCG derivation of the sentence "Alice likes Bob":



We remind the reader that a biclosed category is both left-closed and right-closed, meaning that it is equipped with the following two isomorphisms:

$$\kappa^L_{A,B,C} : \mathcal{B}(A \otimes B, C) \cong \mathcal{B}(B, A \rightarrowtail C)$$
$$\kappa^R_{A,B,C} : \mathcal{B}(A \otimes B, C) \cong \mathcal{B}(A, C \leftarrowtail B)$$

where κ^L corresponds to left-currying and κ^R to right-currying. Diagrammatically:



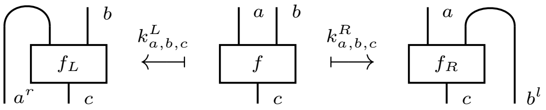
A key observation for the work in this paper is that all basic categorial rules exist naturally in any biclosed category and can emerge solely by currying and uncurrying identity morphisms; this is shown in Figure 3. Hence any CCG derivation using the rules we have met so far[3] exists in a biclosed category freely generated over atomic types and word arrows.

6 From biclosed to compact-closed

We will now define a monoidal functor from a grammar expressed as a biclosed category to DisCoCat diagrams. DisCoCat diagrams exist in a compact-closed category $\mathcal{C}$, where every object is left- and right-dualisable and the left and right internal hom-objects between objects X and Y are isomorphic to $X^r \otimes Y$ and $Y \otimes X^l$ respectively. Thus we can directly define the left and right currying isomorphisms using the dual objects:

$$k^L_{a,b,c} : \mathcal{C}(a \otimes b, c) \cong \mathcal{C}(b, a^r \otimes c)$$
$$k^R_{a,b,c} : \mathcal{C}(a \otimes b, c) \cong \mathcal{C}(a, c \otimes b^l)$$

Left and right currying in compact-closed categories get intuitive diagrammatic representations:



[3] CCG also uses a *crossed* version of composition, which is a special case and discussed in more detail in Section 7.

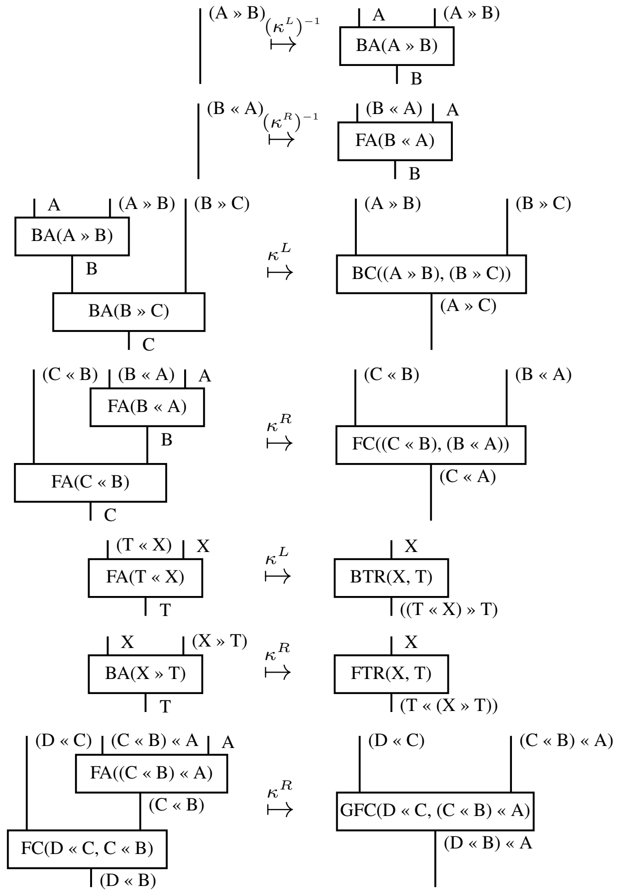


Figure 3: Categorial rules as currying/uncurrying in a biclosed category.

which allows us to functorially convert all categorial grammar rules into string diagrams, as in Figure 4.

Definition 6.1. *F is a closed monoidal functor from the biclosed category $\mathcal{B}$ of CCG derivations to the compact-closed category $\mathcal{C}$ of DisCoCat diagrams.*

Let $\{\mathrm{NP}, \mathrm{S}, \mathrm{PP}\}$ be a set of atomic CCG types, indicating a noun phrase, a sentence, and a prepositional phrase, respectively, and T a lexical type. We define the following mapping:

$$F(\mathrm{NP}) = n \quad F(\mathrm{S}) = s \quad F(\mathrm{PP}) = p$$
$$F(\mathrm{word}_{\mathcal{B}} : I_{\mathcal{B}} \to T) = \mathrm{word}_{\mathcal{C}} : I_{\mathcal{C}} \to F(T)$$

As a closed monoidal functor, $F : \mathcal{B} \to \mathcal{C}$ satisfies the following equations:

$$F(X \circ Y) = F(X) \circ F(Y) \quad F(\mathit{Id}_X) = \mathit{Id}_{F(X)}$$
$$F(X \otimes Y) = F(X) \otimes F(Y) \quad F(I_{\mathcal{B}}) = I_{\mathcal{C}}$$

$$F(X \rightarrowtail Y) = F(X)^r \otimes F(Y)$$
$$F(X \leftarrowtail Y) = F(X) \otimes F(Y)^l$$

Furthermore, for any diagram $d : A \otimes B \to C$,

$$F(\kappa^L_{A,B,C}(d)) = k^L_{a,b,c}(F(d))$$
$$F(\kappa^R_{A,B,C}(d)) = k^R_{a,b,c}(F(d))$$

where $F(A) = a, F(B) = b, F(C) = c$.

Alternatively we can say that the following diagram commutes:

$$\begin{tikzcd}
\mathcal{B}(B, A \rightarrowtail C) \arrow[r, "F"] & \mathcal{C}(b, a^r \otimes c) \\
\mathcal{B}(A \otimes B, C) \arrow[r, "F"'] \arrow[u, "\kappa^L_{A,B,C}"'] \arrow[d, "\kappa^R_{A,B,C}"] & \mathcal{C}(a \otimes b, c) \arrow[u, "k^L_{a,b,c}"'] \arrow[d, "k^R_{a,b,c}"] \\
\mathcal{B}(A, C \leftarrowtail B) \arrow[r, "F"'] & \mathcal{C}(a, c \otimes b^l)
\end{tikzcd}$$

As an example, below you can see how the backward application rule, derived by uncurrying an identity morphism, is converted into a string diagram in $\mathcal{C}$.

$$\begin{aligned}
F(BA(A \rightarrowtail B)) &= F((\kappa^L_{A,A\rightarrowtail B,B})^{-1}(\mathrm{Id}_{A\rightarrowtail B})) \\
&= (k^L_{a,a^r\otimes b,b})^{-1}(F(\mathrm{Id}_{A\rightarrowtail B})) \\
&= (k^L_{a,a^r\otimes b,b})^{-1}(\mathrm{Id}_{F(A\rightarrowtail B)}) \\
&= (k^L_{a,a^r\otimes b,b})^{-1}(\mathrm{Id}_{a^r\otimes b}) \\
&= (k^L_{a,a^r\otimes b,b})^{-1}(\mathrm{Id}_{a^r} \otimes \mathrm{Id}_b)
\end{aligned}$$

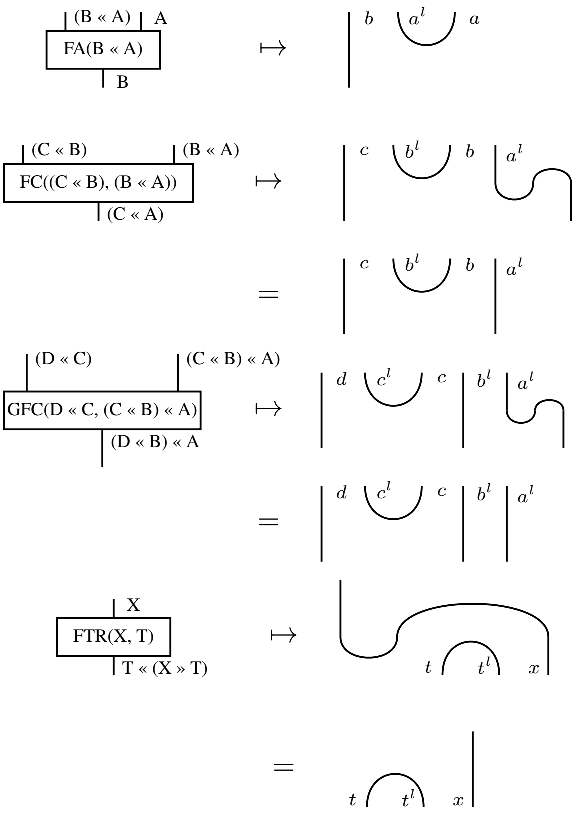


Figure 4: Functorial conversion of "forward" categorial grammar rules in biclosed form into string diagrams.

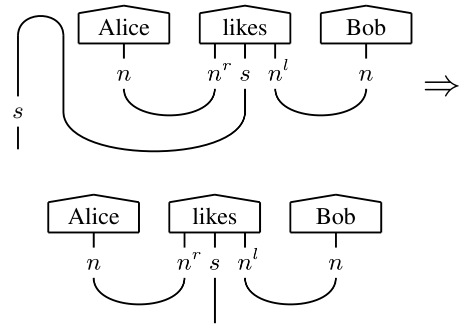


Figure 5: Rewriting of string diagrams. The starting diagram corresponds to the derivation of Figure 2, which uses type-raising. By re-arranging the sentence wire we get the non-type-raised version of Figure 1.

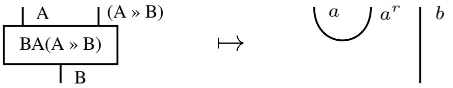
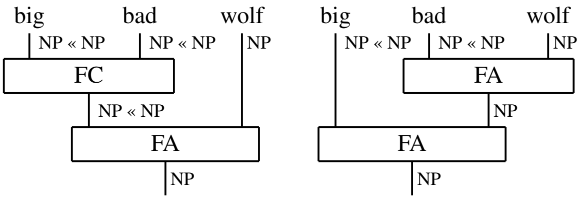
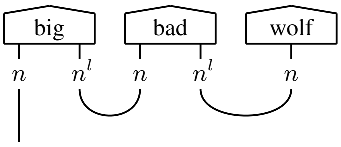


Figure 4 provides the translation of all forward rules into DisCoCat diagrams. The conversion for the backward rules can be obtained by reflecting the diagrams horizontally and replacing the left/right adjoints with right/left adjoints.

One advantage of representing parse trees using compact-closed categories over biclosed categories and categorial grammars is that the rewriting rules of string diagrams enable us to show more clearly the equivalence between two parse trees. Take for example the phrase "big bad wolf", which in biclosed form has two different derivations:



However, when these derivations are sent to a compact-closed category, they become equivalent to the following diagram which is agnostic with regard to composition order:



Another example of this is in the use of the type-raising rule in CCG, which is analogous to expansion in pregroups, and in DisCoCat can be represented using a "cap" (∩). Therefore, the derivations in Figures 1 and 2, when expressed as DisCoCat diagrams, are equal up to planar isotopy (Figure 5).

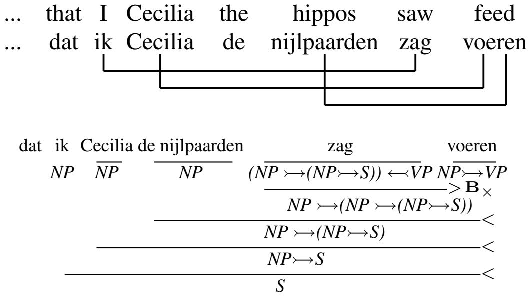

... that I Cecilia the hippos saw feed
... dat ik Cecilia de nijlpaarden zag voeren
dat ik Cecilia de nijlpaarden zag voeren
NP NP NP
(NP ↣(NP↣S)) ↢VP
NP↣VP
> B×
NP ↣(NP ↣(NP↣S))
<
NP ↣(NP↣S)
<
NP↣S
<
S

Figure 6: Cross-serial dependencies in Dutch for the phrase "...that I saw Cecilia feeding the hippos" (Steedman, 2000).

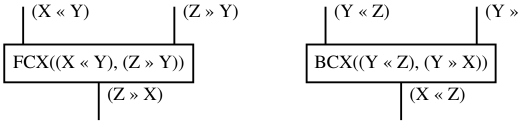
7 Crossed composition

All rules we have met so far are order-preserving, in the sense that they expect all words or textual constituents in a sentence to be found at their canonical positions. This is not always the case though, since language can be also used quite informally. To handle those cases without introducing additional types per word, CCG is equipped with the rule of *crossed composition* (Steedman, 2000), the definition of which is the following:

$$\text{FCX } (BX_{>}) \; \frac{\alpha : X \leftarrowtail Y \qquad \beta : Z \rightarrowtail Y}{\alpha\beta : Z \rightarrowtail X}$$

$$\text{BCX } (BX_{<}) \; \frac{\alpha : Y \leftarrowtail Z \qquad \beta : Y \rightarrowtail X}{\alpha\beta : X \leftarrowtail Z}$$

In biclosed form, the crossed composition rules are expressed as below:



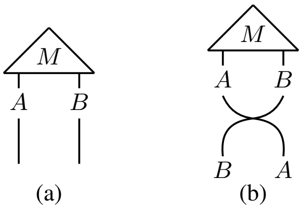
Crossed composition comes also in a generalized form as the standard (or *harmonic*) composition, and allows treatment of *cross-serial dependencies*, similar to those met in Dutch and Swiss-German (Figure 6). In English the rule is used in a restricted form[4], mainly to allow a certain degree of word associativity and permutativity when this is required.

For example, such a case is heavy *NP-shift*, where the adverb comes between the verb and its direct object (Baldridge, 2002). Consider the sentence "John passed successfully his exam", the CCG derivation of which is shown below:

[4] Steedman (2000) disallows the use of the forward version in English, while the backward version is permitted only when $Y = NP \rightarrowtail S$.

$$\cfrac{\cfrac{\text{John}}{NP} \quad \cfrac{\cfrac{\cfrac{\text{passed}}{(NP \rightarrowtail S) \leftarrowtail NP} \quad \cfrac{\text{successfully}}{(NP \rightarrowtail S) \rightarrowtail (NP \rightarrowtail S)}}{(NP \rightarrowtail S) \leftarrowtail NP}{<_{\mathbf{B}\times}} \quad \cfrac{\text{his exam}}{NP}}{NP \rightarrowtail S}{>}}{S}{<}$$

Note that the rule introduces a crossing between the involved types, which is not representable in pregroups. However, we remind the reader that the compact closed category where the DisCoCat diagrams live is a *symmetric* monoidal category, which means that for any two objects A and B it holds that $A \otimes B \cong B \otimes A$. In diagrammatic form this corresponds to a *swap* of the wires, as below:



In the case of **FdVect**, the state above would correspond to a matrix $M \in A \otimes B$ (a), while its swap (b) is nothing more than the transposition of that matrix, M^{T}.

Thus, by exploiting the symmetry of the semantics category, the DisCoCat diagrams for the two crossed composition rules take the form shown in Figure 7.[5]

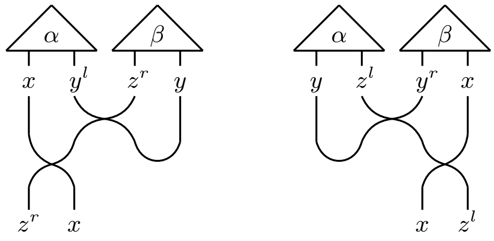
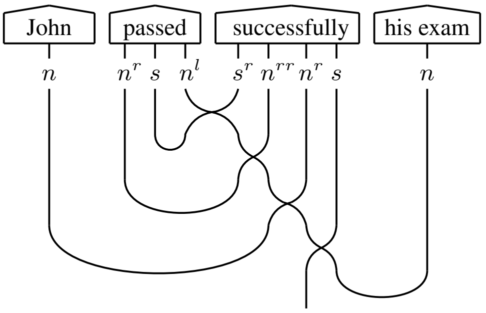


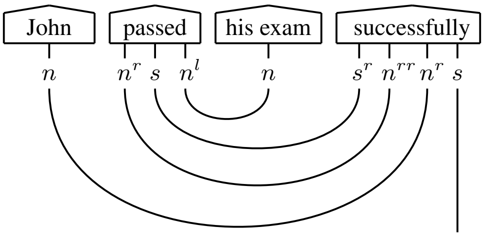
Figure 7: Crossed composition in DisCoCat (forward version on the left, backward on the right).

We are now in position to revisit the functorial passage described in Section 6 in order to include crossed composition. In contrast to other categorial rules, crossed composition does not occur naturally in a biclosed setting, so we have to explicitly add the corresponding boxes in the generating set of category $\mathcal{B}$, which is the domain of our functor. The mapping of these special boxes to compact-closed diagrams is defined in Figure 7. Deriving the generalized versions of the rules in biclosed and compact-closed form similarly to the harmonic cases is a straightforward exercise.

Based on the above, our NP-shift case gets the following diagrammatic representation:



Interestingly, this diagram can be made planar by relocating the state of the object in its canonical (from a grammar perspective) position:



which demonstrates very clearly that, in a proper use of English, permutation-inducing rules become redundant.

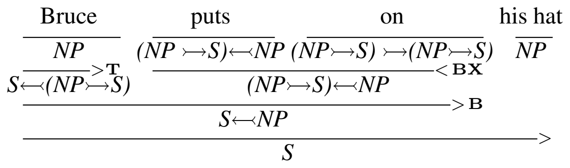
We would like to close this section with a comment on the presence of swaps in the DisCoCat category, and what exactly the implications of this are. Obviously, an unrestricted use of swaps in the semantics category would allow every possible arbitrary permutation of the words, resulting in an overgenerating model that is useless for any practical application. However, as explained in Section 2, DisCoCat is not a grammar, but a *mapping* from a grammar to a semantics. Hence it is always responsibility of the grammar to pose certain restrictions in how the semantic form is generated. In the formulation we detailed in Sections 6 and 7, we have carefully defined a biclosed category as to not introduce extra morphisms to CCG, and a functor that maps to a subcategory of a compact-closed category such that the rigid structure of traditional DisCoCat is preserved.

8 Putting everything together

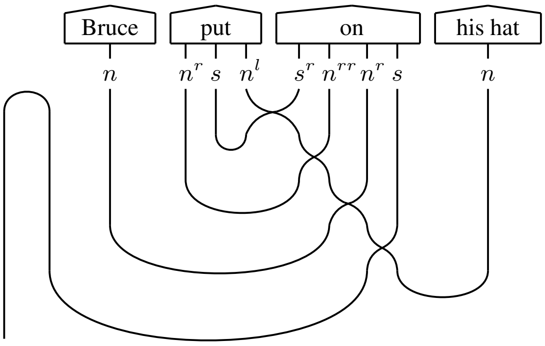
At this point we have the means to represent as a DisCoCat diagram every sentence in English language. In the following example, we consider

[5] The idea of representing crossing rules using swaps also appears in (Grefenstette, 2013); however the mapping provided there is incorrect, since there is a swap clearly missing before the last evaluation in the monoidal diagrams (p. 142, Fig. 7.7) as well as from the symbolic representations of the morphisms (p. 145).

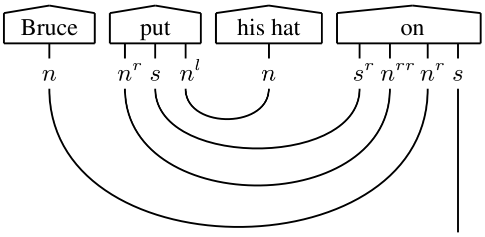
a derivation that includes type-raising, harmonic composition, and crossed composition:



The corresponding DisCoCat diagram is given below:



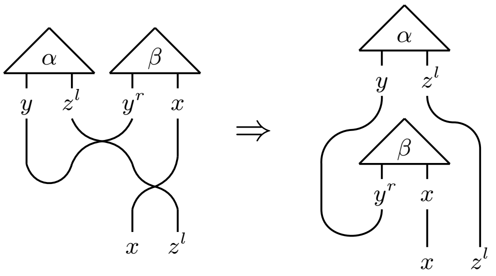
As before, relocating the object and yanking the wires produces a planar version of the diagram:



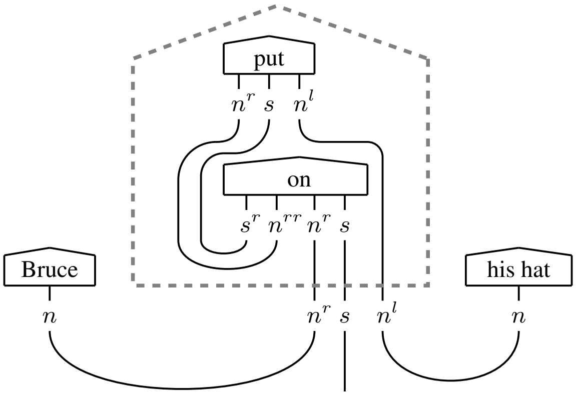
reflecting how the sentence would look if one used the separable[6] version of the phrasal verb.

9 Adhering to planarity

We have seen in Sections 7 and 8 how diagrams for sentences that feature crossed composition can be rearranged to equivalent diagrams that show a planar derivation. It is in fact *always* possible to rearrange the diagram of a derivation containing crossed composition into a planar diagram, since every instance of crossed composition between two subtrees α and β is subject to the following transformation:



By performing this rearrangement recursively on the subtrees, we obtain a planar monoidal diagram for the whole derivation. For example, a sentence containing a phrasal verb gets the following diagram:



Note how the two constituents of the phrasal verb are grouped together in a single state with type $n^r \cdot s \cdot n^l$, forming a proper transitive verb, and how the diagram is planar by construction without the need of any rearrangement.

Being able to express the diagrams without swaps is not only linguistically interesting, but also computationally advantageous. As mentioned before, on classical hardware swaps correspond to transpositions of usually large tensors; on quantum hardware, since a decomposition of a swap gate contains entangling gates, by reducing the number of swaps in a diagram we reduce the currently expensive entangling gates (such as CNOTs) required to synthesise the diagram.

10 A DisCoCat version of "Alice in Wonderland"

We demonstrate the theory of this paper by converting Lewis Carroll's "Alice in Wonderland"[7] in DisCoCat form. Our experiment is based on the following steps:

[6] A phrasal verb is *separable* when its object can be positioned between the verb and the particle.

[7] We used the freely available version of Project Gutenberg (https://www.gutenberg.org).

1. We use DepCCG parser[8] (Yoshikawa et al., 2017) to obtain CCG derivations for all sentences in the book.

2. The CCG derivation for a sentence is converted into biclosed form, as described in Section 5.

3. Finally, the functorial mapping from biclosed to string diagrams is applied, as detailed in Sections 6 and 7.

The DepCCG parser failed to parse 18 of the 3059 total sentences in the book, resulting in a set of 3041 valid CCG derivations, all of which were successfully converted into DisCoCat diagrams based on the methodology of this paper. The new corpus is now publicly available to further facilitate research in DisCoCat[9], and is provided in three formats: biclosed, monoidal, and DisCoCat, while PDF versions of the diagrams are also available. For the representation of the diagrams we used DisCoPy[10] (de Felice et al., 2020), a Python library for working with monoidal categories. Further, a Web tool that allows the conversion of any sentence to DisCoCat diagram providing various configuration and output options, including LaTeX code for rendering the diagram in a LaTeX document, is available at CQC's website[11]. In the Appendix we show the first few paragraphs of the book in DisCoCat form by using this option.

11 Some practical considerations

For the sake of a self-contained manuscript, in this section we discuss a few important technicalities related to CCG parsers that cannot be covered by the theory. The most important is the concept of *unary rules*, where a type is changed in an ad-hoc way at some point of the derivation in order to make an outcome possible. In the following CCG diagram, we see unary rules (U) changing $NP \rightarrowtail S$ to $NP \rightarrowtail NP$ and N to NP at a later point of the derivation.

$$\begin{array}{cccc}
\text{not} & \text{much} & \text{to} & \text{say}\\
\overline{N \leftarrowtail N} & \overline{N} & \overline{(NP \rightarrowtail S) \leftarrowtail (NP \rightarrowtail S)} & \overline{NP \rightarrowtail S}\\
\cline{1-2}\cline{3-4}
\multicolumn{2}{c}{N \quad >} & \multicolumn{2}{c}{NP \rightarrowtail S \quad >}\\
\cline{1-2}\cline{3-4}
\multicolumn{2}{c}{NP \quad <\mathbf{U}} & \multicolumn{2}{c}{NP \rightarrowtail NP \quad <\mathbf{U}}\\
\hline
\multicolumn{4}{c}{NP \quad <}
\end{array}$$

We address this problem by employing an indexing system that links the categorial types with their corresponding arguments in a way that is always possible to traverse the tree backwards and make appropriate replacements when a unary rule is met. For the above example, we get:

$$\begin{array}{cccc}
\text{not} & \text{much} & \text{to} & \text{say}\\
\overline{N_1 \leftarrowtail N_2} & \overline{N_2} & \overline{(NP \rightarrowtail S)_1 \leftarrowtail (NP \rightarrowtail S)_2} & \overline{(NP \rightarrowtail S)_2}\\
\cline{1-2}\cline{3-4}
\multicolumn{2}{c}{N_1 \quad >} & \multicolumn{2}{c}{(NP \rightarrowtail S)_1 \quad >}\\
\cline{1-2}\cline{3-4}
\multicolumn{2}{c}{NP \quad <\mathbf{U}} & \multicolumn{2}{c}{NP \rightarrowtail NP \quad <\mathbf{U}}\\
\hline
\multicolumn{4}{c}{NP \quad <}
\end{array}$$

Applying the unary rules is now distilled into replacing all instances of N_1 with NP and $(NP \rightarrowtail S)_1$ with $NP \rightarrowtail NP$ in the already processed part of the tree, which leads to the following free of unary rules final diagram:

$$\begin{array}{cccc}
\text{not} & \text{much} & \text{to} & \text{say}\\
\overline{NP \leftarrowtail N} & \overline{N} & \overline{(NP \rightarrowtail NP) \leftarrowtail (NP \rightarrowtail S)} & \overline{NP \rightarrowtail S}\\
\cline{1-2}\cline{3-4}
\multicolumn{2}{c}{NP \quad >} & \multicolumn{2}{c}{NP \rightarrowtail NP \quad >}\\
\hline
\multicolumn{4}{c}{NP \quad <}
\end{array}$$

Finally, we discuss conjunctions, which in CCG parsers take the special type *conj*. We essentially treat these cases as unary rules, constructing the destination type by the types of the two conjuncts:

$$\begin{array}{ccc}
\text{apples} & \text{and} & \text{oranges}\\
\overline{NP} & \overline{\text{conj}} & \overline{NP}\\
\cline{2-2}
 & (NP \rightarrowtail NP) \leftarrowtail NP \quad <\mathbf{U} & \\
\cline{2-3}
 & \multicolumn{2}{c}{NP \rightarrowtail NP \quad >}\\
\hline
\multicolumn{3}{c}{NP \quad <}
\end{array}$$

12 Future work and conclusion

In this paper we showed how CCG derivations can be expressed in DisCoCat, paving the way for large-scale applications of the model. In fact, presenting a large-scale experiment based on DisCoCat is a natural next step and one of our goals for the near future. Creating more DisCoCat-related resources, similar to the corpus introduced in this paper, is an important direction with obvious benefits to the community.

Acknowledgments

We would like to thank the anonymous reviewers for their useful comments. We are grateful to Steve Clark for his comments on CCG and the useful discussions on the generative power of the formalism. The paper has also greatly benefited from discussions with Alexis Toumi, Vincent Wang, Ian Fan, Harny Wang, Giovanni de Felice, Will Simmons, Konstantinos Meichanetzidis and Bob Coecke, who all have our sincere thanks.

[8] https://github.com/masashi-y/depccg

[9] https://qnlp.cambridgequantum.com/downloads.html.

[10] https://github.com/oxford-quantum-group/discopy

[11] https://qnlp.cambridgequantum.com/generate.html

References

S. Abramsky and B. Coecke. 2004. A Categorical Semantics of Quantum Protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science*, pages 415–425. IEEE Computer Science Press. arXiv:quant-ph/0402130.

Kazimierz Ajdukiewicz. 1935. Die syntaktische konnexitat. *Studia philosophica*, pages 1–27.

Jason Baldridge. 2002. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh, School of Informatics.

Dea Bankova, Bob Coecke, Martha Lewis, and Dan Marsden. 2019. Graded Entailment for Compositional Distributional Semantics. *Journal of Language Modelling*, 6(2):225–260.

Yehoshua Bar-Hillel. 1953. A quasi-arithmetical notation for syntactic description. *Language*, 29(1):47–58.

Yehoshua Bar-Hillel, Gaifman (C.), Eli Shamir, and C Caifman. 1960. *On categorial and phrase-structure grammars*. Weizmann Science Press.

Joan Bresnan, Ronald M Kaplan, Stanley Peters, and Annie Zaenen. 1982. Cross-serial Dependencies in Dutch. In *The formal complexity of natural language*, pages 286–319. Springer.

Wojciech Buszkowski. 2001. Lambek grammars based on pregroups. In *International Conference on Logical Aspects of Computational Linguistics*, pages 95–109. Springer.

Stephen Clark and James R Curran. 2007. Wide-coverage efficient statistical parsing with CCG and log-linear models. *Computational Linguistics*, 33(4):493–552.

Bob Coecke, Edward Grefenstette, and Mehrnoosh Sadrzadeh. 2013. Lambek vs. Lambek: Functorial Vector Space Semantics and String Diagrams for Lambek Calculus. *Annals of Pure and Applied Logic*, 164(11):1079–1100. Special issue on Seventh Workshop on Games for Logic and Programming Languages (GaLoP VII).

Bob Coecke, Mehrnoosh Sadrzadeh, and Steve Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. In J. van Benthem, M. Moortgat, and W. Buszkowski, editors, *A Festschrift for Jim Lambek*, volume 36 of *Linguistic Analysis*, pages 345–384.

Daniel J. Dougherty. 1993. Closed Categories and Categorial Grammar. *Notre Dame journal of formal logic*, 34(1):36–49.

Giovanni de Felice, Alexis Toumi, and Bob Coecke. 2020. DisCoPy: Monoidal Categories in Python. In *Proceedings of the 3rd Annual International Applied Category Theory Conference*. EPTCS.

Edward Grefenstette. 2013. *Category-theoretic Quantitative Compositional Distributional Models of Natural Language Semantics*. Ph.D. thesis, University of Oxford, Department of Computer Science.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011. Experimental Support for a Categorical Compositional Distributional Model of Meaning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404. Association for Computational Linguistics.

Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Stephen Pulman. 2012. A Unified Sentence Space for Categorical Distributional-Compositional Semantics: Theory and Experiments. In *COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Posters, 8-15 December 2012, Mumbai, India*, pages 549–558.

Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, Stephen Pulman, and Bob Coecke. 2016. *Reasoning about meaning in natural language with compact closed categories and Frobenius algebras*, Lecture Notes in Logic, page 199–222. Cambridge University Press.

Marco Kuhlmann, Alexander Koller, and Giorgio Satta. 2015. Lexicalization and Generative Power in CCG. *Computational Linguistics*, 41(2):187–219.

J. Lambek. 2008. *From Word to Sentence*. Polimetrica, Milan.

Joachim Lambek. 1988. Categorial and Categorical Grammars. In *Categorial grammars and natural language structures*, pages 297–317. Springer.

Martha Lewis. 2019. Modelling Hyponymy for DisCoCat. In *Proceedings of the Applied Category Theory Conference*, Oxford, UK.

Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis, and Bob Coecke. 2021. QNLP in Practice: Running Compositional Models of Meaning on a Quantum Computer. *arXiv preprint arXiv:2102.12846*.

Jean Maillard, Stephen Clark, and Edward Grefenstette. 2014. A Type-Driven Tensor-Based Semantics for CCG. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pages 46–54, Gothenburg, Sweden. Association for Computational Linguistics.

Konstantinos Meichanetzidis, Alexis Toumi, Giovanni de Felice, and Bob Coecke. 2020. Grammar-Aware Question-Answering on Quantum Computers.

Robin Piedeleu, Dimitri Kartsaklis, Bob Coecke, and Mehrnoosh Sadrzadeh. 2015. Open System Categorical Quantum Semantics in Natural Language Processing. In *Proceedings of the 6th Conference on Algebra and Coalgebra in Computer Science*, Nijmegen, Netherlands.

Stuart M Shieber. 1985. Evidence Against the Context-Freeness of Natural Language. In *Philosophy, Language, and Artificial Intelligence*, pages 79–89. Springer.

Mark Steedman. 1987. Combinatory grammars and parasitic gaps. *Natural Language & Linguistic Theory*, 5(3):403–439.

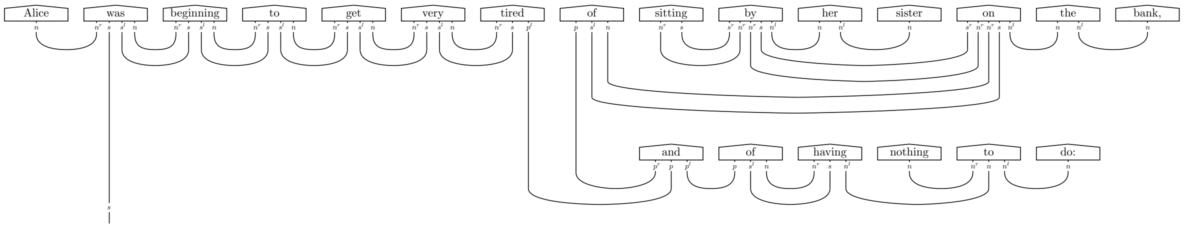
Mark Steedman. 1996. A very short introduction to ccg. *Unpublished paper. http://www. coqsci. ed. ac. uk/steedman/paper. html*.

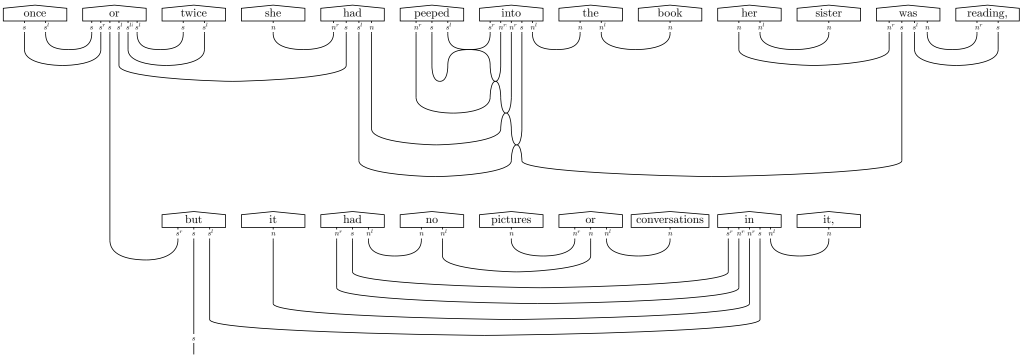
Mark Steedman. 2000. *The Syntactic Process*. MIT Press.

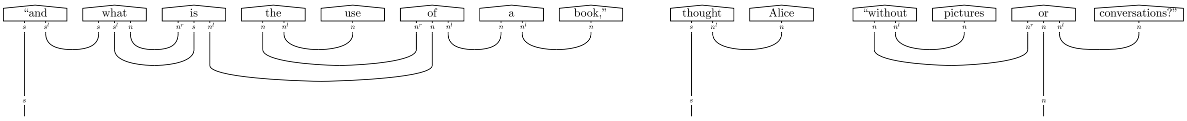
Krishnamurti Vijay-Shanker and David J Weir. 1994. The equivalence of four extensions of context-free grammars. *Mathematical systems theory*, 27(6):511–546.

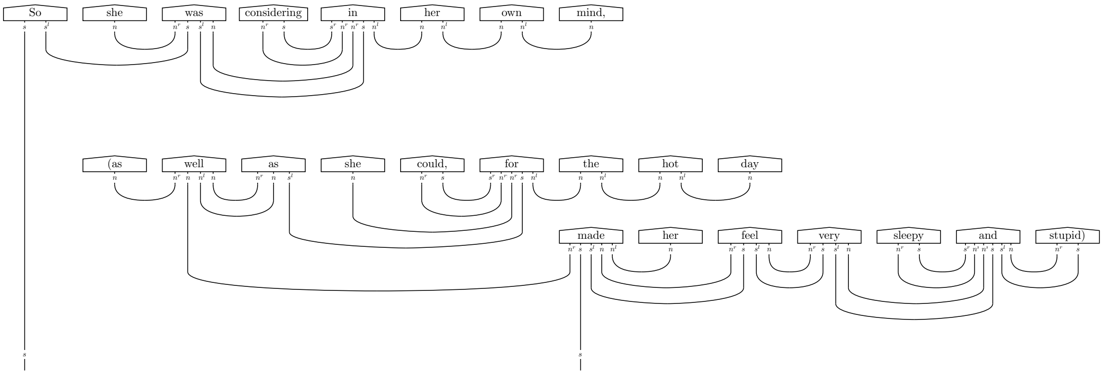
Masashi Yoshikawa, Hiroshi Noji, and Yuji Matsumoto. 2017. A* ccg parsing with a supertag and dependency factored model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 277–287. Association for Computational Linguistics.

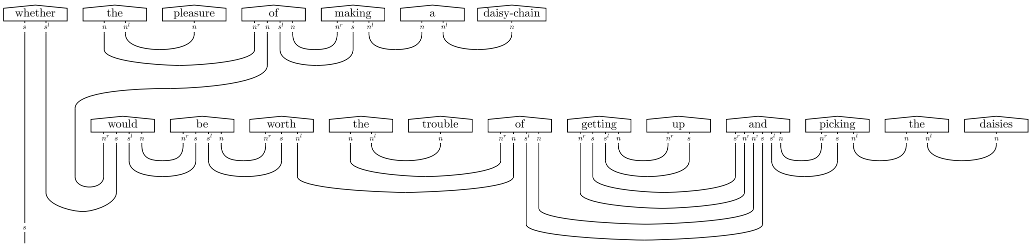
A Appendix

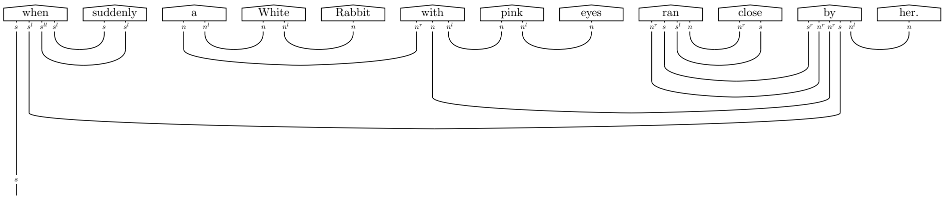

Alice was beginning to get very tired of sitting by her sister on the bank,
and of having nothing to do:


once or twice she had peeped into the book her sister was reading,
but it had no pictures or conversations in it,


“and what is the use of a book,” thought Alice “without pictures or conversations?”


So she was considering in her own mind,
(as well as she could, for the hot day
made her feel very sleepy and stupid)


whether the pleasure of making a daisy-chain
would be worth the trouble of getting up and picking the daisies


when suddenly a White Rabbit with pink eyes ran close by her.

Grammar Equations

Bob Coecke
Oxford-based QNLP-team
Cambridge Quantum Computing Ltd.
bob.coecke@cambridgequantum.com

Vincent Wang
Department of CS
University of Oxford
vincent.wang@stcatz.ox.ac.uk

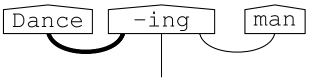
Abstract

Diagrammatically speaking, grammatical calculi such as pregroups provide wires between words in order to elucidate their interactions, and this enables one to verify grammatical correctness of phrases and sentences. In this paper we also provide wirings within words. This will enable us to identify grammatical constructs that we expect to be either equal or closely related. Hence, our work paves the way for a new theory of grammar, that provides novel 'grammatical truths'. We give a nogo-theorem for the fact that our wirings for words make no sense for preordered monoids, the form which grammatical calculi usually take. Instead, they require diagrams – or equivalently, (free) monoidal categories.

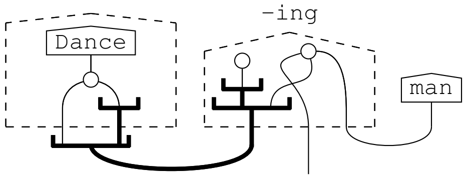
1 Introduction

Grammatical calculi (Lambek, 1958; Grishin, 1983; Lambek, 1999) enable one to verify grammatical correctness of sentences. However, there are certain grammatical constructs that we expect to be closely related, if not the same, but which grammatical calculi fail to identify. We will focus on pregroups (Lambek, 2008), but the core ideas of this paper extend well beyond pregroup grammars, including CCGs (Steedman, 1987), drawing on the recent work in (Yeung and Kartsaklis, 2021) that casts CCGs as augmented pregroups.

In this paper we both modify and extend grammatical calculi, by providing so-called 'internal wirings' for a substantial portion of English. Diagrammatically speaking, while grammatical calculi provide wires between words in order to elucidate their interactions, we also provide wirings within words. For example, a *pregroup diagram* for the phrase:



will become:



We show how these additional wirings enable one to identify grammatical constructs that we expect to be closely related. Providing these internal wirings in particular involves decomposing basic types like sentence-types over noun-types, and this decomposition may vary from sentence to sentence. Hence our refinement of grammar-theory also constitutes a departure from some of the practices of traditional grammatical calculi.

Additional structure for grammatical calculi was previously introduced by providing semantics to certain words, for example, quantifiers within Montague semantics (Montague, 1973). This is not what we do. We strictly stay within the realm of grammar, and grammar only. Hence, our work paves the way for a new theory of grammar, that provides novel 'grammatical truths'.

Usually grammatical calculi take the form of preordered monoids (Coecke, 2013). However, the internal wirings cannot be defined at the poset level, for which we provide a nogo-theorem. Hence passing to the realm of diagrammatic representations – which correspond to proper free monoidal categories – is not just a convenience, but a necessity for this work. They moreover provide a clear insight in the flow of meanings.

Internal wirings were proposed within the DisCoCat framework (Coecke et al., 2010), for relative pronouns and verbs (Sadrzadeh et al., 2013, 2016; Grefenstette and Sadrzadeh, 2011; Kartsaklis and Sadrzadeh, 2014; Coecke et al., 2018; Coecke, 2019; Coecke and Meichanetzidis, 2020). They are made up of 'spiders' (a.k.a. certain

Proceedings of the 2021 Workshop on Semantic Spaces at the Intersection of NLP, Physics, and Cognitive Science, pages 32–41
June 16th, 2021. ©2021 Association for Computational Linguistics

Frobenius algebras) (Coecke et al., 2013; Coecke and Kissinger, 2017). We point out a shortcoming of those earlier proposed internal wirings, and fix them by introducing a 'wrapping gadget', that forces certain wires to stay together. This reintroduces composite types such as sentence types.

What we present here is only part of the full story. For the latter we refer to a forthcoming much longer paper (Coecke and Wang), which besides providing many more internal wirings than given here, also uses them to provide bureaucracy-free grammar as circuits, the equivalence classes for the equations introduced here. These circuits also have direct practical applications within natural language processing – see e.g. (Coecke et al., 2020).

2 Statement of the problem

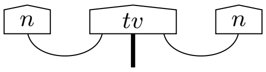
For our purposes, a pregroup has a set of 'basic types' $n, s, \ldots$ each of which admit left and right inverses ${}^{-1}n$ and n^{-1}. Each grammatical type is assigned a string of these, e.g. a transitive verb in English gets: $tv = {}^{-1}n \cdot s \cdot n^{-1}$. The inverses 'cancel out' from one direction:

$$n \cdot {}^{-1}n \to 1 \qquad\qquad n^{-1} \cdot n \to 1 \qquad (1)$$

A sentence is grammatical if when taking the string of all of its grammatical types, the inverses cancel to leave a special, 'final', basic type s (for sentence), like here for $n \cdot tv \cdot n$:

$$\begin{aligned} & n \cdot \left({}^{-1}n \cdot s \cdot n^{-1}\right) \cdot n \\ \xrightarrow{(assoc.)} \; & \left(n \cdot {}^{-1}n\right) \cdot s \cdot \left(n^{-1} \cdot n\right) \\ \xrightarrow{(1)} \; & 1 \cdot s \cdot 1 \\ \xrightarrow{(unit)} \; & s \end{aligned}$$

This calculation can be represented diagrammatically:



(2)

Now consider the following examples:

`Alice likes the flowers that Bob gives Claire`

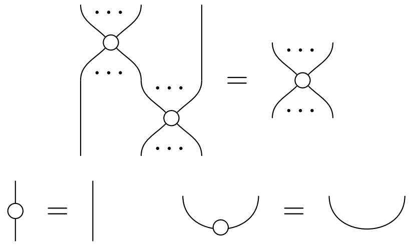
`Bob gives Claire the flowers that Alice likes`

The pregroup diagrams now look as in Figure (1). Without any further context the factual data conveyed by these two sentences is the same.[1] How can we formally establish this connection between the two sentences?

3 Rewriting pregroup diagrams via internal wirings

What is needed are 'internal wirings' of certain words, that is, not treating these words as 'black boxes', but specifying what is inside, at least to some extent. Equationally speaking, they provide a congruence for pregroup diagrams, and we can establish equality by means of topological deformation.

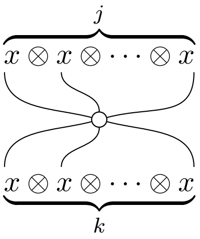
For constructing these internal wirings we make use of 'spiders' (Coecke et al., 2013; Coecke and Kissinger, 2017) (a.k.a. Frobenius algebras (Carboni and Walters, 1987; Coecke and Paquette, 2011)). One can think of these spiders as a generalisation of wires to multi-wires, as rather than having two ends, they can have multiple ends. Still, all they do, like wires, is connect stuff, and when you connect connected stuff to other connected stuff (a.k.a. 'spider-fusion'):



We presented internal wiring in terms of pregroup diagrams. This is because they do not make sense in terms of symbolic pregroups presented as preordered monoids:

Theorem 3.1. A pregroup with spiders is trivial. Concretely, given a preordered monoid $(X, \leq, \otimes)$ with unit 1, if for $x \in X$ there are spiders with x as its legs, then $x \simeq 1$.

Proof. Having spiders on x means that for all $j, k \in \mathbb{N}$ there exists:



j
$x \otimes x \otimes \cdots \otimes x$
$x \otimes x \otimes \cdots \otimes x$
k

[1] Additional context could indicate a causal connection between the two parts of the sentence, which could result in the two sentences having different meanings – see (Coecke and Wang) for more details.

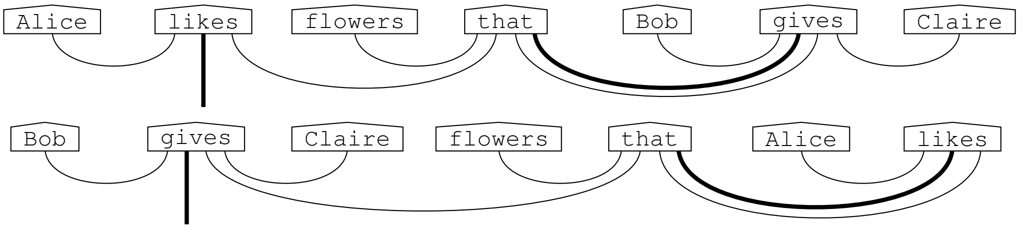


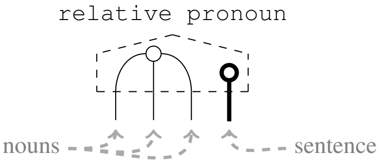
Figure 1

that is, we have $\bigotimes^j x \le \bigotimes^k x$. So in particular, $x \le 1$ and $1 \le x$, so $x \simeq 1$. □

Hence this paper requires diagrams in a fundamental manner.[2]

3.1 Internal wiring for relative pronouns

For relative pronouns we start with the internal wirings that were introduced in (Sadrzadeh et al., 2013, 2016):

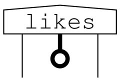


(3)

Substituting this internal wiring in the pregroup diagrams we saw above: and permuting the boxes a bit, more specifically, swapping Bob gives Claire and Alice likes in the 2nd diagram, the two diagrams start to look a lot more like each other, as can be seen in figure 2. Their only difference is a twist which vanishes if we take spiders to be commutative,[3] and either a loose sentence-type wire coming out of the verb likes in the first diagram, versus coming out the verb give in the second diagram, the other verb having its sentence type deleted.

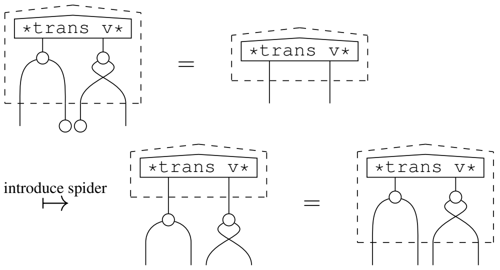
3.2 Internal wiring for verbs

The deleting of sentence-types of verbs:



(4)

by the internal wiring of relative pronouns seems to prevent us from bringing the diagrams of Figure (2) any closer to each other. However, this irreversibility does not happen for a particular kind of internal wiring for the verb (Grefenstette and Sadrzadeh, 2011; Kartsaklis and Sadrzadeh, 2014; Coecke, 2019; Coecke and Meichanetzidis, 2020), here generalised to the non-commutative case as demonstrated by the transitive verb in Figure (5). For transitive verbs in spider-form, if the sentence type gets deleted we can bring back the original form by copying the remaining wires:



So nothing was ever lost. To conclude, for the internal wiring of verbs proposed above, the copying and deleting spiders now guarantee that in (4) nothing gets lost.

3.3 Rewriting pregroup diagrams into each other

Introducing the internal wiring (5) and deleting all outputs, our example sentences now appear as in the first two diagrams of Figure (3). Except for the twist the two pregroup diagrams have become the same. As we have no outputs anymore, let's just stick in a copy-spider for all nouns, and then after fusing all deletes away, our sentence is transformed into the third diagram of figure 3.

The recipe we followed here is an instance of a general result that allows us to relate sentences for a substantial portion of English, by providing internal wirings for that fragment. In Section 5 we will provide internal wirings some grammatical word classes – in (Coecke and Wang) we provide a much larger catalog – that will generate

[2] One SEMSPACE referee requested a category-theoretic generalisations of the above stated nogo-theorem. Such a generalisation has been provided on Twitter following our request (Hadzihasanovic, 2021). Our result should also not be confused with the (almost contradictory sounding) following one, which states that pregroups are spiders in the category of preordered relations (Pavlovic, 2021).

[3] Non-commutativity can be seen as a witness for the fact that within a broader context the two sentences may defer in meaning due to a potential causal connection between its two parts – see (Coecke and Wang) for more details.

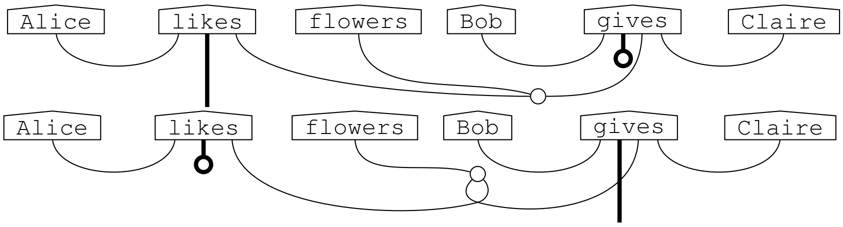


Figure 2

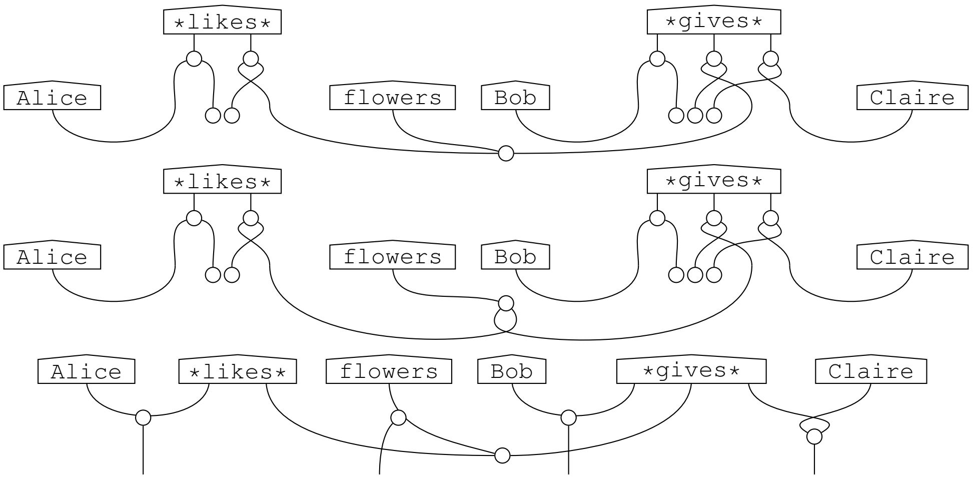


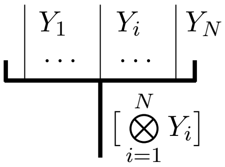
Figure 3

correspondences between grammatical constructs, just like the one established above. In Section 6 we provide some further examples of this. In (Coecke and Wang) we also provide a normal induced by grammar equations.

4 The wrapping gadget

Above in (5) we saw that sentence wires were decomposed into noun wires. However, for pregroup proofs it is important to know that those wires do belong together, so we need to introduce a tool that enables us to express that they belong together.

Definition 4.1. The wrapping gadget forces a number of wires to be treated as one, i.e. it wraps them, and is denoted as follows:



By unfolding we mean dropping the restrictions imposed by the wrapping gadget. Cups and spiders carry over to wrapped wires in the expected way, following the conventions of (Coecke and Kissinger, 2017).

In fact, in the case of relative pronouns simply wrapping the noun-wires making up the sentence type isn't enough, as the counterexample in Figure (4) shows.

5 Some more internal wirings

We now provide internal wirings for some grammatical word classes that feature in the examples of the next section. We distinguish between 'content words', like the verbs in (5), and 'functional words', like the relative pronouns in (7).

5.1 Content words

We provide internal wirings for intransitive and transitive verbs in Figure (5), and predicative and attributive adverbs for transitive verbs in Figure (6).

5.2 Functional words

We provide internal wirings for subject and object relative pronouns for intransitive verbs, and a passive-voice construction 'word' for transitive verbs in Figure (7).

6 Proof-of-concept

We provide a number of examples of how the internal wirings proposed above enable us to re-

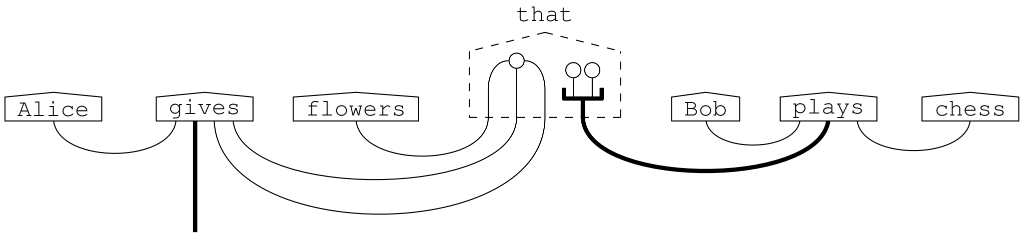


Figure 4: The deleting of the sentence type of `plays` belongs together with the noun-wire now connecting the relative pronoun with `gives`, like in Figure (1). This is enforced by the internal wiring of the object relative pronoun in Figure (7)

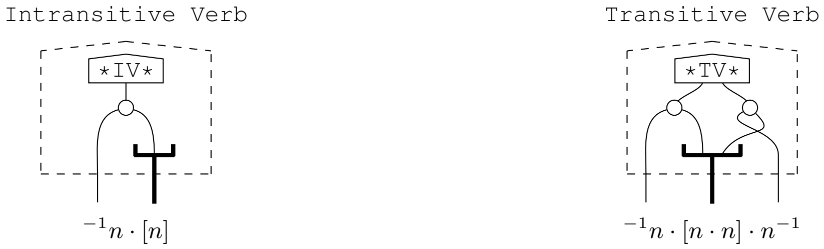


Figure 5

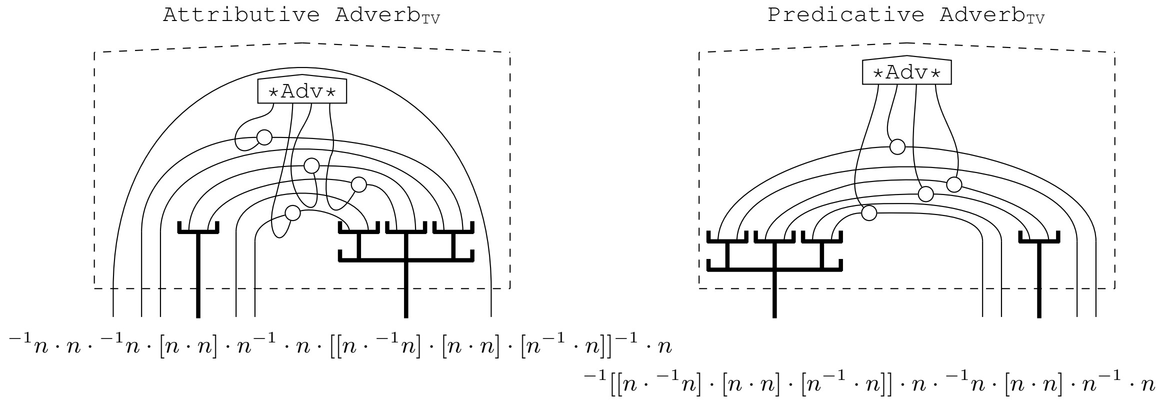

Attributive Adverb$_{\text{TV}}$
⋆Adv⋆
$^{-1}n \cdot n \cdot {}^{-1}n \cdot [n \cdot n] \cdot n^{-1} \cdot n \cdot [[n \cdot {}^{-1}n] \cdot [n \cdot n] \cdot [n^{-1} \cdot n]]^{-1} \cdot n$
Predicative Adverb$_{\text{TV}}$
⋆Adv⋆
$^{-1}[[n \cdot {}^{-1}n] \cdot [n \cdot n] \cdot [n^{-1} \cdot n]] \cdot n \cdot {}^{-1}n \cdot [n \cdot n] \cdot n^{-1} \cdot n$

Figure 6

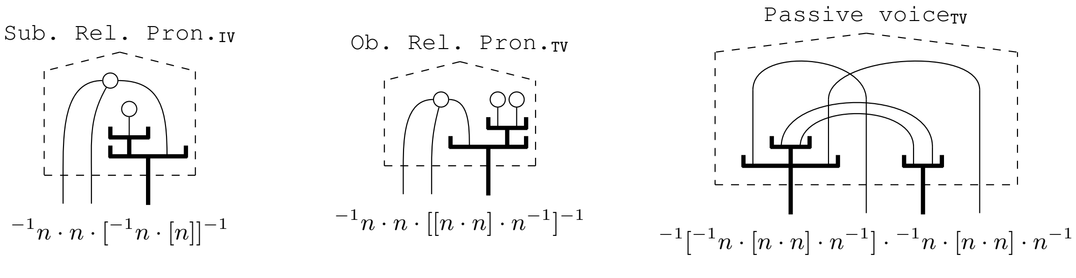

Sub. Rel. Pron.$_{\text{IV}}$
$^{-1}n \cdot n \cdot [{}^{-1}n \cdot [n]]^{-1}$
Ob. Rel. Pron.$_{\text{TV}}$
$^{-1}n \cdot n \cdot [[n \cdot n] \cdot n^{-1}]^{-1}$
Passive voice$_{\text{TV}}$
$^{-1}[{}^{-1}n \cdot [n \cdot n] \cdot n^{-1}] \cdot {}^{-1}n \cdot [n \cdot n] \cdot n^{-1}$

Figure 7

late different grammatical constructs just as in the case of what the relative pronoun and verb internal wirings did for the sentences in Figure (1). We omit the pregroup typings, instead depicting the pregroup diagrams directly. Wrapping gadgets correspond to bracketing pregroup types together.

References

A. Carboni and R. F. C. Walters. 1987. Cartesian bicategories I. *Journal of Pure and Applied Algebra*, 49:11–32.

B. Coecke. 2013. An alternative Gospel of structure: order, composition, processes. In C. Heunen, M. Sadrzadeh, and E. Grefenstette, editors, *Quantum Physics and Linguistics.*, pages 1 – 22. Oxford University Press. ArXiv:1307.4038.

B. Coecke. 2019. The mathematics of text structure. ArXiv:1904.03478.

B. Coecke, G. de Felice, K. Meichanetzidis, and A. Toumi. 2020. Foundations for near-term quantum natural language processing.

B. Coecke and A. Kissinger. 2017. *Picturing Quantum Processes. A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press.

B. Coecke, M. Lewis, and D. Marsden. 2018. Internal wiring of cartesian verbs and prepositions. In Procs. of the 2018 Workshop on *Compositional Approaches in Physics, NLP, and Social Sciences*, volume 283 of *Electronic Proceedings in Theoretical Computer Science*, pages 75–88.

B. Coecke and K. Meichanetzidis. 2020. Meaning updating of density matrices. *arXiv:2001.00862*.

B. Coecke and É. O. Paquette. 2011. Categories for the practicing physicist. In B. Coecke, editor, *New Structures for Physics*, Lecture Notes in Physics, pages 167–271. Springer. arXiv:0905.3010.

B. Coecke, D. Pavlović, and J. Vicary. 2013. A new description of orthogonal bases. *Mathematical Structures in Computer Science, to appear*, 23:555–567. arXiv:quant-ph/0810.1037.

B. Coecke, M. Sadrzadeh, and S. Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. In J. van Benthem et al., editor, *A Festschrift for Jim Lambek*, volume 36 of *Linguistic Analysis*, pages 345–384. Arxiv:1003.4394.

B. Coecke and V. Wang. Distilling grammar into circuits (or, de-humanising grammar for efficient machine use).

E. Grefenstette and M. Sadrzadeh. 2011. Experimental support for a categorical compositional distributional model of meaning. In *The 2014 Conference on Empirical Methods on Natural Language Processing.*, pages 1394–1404. ArXiv:1106.4058.

V.N. Grishin. 1983. On a generalization of the Ajdukiewicz-Lambek system. In *Studies in nonclassical logics and formal systems*, pages 315–334. Nauka, Moscow.

A. Hadzihasanovic. 2021. Twitter reply.

D. Kartsaklis and M. Sadrzadeh. 2014. A study of entanglement in a categorical framework of natural language. In *Proceedings of the 11th Workshop on Quantum Physics and Logic (QPL)*. Kyoto Japan.

J. Lambek. 1958. The mathematics of sentence structure. *American Mathematics Monthly*, 65.

J. Lambek. 1999. Type grammar revisited. *Logical Aspects of Computational Linguistics*, 1582.

J. Lambek. 2008. From word to sentence. *Polimetrica, Milan*.

R. Montague. 1973. The proper treatment of quantification in ordinary English. In *Approaches to natural language*, pages 221–242. Springer.

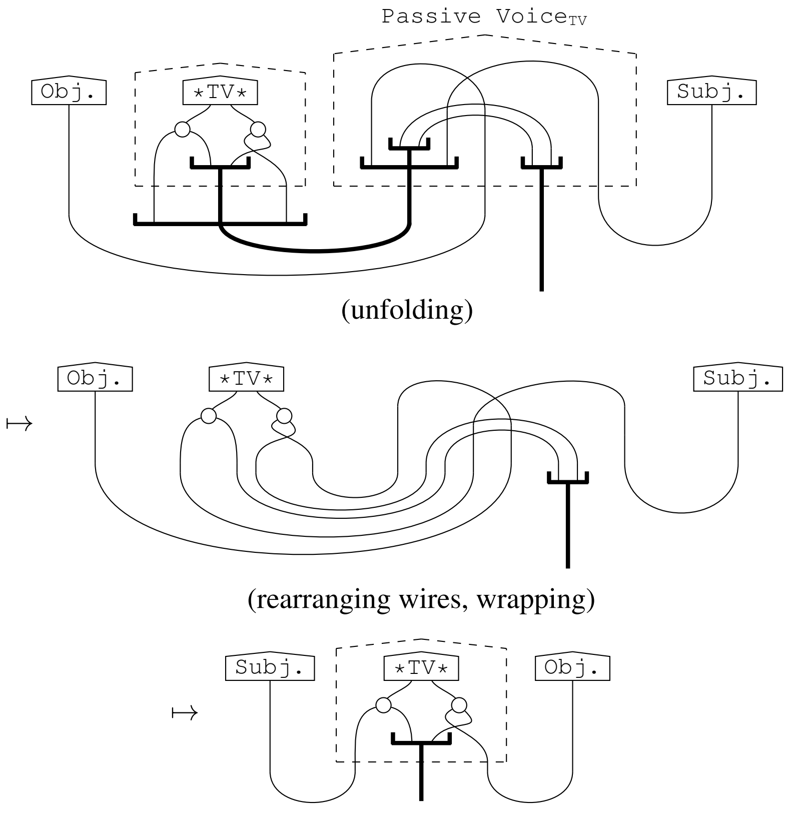
D. Pavlovic. 2021. Lambek pregroups are frobenius spiders in preorders. *arXiv preprint arXiv:2105.03038*.

M. Sadrzadeh, S. Clark, and B. Coecke. 2013. The Frobenius anatomy of word meanings I: subject and object relative pronouns. *Journal of Logic and Computation*, 23:1293–1317. ArXiv:1404.5278.

M. Sadrzadeh, S. Clark, and B. Coecke. 2016. The Frobenius anatomy of word meanings II: possessive relative pronouns. *Journal of Logic and Computation*, 26:785–815. ArXiv:1406.4690.

M. Steedman. 1987. Combinatory grammars and parasitic gaps. *Natural Language & Linguistic Theory*, 5:403–439.

R. Yeung and D. Kartsaklis. 2021. A CCG-based version of the DisCoCat framework. *Accepted for SEMSPACE*.



(unfolding)

(rearranging wires, wrapping)

Figure 8: **We relate:** `Alice` $\underbrace{\texttt{is bored by}}_{\text{passive voice}}$ `the class` **to:** `The class bores Alice`

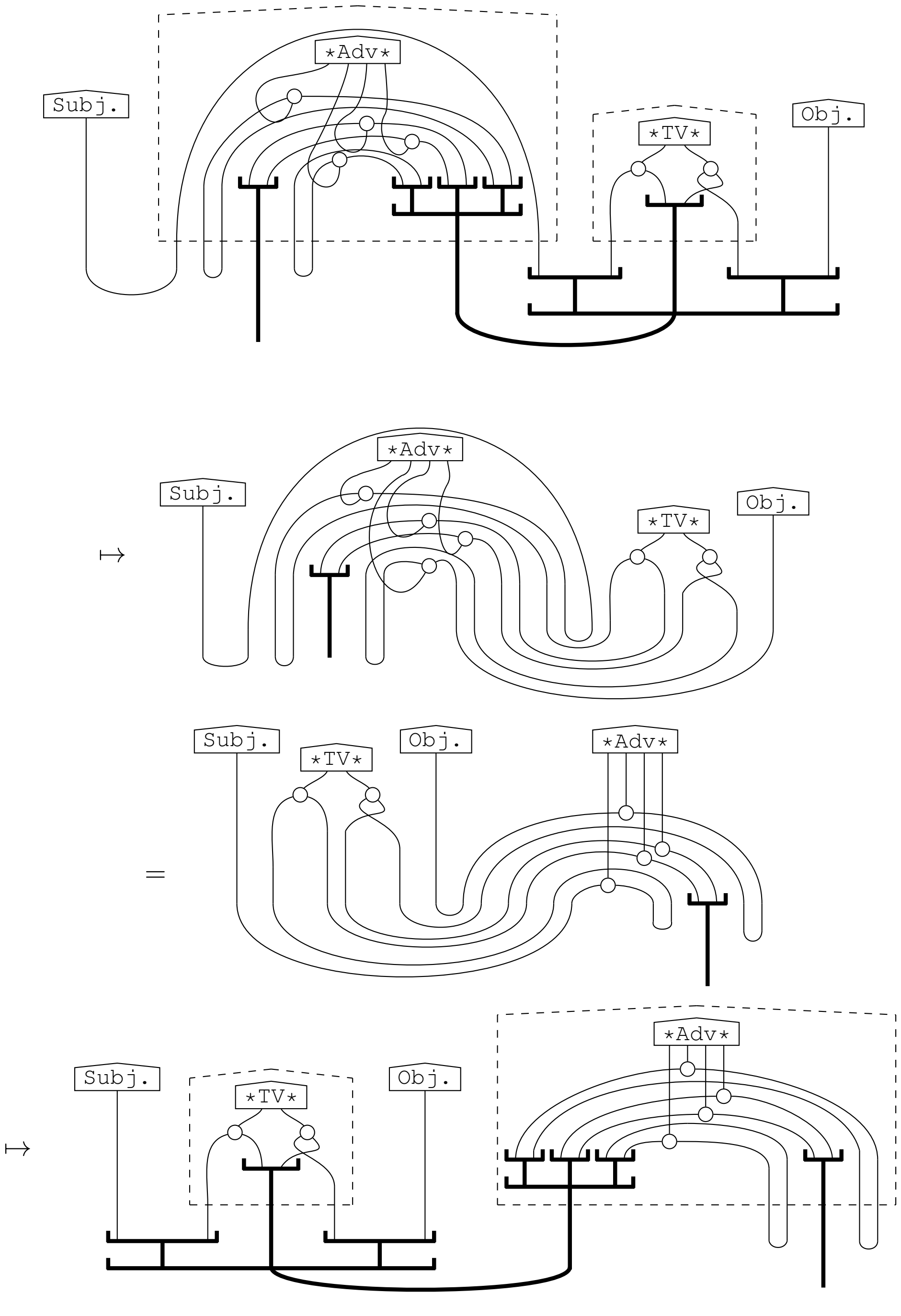
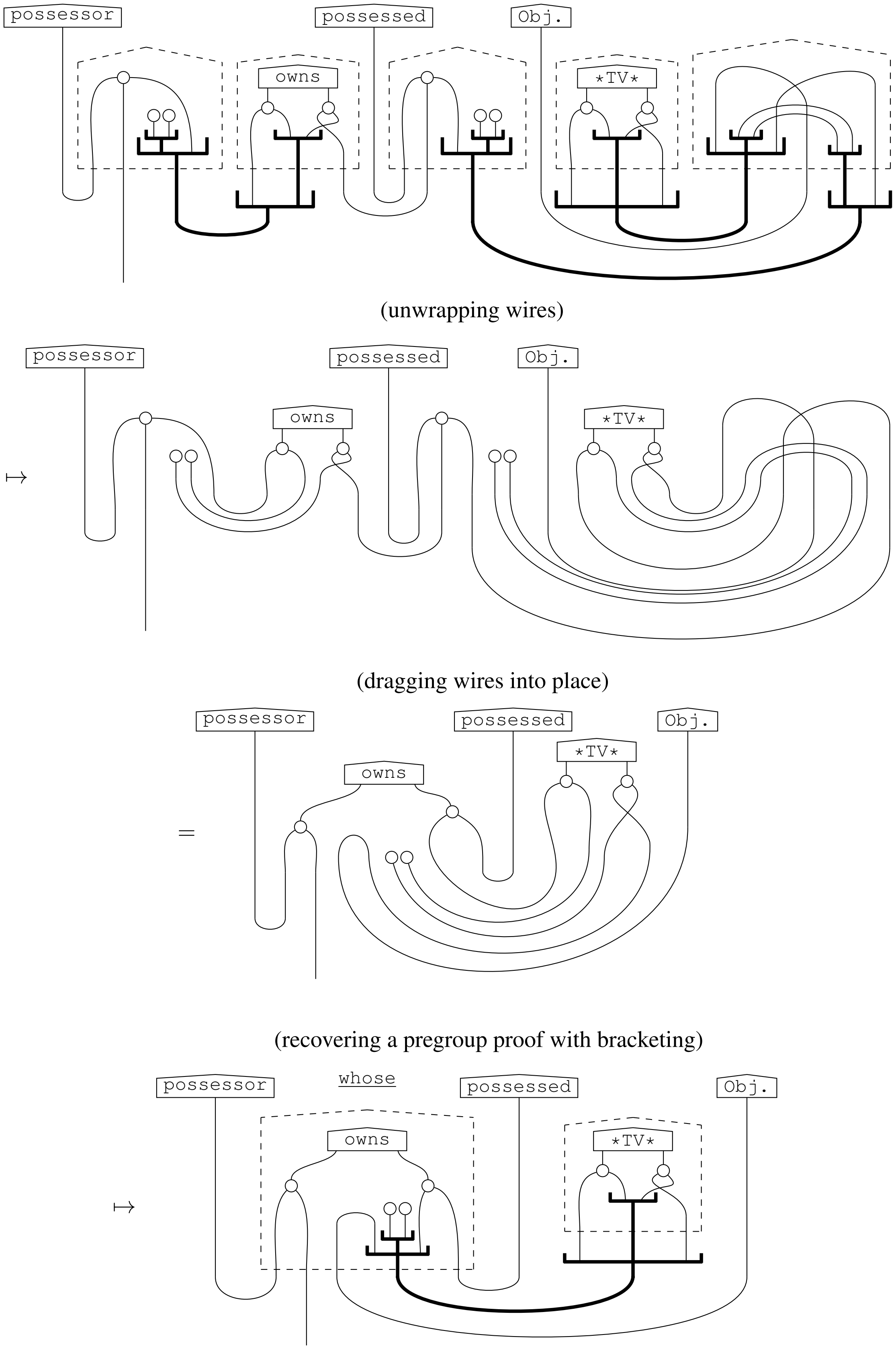


Figure 9: **We relate:** `Alice washes Fido gently` **to:** `Alice gently washes Fido`



(unwrapping wires)

(dragging wires into place)

(recovering a pregroup proof with bracketing)

Figure 10: **From:** author that owns book that John (was) entertain(s) -ed (by) **we derive a possessive relative pronoun:** author whose book entertained John

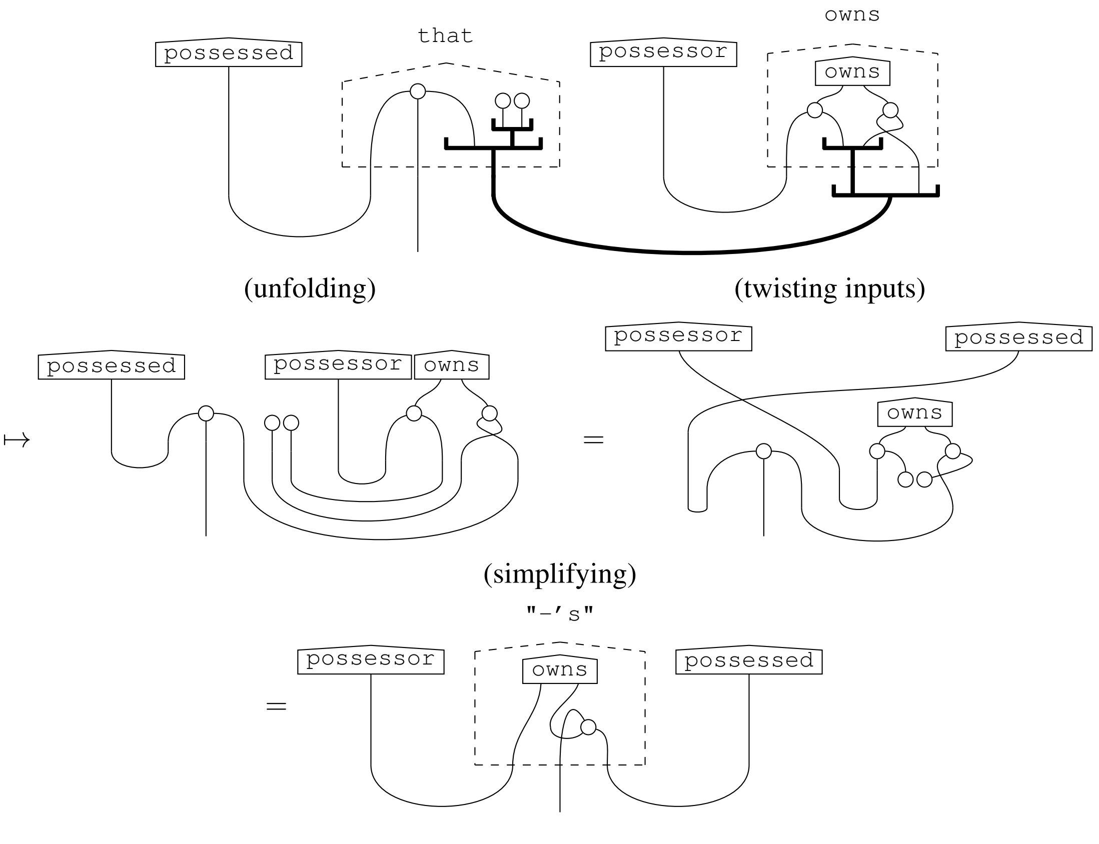


Figure 11: **From:** (possessed) that (possessor) owns **we derive the possessive modifier:** (possessor) ’s (possessed)

On the Quantum-like Contextuality of Ambiguous Phrases

Daphne Wang **Mehrnoosh Sadrzadeh**
University College London

Samson Abramsky
Oxford University

Víctor H. Cervantes
University of Illinois
at Urbana-Champaign

Abstract

Language is contextual as meanings of words are dependent on their contexts. Contextuality is, concomitantly, a well-defined concept in quantum mechanics where it is considered a major resource for quantum computations. We investigate whether natural language exhibits any of the quantum mechanics' contextual features. We show that meaning combinations in ambiguous phrases can be modelled in the sheaf-theoretic framework for quantum contextuality, where they can become possibilistically contextual. Using the framework of Contextuality-by-Default (CbD), we explore the probabilistic variants of these and show that CbD-contextuality is also possible.

1 Introduction

We start with a peculiar observation: even though polysemy and homonymy are common phenomena of natural language, i.e. many words have more than one meaning, this does not create a considerable obstacle in our day-to-day comprehension of texts and conversations. For example, the word *charge* has 40 different senses in English according to WordNet, however, its meaning in the sentence *The bull charged.* remains fairly unambiguous. On the other hand, polysemy and word sense disambiguation are computationally difficult tasks and is amongst the challenges faced by linguists (Rayner and Duffy, 1986; Pickering and Frisson, 2001; Frazier and Rayner, 1990).

The emergence of the field of quantum methods in Natural Language Processing offers promising leads for introducing quantum methods to classical NLP tasks, e.g. language modelling (Basile and Tamburini, 2017), distributional semantics (Blacoe et al., 2013), mental lexicon (Bruza et al., 2009), narrative structure (Meichanetzidis et al., 2020), emotion detection (Li et al., 2020b), and classification (Liu et al., 2013; Li et al., 2020a).

Distributional semantics is a natural language semantic framework built on the notion of contextuality. Herein, frequencies of co-occurrences of words are computed from their contexts and the resulting vector representations are used in automatic sense discrimination (Schütze, 1998). An issue with this framework is that the grammatical structure of phrases and sentences is ignored and the focus is mainly on large-scale statistics of data. Oppositely, even though the interaction between context and syntax has been studied in the past (Barker and Shan, 2015), no distributional data has been considered in them. Finally, distributional and compositional models of language have been proposed (Coecke et al., 2010), small experiments have been implemented on quantum devices (Meichanetzidis et al., 2020), and choices of meaning in concept combinations have been analysed using superposition (Bruza et al., 2015; Piedeleu et al., 2015). Our work complements these lines of research by modelling the underlying structure of contextuality using distributional data.

We investigate the contextual nature of meaning combinations in ambiguous phrases of natural language, using instances of the data gathered in psycholinguistics (Pickering and Frisson, 2001; Tanenhaus et al., 1979; Rayner and Duffy, 1986), frequencies mined from large scale corpora (BNC, 2007; Baroni et al., 2009), and models coming from the sheaf-theoretic framework (Abramsky and Brandenburger, 2011; Abramsky and Hardy, 2012) and the Contextuality-By-Default (CbD) theory (Dzhafarov and Kujala, 2016). We consider phrases with two ambiguous words, in subject-verb and verb-object predicate-argument structures and find instances of logical and CbD contextuality.

The structure of the paper is as follows. We start by introducing the main concepts behind quantum contextuality (section 2). We then introduce the sheaf-theoretic framework and logical contextual-

Proceedings of the 2021 Workshop on Semantic Spaces at the Intersection of NLP, Physics, and Cognitive Science, pages 42–52
June 16th, 2021. ©2021 Association for Computational Linguistics

ity (section 3), before applying it to possibilistic natural language models (section 4 and 5). In section 6 and 7, we discuss probabilistic models and signalling in natural language respectively. In section 8, we offer the possibility of studying contextuality in signalling models via the Contextuality-by-Default framework and discuss two CbD contextual examples that we found. We then close the paper with insights on how to perform a large scale experiment and the possibility of finding more CbD-contextual examples in natural language.

2 Quantum contextuality

Early critics of quantum mechanics claimed that quantum theory was not complete (Einstein et al., 1935), but instead was subject to unobserved hidden variables, and claimed that any physical theory should satisfy local realism. By local realism, one means that in a "complete" physical theory, the global behaviour of a system is entirely, and deterministically, fixed by a set of local variables. However, the well-known Bell theorem (Bell, 1964), supported by experimental data (Hensen et al., 2015), shows that a description of quantum mechanics cannot comply with local realism; if quantum systems need to have a "reality" independent of the observers (realism), one should allow interactions between systems to be unrestricted spatially (non-local).

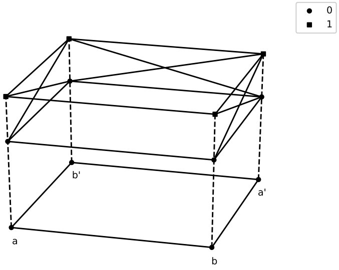
The Bell inequality offers a proof by contradiction that one cannot extend the probabilistic models obtained from observations of quantum systems to a deterministic hidden-variable model. In Kochen and Specker (1967), the authors prove a stronger statement about the existence of hidden-variable models via a logical argument. This more general result provides a description of *contextuality* as it is understood in quantum mechanics.

3 Presheaves and logical contextuality

The sheaf-theoretic framework of contextuality starts from the observation that contextuality in quantum mechanics translates to the impossibility of finding a global section in special presheaves. In other words, a model is contextual if some of its *local* features cannot be extended *globally*.

A	B	(0,0)	(0,1)	(1,0)	(1,1)
a	b	1/2	0	0	1/2
a	b'	3/8	1/8	1/8	3/8
a'	b	3/8	1/8	1/8	3/8
a'	b'	1/8	3/8	3/8	1/8

(a) Probability distributions



(b) Bundle diagram of the logical model

Figure 1: Empirical model associated with the measurement of the bipartite state $|\Psi\rangle = \frac{1}{\sqrt{2}}\Big(|00\rangle + |11\rangle\Big)$ with local measurements $a, b = |1\rangle\langle 1|_{A,B}$ and $a', b' = |\phi\rangle\langle\phi|_{A,B}$ where $|\phi\rangle = \frac{\sqrt{3}}{2}|0\rangle + i\frac{1}{2}|1\rangle$.

The presheaves considered in the framework developed by Abramsky et. al. (Abramsky and Brandenburger, 2011; Abramsky et al., 2015) are so-called distribution presheaves on events. An *empirical model* corresponds to the experiment that is undertaken; it consists of the list of measurements that can be made, which measurements can be made together and what are the associated probability distributions. For example, Fig. 1 can represent a standard Bell experiment where the list of measurements is the list of all local measurements that can be made on the two qubits involved in the experiment, i.e. $\{a, a', b, b'\}$, under the condition that each laboratory performs exactly one measurement at each run of the experiment, e.g. (a, b) can be a joint measurement, but (a, a') cannot. The distribution presheaf then associates the observed (or theoretical) probabilities for the global measurement outcomes, that is, the joint outcomes of both parties in a Bell scenario, for each measurement context. In this framework, a *global assignment* corresponds to an assignment of an outcome for every local measurement. A *global section* will on the other hand represent a distribution defined on all global assignments, which is consistent with all the observed probabilities.

The framework of Abramsky et al. (Abramsky and Brandenburger, 2011; Abramsky et al., 2015; Abramsky and Hardy, 2012) also introduces a stronger type of contextuality, called *logical* (or *possibilistic*) *contextuality*. Indeed, they have found that the contextuality of some systems can be established from the *support* of each of the context-dependent distributions. These are referred to as *possibilistic empirical models*. In these models, we are only interested in whether an outcome of a local measurement (given a global measurement context) is *possible*. A consistent global assignment will then be an assignment of a possible outcome to every measurement, and hence can be represented by a logical statement about a subsystem. A global section will then be a disjunction of consistent global assignments that describes the entirety of the model. Hence, one can prove logical contextuality, i.e. the impossibility of being able to write such a logical statement about the system, by finding a locally possible outcome that cannot be extended to a consistent global assignment.

For small systems, it is convenient to represent possibilistic models by bundle diagrams (Abramsky et al., 2015). In these diagrams, we represent each of the local measurements by a vertex. There is an edge[1] between every two of these vertices if the joint measurement is possible. We then depict, for each individual measurements, the set of possible outcomes as a set "sitting" on top of the associated vertex. Similarly, an edge is added between two of the "outcome"-vertices if the joint measurement has a non-zero probability (e.g. see Fig. 1b). In particular, global assignments can be seen in these bundle diagrams as shapes going through exactly one outcome for each of the measurements that mirror the structure of the base (measurements). In Fig. 1b for example, global assignments correspond to connected loops.

The sheaf-theoretic framework relies on the fact that the described distribution presheaf is indeed a presheaf. That is, the distributions associated with measurement contexts that intersect at a local measurement (i.e. two contexts where at least one party performs the same measurement) agree on their restrictions. These are here defined as the marginals of the distributions of interest. This requirement coincides with the *non-signalling* condition in quantum mechanics. This condition is stated for possibilistic models by requiring that the supports of intersecting distributions coincides.

As we will see, many empirical models from natural language will be *signalling*. That is also the case for many behavioural and psychological experiments (see e.g. Bruza et al. (2015); Dzhafarov et al. (2016)), and in fact, there is no reason why natural language systems *should* be non-signalling and we will discuss this issue in sections 7.

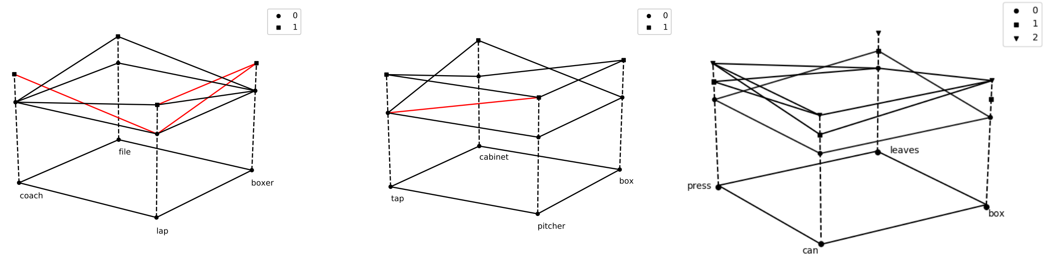
4 Contextuality and ambiguity in natural language

We are interested in studying the influence of the context on the process of meaning selection in ambiguous phrases. Indeed, homonymy and polysemy in natural language give rise to an interpretation for context-dependent probability distributions. Probabilities will correspond to the likelihood that a certain meaning of a word is selected in the context of interest. By analogy with quantum contextuality, existence of contextual natural language examples confirms that the context in which words are found plays a non-trivial role in the selection of an appropriate interpretation for them and the following question arises: given that a certain interpretation of a word is selected within a certain context, can we use this information to deduce how the same word may be interpreted in a different context (e.g. in different phrases) in the corpus?

Our intuition is that this is not the case. Consider the ambiguous adjective *green*: this either refers to the colour of its modifier (e.g. *a green door*), or the environmental-friendly nature of it (e.g. *the Green party*). Now, if we consider an unambiguous adjective such as *new*, then trivially, the same interpretations of *new* can be selected in both of the phrases *new paint* and *new policy*. This, however, does not imply that the same interpretations of *green* will be selected in *green paint* and *green policy*. With this intuition in mind, we start by considering the basic structure of ambiguous phrases of English by considering only the support of probability distributions attributed to these phrases and for now appeal to our common sense to determine the values of these supports.

In the first part of the paper, we consider a structure similar to Bell scenarios with multiple parties, or agents, each of which will choose one measurement context from a predetermined set. A "measurement" will be associated with each word and will return the activated meaning according to a fixed encoding. For example the two meanings of *green* could be encoded as: *relative to colour* $\mapsto 0$, *environmental-friendly* $\mapsto 1$. In a given context, multiple ambiguous words will be allowed to "interact" and form a phrase. A measurement context will then be labelled by the words in this phrase. The interaction will be dictated by some predetermined rules, such as which part-of-speech

[1] More generally simplices if multiple measurements are carried out simultaneously.



(a) $\{coach, boxer\} \times \{lap, file\}$ (b) $\{tap, box\} \times \{pitcher, cabinet\}$ (c) $\{press, box\} \times \{can, leaves\}$

Figure 2: Instances of bundle diagrams arising from ambiguous phrases. The local assignments which cannot be extended to a global one are depicted in red.

each word will correspond to. For each global measurement context, the recorded activated meanings will then form a joint distribution. These distributions can be represented in the form of an empirical model as described in section 3. In order to obtain a valid empirical model, all the possible combinations of words need to make sense. For example, take two parties A and B such that A chooses an adjective in the set {*green*,*new*} and B chooses its modifier within the set {*paint*, *policy*}. All the combinations of A and B are possible, i.e. phrases *green paint*, *green policy*, *new paint* and *new policy* all make sense and can indeed be found in natural language corpora. However, if the set of adjectives is changed to {*blue*, *new*}, we will face a problem since the phrase *blue policy* does not make much sense and we could not find any occurrence of it in the corpora considered in this paper.[2] In order to keep the models and computations simple, we work with 2-word phrases, where each word of the phrase is ambiguous. From the analogy with Bell scenarios, this means that we are working with bipartite scenarios (see Fig. 3). The set of ambiguous words is taken from experimental data sets from the studies: Pickering and Frisson (2001); Rayner and Duffy (1986); Tanenhaus et al. (1979).

In sections 5.4 and 6.4, we introduce another kind of experiment which departs from Bell scenarios. Measurements of these examples have the same interpretation as before, but the focus is on combinations involving a single verb and a single noun for which both of subject-verb and verb-object phrases are possible. This structure is analogous to the scenario in behavioural sciences for the "Question Order effect" (Wang and Busemeyer, 2013). In the sheaf-theoretic framework measurement contexts are dictated only by the choices of local measurements and we face two possibilities when modelling these examples. In the first possibility, one can consider the two contexts subject-verb/verb-object as disjoint and as a result lose some semantic information. This is because, for example, *adopt* in *adopt boxer* would be treated as completely unrelated to *adopt* in *boxer adopts*. In this case, all such systems will be trivially non-contextual, as there will be no intersecting local measurements. In our paper, on the other hand, we choose a second possibility and decide to keep the semantic information but as a result any system for which the distribution arising from the verb-object context differs from the one associated with subject-verb context will be signalling. This type of model does not easily lend itself to a sheaf-theoretic analysis but admits a straightforward CbD analysis[3].

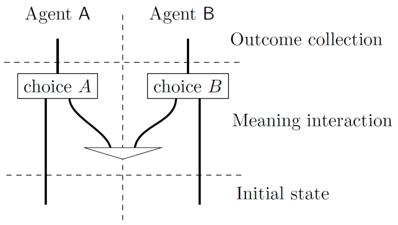


Figure 3: Example of a 2-words scenario. The state (triangle) represents the predefined conditions of the interaction (e.g. $verb - object$).

5 Possibilistic examples

We demonstrate the methodology by choosing three sets of phrases from the sets considered by Wang (2020) as well as two verb-object/subject-

[2] One may imagine a metaphorical meaning of this phrase, e.g. when referring to a *depressing policy*. In this paper, however, we work with non metaphorical meanings in order to keep the hand annotations of interpretations manageable.

[3] We are not aware of any theoretical reason why Bell-scenario-like models could not be CbD-contextual, none, however, have been found using the corpus mining methodology of this paper.

verb examples. For each of these phrases, we tabulate how we encoded the meanings of each word, provide an empirical Bell-style table for the possibilistic cases and outline the different types of contextual features each example demonstrates.

5.1 $\{$***coach, boxer***$\} \times \{$***lap, file***$\}$

Encoding	Meanings of			
	coach	*boxer*	*lap*	*file*
0	*sport*	*boxing*	*run*	*document*
1	*bus*	*dog*	*drink*	*smooth*

(a) Encoding of meanings of *coach*, *boxer*, *lap* and *file*.

$subject$	$verb$	(0,0)	(0,1)	(1,0)	(1,1)
$coach$	lap	1	1	1	0
$coach$	$file$	1	1	0	0
$boxer$	lap	1	1	1	1
$boxer$	$file$	1	1	0	0

(b) Empirical model

Figure 4: Possibilistic model associated with the subject-verb model $\{$*coach*, *boxer*$\} \times \{$*lap*, *file*$\}$.

We start with two subject-verb phrases where both of the subjects and both of the verbs are ambiguous. The verbs are *lap* and *file*, which can be understood as drinking a liquid (e.g. *the dog lapped the water*) or going past someone on a track (e.g. *the runner lapped their competitor*) for *lap*, and storing information (e.g. *filing a complaint*) or smoothing surfaces with a tool (e.g. *filing nails or teeth*) for *file*. The nouns *coach* and *boxer* mean a person who trains athletes (e.g. *a sport coach*) or a type of bus (e.g. *a coach trip*), and a person practising boxing (e.g. *a heavyweight boxer*), or a specific dog breed respectively. This example is modelled possibilistically in Fig. 4b and depicted in the bundle diagram of Fig. 2a. Not all of the local assignments can be extended to a global one, for example, the assignment *coach* ↦ *bus* is possible in the phrase *the coach laps*, but this assignment cannot be extended in the phrase *the coach files*.

This apparent "contextuality", however, is entirely due to the fact that the model is possibilistically signalling and can be seen by the fact that the support of the contexts *the coach lap* and *the coach files*, restricted to the measurement *coach* do not coincide ($[coach \mapsto bus] \in coach\ lap|_{coach}$ but $[coach \mapsto bus] \notin coach\ file|_{coach}$). Hence, we cannot judge the contextuality of this model in the sheaf-theoretic framework.

5.2 $\{$***tap, box***$\} \times \{$***pitcher, cabinet***$\}$

We now consider an empirical model which is possibilistically non-signalling, and in fact contextual. This model deals with two verb-object phrases where the verbs are $\{$*tap*, *box*$\}$, and their objects are $\{$*pitcher*, *cabinet*$\}$. Here, *tap* is taken to mean either gently touching (e.g. *tapping somebody on the shoulder*) or secretly recording (e.g. *tapping phones*); other meanings of the verb *tap* exist (e.g. doing tap dancing, tapping resources, etc.), but since these other meanings are irrelevant in the phrases of interest, we restrict ourselves to these two meanings. In addition, the verb *box* is understood as putting in a container and practising boxing. Again, other meanings of the verb *to box* exist, but as before, we worked with two dominant meanings and ignored the rest. The noun *cabinet* either represents a governmental body (e.g. *the Shadow Cabinet*) or a piece of furniture, and finally the noun *pitcher* either refers to a jug or a baseball player. As we can see in Fig. 2b, the assignment *tap* ↦ *touch* cannot be extended to a global assignment and is therefore possibilistically contextual.

Encoding	Meanings of			
	tap	*box*	*cabinet*	*pitcher*
0	*touch*	*put in boxes*	*government*	*jug*
1	*record*	*fight*	*furniture*	*baseball player*

(a) Encoding of meanings of *tap*, *box*, *cabinet* and *pitcher*.

$verb$	$object$	(0,0)	(0,1)	(1,0)	(1,1)
tap	$pitcher$	1	1	0	1
tap	$cabinet$	0	1	1	0
box	$pitcher$	1	0	0	1
box	$cabinet$	0	1	1	0

(b) Empirical model

Figure 5: Possibilistic model associated with the verb-object model $\{$*tap*, *box*$\} \times \{$*pitcher*, *cabinet*$\}$.

As we move to section 6 and mine probability distributions from corpus for this same model, we see that this possibilistically non-signalling model becomes probabilistically signalling.

5.3 $\{$***press, box***$\} \times \{$***can, leaves***$\}$

In this model, each word has multiple grammatical types and different meanings as follows:

- *to press* (v): Exert pressure upon something
- *press* (n): Media which publishes newspapers and magazines
- *press* (n): Device used to apply pressure. (e.g. *They used to use printing presses before the invention of printers.*)

- *to box* (v): To put in a box
- *to box* (v): To fight, to practice boxing
- *box* (n): Container
- *can* (n) : Tin container
- *to can* (v): To preserve food in a can (e.g. *He cans his own sardines.*)
- *can* (auxiliary): To be able to
- *leaves* (v) : Conjugated form of *to leave*
- *leaves* (n): Plural of *leaf*

As we can see in the bundle diagram associated with the model (Fig. 2c), the marginals of the possibilistic distributions which share a local measurement have the same support, making this model possibilistically non-signalling. In addition, every local section can be extended to a global assignment, which makes the model non-contextual. In section 6.3 and section 8, we endeavour to see whether this model is probabilistically contextual.

Encoding	Meanings of			
	press	*box*	*can*	*leaves*
0	*push*	*put in boxes*	*tin*	*leave*
1	*media*	*fight*	*preserve*	*leaf*
2	*machine*	*container*	*able to*	⋆

(a) Encoding of meanings of *press*, *box*, *can* and *leaves*.

A	B	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)	(2,0)	(2,1)	(2,2)
press	*can*	1	0	0	0	0	1	0	1	1
press	*leaves*	0	1	0	1	0	0	1	0	0
box	*can*	1	0	0	0	0	0	0	1	1
box	*leaves*	0	1	0	0	0	0	1	0	0

(b) Empirical model

Figure 6: Possibilistic model associated with the model $\{press, box\} \times \{can, leaves\}$.

5.4 Subject-verb v. Verb-object

We now introduce two models for which both of subject-verb and verb-object contexts are possible and provide two examples. These are the combinations *adopt boxer/boxer adopts* and *throw pitcher/pitcher throws*, where *boxer* and *pitcher* are defined as in sections 5.1 and 5.2 respectively, and the verbs *adopt* and *throw* can either take literal (e.g. *adopt a child or a pet*, *throwing a projectile*) or figurative (e.g. *adopt a new feature*, or *throwing shadows*) interpretations. The possibilistic models associated with these examples are depicted in Fig. 7 and in the bundle diagrams of Fig. 8. The models are signalling and hence, a sheaf-theoretic analysis would not be possible.

6 Probabilistic variants

We consider the same examples as in the previous section, but from a probabilistic point of view.

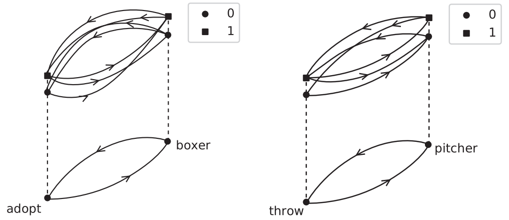
(*adopt*, *boxer*)	(0,0)	(0,1)	(1,0)	(1,1)
adopt → *boxer*	0	1	1	1
adopt ← *boxer*	1	1	1	1

(a) *adopt boxer/boxer adopts*

(*throw*, *pitcher*)	(0,0)	(0,1)	(1,0)	(1,1)
throw → *pitcher*	1	0	1	1
throw → *pitcher*	0	1	1	1

(b) *throw pitcher/pitcher throws*

Figure 7: Empirical models for the pairs of words examples. Here, the different contexts are depicted as follows: *verb*→ *noun* corresponds to the verb-object context while *verb*←*noun* corresponds to the subject-verb phrase. The outcomes labels are the same for both contexts; for example $(0, 1)$ in (a) means *adopt*↦ 0, *boxer*↦ 1 for both contexts.



(a) *adopt boxer/boxer adopts*

(b) *throw pitcher/pitcher throws*

Figure 8: Bundle diagrams of the two noun-verb pairs with contexts verb-object and subject-verb. The encoding of the nouns are the same as in Figs. 4a and 5a; for verbs, outcomes 0 and 1 represent literal and figurative meanings, respectively. The measurement contexts (verb-object or subject-verb) are depicted by arrows on the associated edges.

The probability distributions are obtained from the British National Corpus (BNC, 2007) and UKWaC (Baroni et al., 2009). BNC is an open-source text corpus comprising of 100 million words, spread across documents of different nature (including press articles, fiction, transcription of spoken language, and academic publications). UKWaC is a 2 billion word corpus constructed from the Web limiting the crawl to the .uk domain. Both BNC and UKWaC are part-of-speech tagged, hence, they provide grammatical relations and the lemma forms of words. The semantic interpretation of the words and phrases are absent from these corpora and had to be decided by the authors manually.

6.1 $\{$*coach, boxer*$\} \times \{$*lap, file*$\}$

Recall that the model in section 5.1 was possibilistically signalling. The frequencies mined from corpora were found to have the same support as the model described in section 5.1 (see Fig. 9), whence the probabilistic analogue remains signalling.

subject	*verb*	(0,0)	(0,1)	(1,0)	(1,1)
coach	*lap*	2/11	7/11	2/11	0
coach	*file*	43/44	1/44	0	0
boxer	*lap*	11/53	22/53	8/53	12/53
boxer	*file*	35/54	19/54	0	0

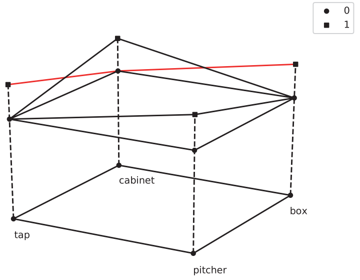
Figure 9: Empirical model associated with the probabilistic model of $\{coach, boxer\} \times \{lap, file\}$

6.2 $\{$***tap*, *box***$\} \times \{$***pitcher*, *cabinet***$\}$

By mining frequencies of co-occurrences of phrases in our two corpora, the model described in section 5.2 becomes probabilistically signalling, see Fig. 10a. We therefore cannot decide whether this model is probabilistically contextual in the sheaf-theoretic framework.

verb	*object*	(0,0)	(0,1)	(1,0)	(1,1)
tap	*pitcher*	17/22	15/22	0	0
tap	*cabinet*	1/21	3/7	11/21	0
box	*pitcher*	3/4	1/4	0	0
box	*cabinet*	3/7	10/21	2/21	0

(a) Empirical model



(b) Bundle diagram

Figure 10: Probabilistic model associated with the probabilistic model of $\{tap, box\} \times \{pitcher, cabinet\}$.

It is important to note that, given the finite size and the nature of the corpora considered, many interpretations of the phrases considered did not occur; for example, there was no instance of baseball players' (pitchers') phones or conversations being recorded (tapped). On the other hand, several other interpretations of the phrases did occur, for example figuratively putting cabinet members in boxes or black-boxing a group of ministers.

6.3 $\{$***press*, *box***$\} \times \{$***can*, *leaves***$\}$

The possibilistic version of this example, presented in section 5.3, was non-signalling. Even if tabulating the observed frequencies did not change the support of the distributions, the model has become probabilistically signalling. Indeed, one can check that:

$$\begin{aligned} P\left[box\ leaves|_{box} \mapsto put\ in\ boxes\right] &= 2/3 \\ \neq P\left[box\ can|_{box} \mapsto put\ in\ boxes\right] &= 7/74 \qquad (1) \end{aligned}$$

Yet again, we cannot use the sheaf-theoretic framework to evaluate the contextuality of this model.

A	B	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)	(2,0)	(2,1)	(2,2)
press	*can*	2/25	0	0	0	0	41/50	0	1/50	2/25
press	*leaves*	0	6/13	0	5/13	0	0	2/13	0	0
box	*can*	7/74	0	0	0	0	0	0	1/74	33/37
box	*leaves*	0	2/3	0	0	0	0	1/3	0	0

Figure 11: Empirical model associated with the probabilistic model of $\{press, box\} \times \{can, leaves\}$.

6.4 Subject-verb v. Verb-object

We now present the probability distribution arising from the examples in section 5.4. As some of the previous models, the two corpora did not have instances of all the possible readings of each of the contexts. The obtained probability distributions are shown in Fig. 12. As expected, the probability distribution is also signalling.

(*adopt*, *boxer*)	(0,0)	(0,1)	(1,0)	(1,1)
adopt $\to$ *boxer*	0	29/30	1/30	0
adopt $\leftarrow$ *boxer*	1/4	0	0	3/4

(a) *adopt boxer/boxer adopts*

(*throw*, *pitcher*)	(0,0)	(0,1)	(1,0)	(1,1)
throw $\to$ *pitcher*	2/5	0	1/10	1/2
throw $\leftarrow$ *pitcher*	0	2/3	1/3	0

(b) *throw pitcher/pitcher throws*

Figure 12: Empirical models for the pairs of words.

7 Non-signalling and ambiguity in natural language

Non-signalling is a necessary condition for demonstrating non-locality in quantum mechanics. In experiments such as the one described in Einstein et al. (1935), this assumption models the space-like separation between the systems, e.g. two entangled qubits which are measured in geographically different labs, or more generally the fact that no communication between these systems is possible after their preparation. Non-signalling is a property that ensures some laws of quantum mechanics hold in specific systems and certainly there is no reason to assume it for natural language. In order to understand why not, let's try and use an analogy with quantum systems. In our experiment, ambiguous phrases become analogous to entangled quantum systems and each word within a phrase to a qubit. In the *subject-verb* phrases we considered, a form of communication between words within a phrase becomes possible if after, say the subject-measuring agent determines the meaning of the subject, the verb-measuring agent has a more limited choice in determining the meaning of the verb. A similar situation is true for the *verb-object* phrases. In these cases, communication between the words of a phrase may seem possible but will definitely not in general. For instance, consider the *coach lap* phrase, if the subject-measuring agent decides that the meaning of *coach* is *bus*, the verb-measuring agent does not get a choice, since *buses* cannot *drink*. In this case, communication between the subject and verb-measuring agents is needed. If the subject-agent, however, sets the meaning of *coach* to be *sports trainer*, the verb-measuring agent still gets a choice for the meaning of *lap*, since a *trainer* can *run in circles* as well as *drink something up*. In this case, communication between the agents is not as clearly possible as before.

8 Contextuality-by-Default

We will now study the contextuality of the probabilistic signalling systems we obtained in section 6 using the Contextuality-by-Default framework. In this framework, each set of jointly distributed measurements of the empirical model is called a context, and the contextuality of a system is defined by the impossibility of creating a global joint distribution in which the variables corresponding to each measurement in each pair of contexts where they appear are equal to each other with maximal probability (instead of always). For example, in expression (1) we noticed that the proportions with which the word "box" is assigned the meaning "put in boxes" differs between the contexts with measurements "box leaves" and "box can". This difference makes the system signalling and implies that the two variables cannot be treated as equal to each other within a global assignment. They need to be treated as different random variables. The maximal probability that those two random variables can both receive the assignment "put in boxes" is $\min\{2/3, 7/74\} = 7/74$. These probabilities can be found for every pair of variables corresponding to each measurement. Continuing with the example of the variables corresponding to the measure of "box" from the example in Section 6.3, the maximal probability with which they could be assigned the meaning "fight" in the contexts "box leaves" and "box can" is equal to $\min\{0, 0\} = 0$, and the probability with which they both can be assigned "container" is $\min\{1/3, 67/74\} = 1/3$.

The task of finding whether a global joint distribution that maximizes these probabilities for every measurement exists can be solved by linear programming. Dzhafarov and Kujala (2016) describe how to define this task for systems that include measurements with a finite number of outcomes by taking all possible dichotomizations of their respective outcome sets. We illustrate the procedure with the proportions of the system in Section 6.3. The description of this system simplifies by noting that the word "leaves" could only be assigned two meanings and that for the word "box"

$$\begin{aligned} P\left[\mathit{box\ leaves}|_{\mathit{box}} \mapsto \mathit{fight}\right] &= \quad 0, \\ P\left[\mathit{box\ can}|_{\mathit{box}} \mapsto \mathit{fight}\right] &= \quad 0, \end{aligned}$$

effectively making those variables also binary. Thus, we need only consider dichotomizations of variables corresponding to the measurements for "press" and "box".

A global joint distribution of all dichotomized variables in our system must define probabilities for 2^{16} different events. They are the combination of the outcomes of 16 binary random variables: a) 6 in context "press can" including the three dichotomizations of $\mathit{press\ can}|_{\mathit{press}}$ and $\mathit{press\ can}|_{\mathit{can}}$; b) 4 in context "press leaves"; c) 4 in context "box can"; and d) 2 in context "box leaves". The 2^{16} probabilities are restricted by the probabilites estimated in Section 6.3, which total 97 linear constrains considering the joint events of the dichotomizations, individual margins (as the ones in expression (1)), and that probabilities in a distribution add to unity. These probabilities are further restricted by the maximal probabilities computed for the pairs of variables corresponding to the same dichotomization of the same measurement. These maximal probabilities are computed by taking the minimum of the two compared probabilities as explained above, and they amount to 8 linear constrains. In all, a total of 105 linear constrains that the probabilities of the 2^{16} events must satisfy, and that can be represented in a 105×2^{16} matrix of coefficients. Solving the set of linear equations for this example showed that it was possible to find such a global joint distribution. Whence, the system is not contextual.

The systems in sections 6.1 and 6.2 can be shown to be non-contextual within the CbD framework from a Bell inequality for certain signalling systems which was proved in Kujala and Dzhafarov (2016).

8.1 Subject-verb v. Verb-object

Let us now return to the pairs of words introduced in 5.4. The probability distributions for the models *adopt boxer/boxer adopts* and *throw pitcher/pitcher throws*, mined as in section 6, are depicted in Figs. 12a and 12b, respectively.

Unlike the systems considered above, these two are contextual within the CbD framework. This can be shown using the Bell-type inequality proved in Kujala and Dzhafarov (2016) and, using the results from Dzhafarov et al. (2020), we can measure the degree of contextuality of each of these two systems. The contextuality measure for the *adopt-boxer* pair is $1/30$ and the measure for the *throw-pitcher* pair is $7/30$. These measures indicate how far from becoming non-contextual is each system.

Clearly, the system for the pair *adopt-boxer* could easily become non-contextual if the corpora search in the verb-object context had failed to find a figurative meaning of adopt, together with the fighter meaning of boxer for any occurrence of the words "adopt boxer". More generally, we can assess how reliably contextual is this system by means of parametric bootstrap. We find that the probability with which we could find a non-contextual system based on the distributions in Fig. 12a is larger than .56.

The contextuality for the pair *throw-pitcher* is much larger, and indeed the system would need to exhibit many occurrences of meaning assignments that contravene the general patterns exhibited within each of the contexts. For example, the system would be deemed non-contextual if the proportion of times where *throw* took the literal meaning together with an interpretation of pitcher as a *jug* in the expression "throw pitcher" increased from $1/10$ to $1/3$ while preserving the overall proportions with which each of the words was interpreted with a given meaning (say, *throw* remains interpreted literally $3/5$ of the times). Analogously to the previous computation, given the probabilities estimated in Fig. 12b , the probability of finding the system non-contextual is larger than .08.

9 Conclusions and Discussion

Undoubtedly, the context of ambiguous words plays an important role in their disambiguation process. The nature of this role, on the other hand, is not properly understood and quantified. In this work, we find ambiguous phrases that are possibilistically (i.e. logically) contextual in the sheaf-theoretic model, but show that their probabilistic extensions become signalling. In the presence of signalling, we analyse these examples in the CbD framework and discover some of them are not CbD-contextual. At the same time, however, we do find examples that are CbD-contextual albeit signalling. We then argue that the use of different contextuality frameworks allows us to formally study the effect of the context on choices of interpretation of ambiguous phrases, paving the way for a systematic study of general contextual influences in natural language.

This study was restricted by the nature of the types of meanings we considered and the size of our corpora. Indeed, the observed frequencies of phrases were not always consistent with our intuition, and in some cases, meaningful phrases did not appear in the corpus altogether. An example was the word *coach*, which could either mean *a sports trainer* or a *type of bus*. In the corpora we considered, the latter meaning was in fact quite rare. Our conjecture is that this is due to the fact that the corpora we considered were both almost exclusively based on British English, whereas, the *bus* meaning of *coach* is mainly American. Regarding types of meaning, in order to facilitate our manual search for occurrences of interpretations, we restricted the domain of possible meanings and did not consider figuratively metaphorical options. An example is the verb *boxing*, which can also mean *labelling* or *ignoring the workings of*, but we only considered its *putting in a box* and *fist fighting* meanings. In future work, we aim to overcome this restrictions by widening our experimental data and gather human judgement on degrees of likelihood of each meaning combination. This will allow also us to consider a wider range of grammatical relations in the contexts and also study the effects of these structures on the disambiguation process as well as allowing a more reliable estimation of the probability distributions.

Acknowledgments

This work was supported by the EPSRC Centre for Doctoral Training in Delivering quantum Technologies, grant ref. EP/S021582/1. The authors would like to thank Ruth Kempson, Shane Mansfield, and Ehtibar Dzhafarov for fruitful discussions.

References

2007. The British National Corpus. Distributed by Bodleian Libraries, University of Oxford, on behalf of the BNC Consortium. `http://www.natcorp.ox.ac.uk/`. Version 3 (BNC XML Edition).

Samson Abramsky, Rui Soares Barbosa, Kohei Kishida, Raymond Lal, and Shane Mansfield. 2015. Contextuality, Cohomology and Paradox. In *24th EACSL Annual Conference on Computer Science Logic (CSL 2015)*, volume 41 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 211–228, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Samson Abramsky and Adam Brandenburger. 2011. The sheaf-theoretic structure of non-locality and contextuality. *New J. Phys.*, 13:113036.

Samson Abramsky and Lucien Hardy. 2012. Logical Bell inequalities. *Physical Review A*, 85(6).

Chris Barker and Chung-chieh Shan. 2015. *Continuations and Natural Language*. Oxford Scholarship.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Ivano Basile and Fabio Tamburini. 2017. Towards quantum language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1840–1849, Copenhagen, Denmark. Association for Computational Linguistics.

John S. Bell. 1964. On the Einstein Podolsky Rosen paradox. *Physics Physique Fizika*, 1:195–200.

William Blacoe, Elham Kashefi, and Mirella Lapata. 2013. A quantum-theoretic approach to distributional semantics. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 847–857, Atlanta, Georgia. Association for Computational Linguistics.

Peter D. Bruza, Kirsty Kitto, D. Nelson, and C. McEvoy. 2009. Is there something quantum-like about the human mental lexicon? *Journal of Mathematical Psychology*, 53:362–377.

Peter D. Bruza, Kirsty Kitto, Brentyn J. Ramm, and Laurianne Sitbon. 2015. A probabilistic framework for analysing the compositionality of conceptual combinations. *Journal of Mathematical Psychology*, 67:26 – 38.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a compositional distributional model of meaning.

Ehtibar N. Dzhafarov and Janne V. Kujala. 2016. Context–content systems of random variables: The Contextuality-by-Default theory. *Journal of Mathematical Psychology*, 74:11 – 33. Foundations of Probability Theory in Psychology and Beyond.

Ehtibar N. Dzhafarov, Janne V. Kujala, and Víctor H. Cervantes. 2020. Contextuality and noncontextuality measures and generalized bell inequalities for cyclic systems. *Phys. Rev. A*, 101:042119.

Ehtibar N. Dzhafarov, Ru Zhang, and Janne Kujala. 2016. Is there contextuality in behavioural and social systems? *Philosophical Transactions of the Royal Society of London Series A*, 374(2058):20150099.

Albert Einstein, Boris Podolsky, and Nathan Rosen. 1935. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780.

Lyn Frazier and Keith Rayner. 1990. Taking on semantic commitments: Processing multiple meanings vs. multiple senses. *Journal of Memory and Language*, 29(2):181–200.

B. Hensen, H. Bernien, A. E. Dréau, A. Reiserer, N. Kalb, M. S. Blok, J. Ruitenberg, R. F. L. Vermeulen, R. N. Schouten, C. Abellán, W. Amaya, V. Pruneri, M. W. Mitchell, M. Markham, D. J. Twitchen, D. Elkouss, S. Wehner, T. H. Taminiau, and R. Hanson. 2015. Loophole-free bell inequality violation using electron spins separated by 1.3 kilometres. *Nature*, 526(7575):682–686.

Simon B. Kochen and Ernst P. Specker. 1967. The Problem of Hidden Variables in Quantum Mechanics. *Journal of Mathematics and Mechanics*, 17(1):59–87.

Janne Kujala and Ehtibar Dzhafarov. 2016. Proof of a Conjecture on Contextuality in Cyclic Systems with Binary Variables. *Foundations of Physics*, 46.

Guangxi Li, Zhixin Song, and Xin Wang. 2020a. Variational shadow quantum learning for classificationn. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*.

Qiuchi Li, Dimitris Gkoumas, Alessandro Sordoni, Jian-Yun Nie, and Massimo Melucci. 2020b. Quantum-inspired neural network for conversational emotion recognition. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*.

Ding Liu, Xiaofang Yang, and Minghu Jiang. 2013. A novel classifier based on quantum computation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–488, Sofia, Bulgaria. Association for Computational Linguistics.

Konstantinos Meichanetzidis, Stefano Gogioso, Giovanni De Felice, Nicolò Chiappori, Alexis Toumi, and Bob Coecke. 2020. Quantum Natural Language Processing on Near-Term Quantum Computers.

Martin Pickering and Steven Frisson. 2001. Processing Ambiguous Verbs: Evidence from Eye Movements. *Journal of experimental psychology. Learning, memory, and cognition*, 27:556–73.

Robin Piedeleu, Dimitri Kartsaklis, Bob Coecke, and Mehrnoosh Sadrzadeh. 2015. Open system categorical quantum semantics in natural language processing.

Keith Rayner and Susan A. Duffy. 1986. Lexical complexity and fixation times in reading: Effects of word frequency, verb complexity, and lexical ambiguity. *Memory & Cognition*, 14(3):191–201.

Hinrich Schütze. 1998. Automatic Word Sense Discrimination. *Computational Linguistics*, 24(1):97–123.

Michael K. Tanenhaus, James M. Leiman, and Mark S. Seidenberg. 1979. Evidence for multiple stages in the processing of ambiguous words in syntactic contexts. *Journal of Verbal Learning and Verbal Behavior*, 18(4):427 – 440.

Daphne Wang. 2020. Sheaf theoretic models of contextuality from quantum measurements to natural language. Master's thesis, MRes in Quantum Technologies, Centre for Doctoral Training in Delivering Quantum Technologies University College London.

Zheng Wang and Jerome R. Busemeyer. 2013. A quantum question order model supported by empirical tests of an a priori and precise prediction. *Topics in Cognitive Science*, 5(4):689–710.

Conversational Negation using Worldly Context in Compositional Distributional Semantics

Benjamin Rodatz
Computer Science,
University of Oxford
benjamin.rodatz
@cs.ox.ac.uk

Razin A. Shaikh
Mathematical Institute,
University of Oxford
razin.shaikh
@maths.ox.ac.uk

Lia Yeh
Quantum Group,
Computer Science,
University of Oxford
lia.yeh@cs.ox.ac.uk

All authors have contributed equally.

Abstract

We propose a framework to model an operational conversational negation by applying *worldly context* (prior knowledge) to logical negation in compositional distributional semantics. Given a word, our framework can create its negation that is similar to how humans perceive negation. The framework corrects logical negation to weight meanings closer in the entailment hierarchy more than meanings further apart. The proposed framework is flexible to accommodate different choices of logical negations, compositions, and worldly context generation. In particular, we propose and motivate a new logical negation using matrix inverse.

We validate the sensibility of our conversational negation framework by performing experiments, leveraging density matrices to encode graded entailment information. We conclude that the combination of subtraction negation ($\neg_{sub}$) and phaser in the basis of the negated word yields the highest Pearson correlation of 0.635 with human ratings.

1 Introduction

Negation is fundamental to every human language, marking a key difference from how other animals communicate (Horn, 1972). It enables us to express denial, contradiction, and other uniquely human aspects of language. As humans, we know that negation has an operational interpretation: if we know the meaning of *A*, we can infer the meaning of *not A*, without needing to see or hear *not A* explicitly in any context.

Formalizing an operational description of how humans interpret negation in natural language is a challenge of significance to the fields of linguistics, epistemology, and psychology. Kruszewski et al. (2016) notes that there is no straightforward negation operation that, when applied to the distributional semantics vector of a word, derives a negation of that word that captures our intuition. This work proposes and experimentally validates an operational framework for conversational negation in compositional distributional semantics.

In the field of distributional semantics, there have been developments in capturing the purely logical form of negation. Widdows and Peters (2003) introduce the idea of computing negation by mapping a vector to its orthogonal subspace; Lewis (2020) analogously model their logical negation for density matrices. However, logical negation alone is insufficient in expressing the nuances of negation in human language. Consider the sentences:

a) `This is not an apple;`
`this is an orange.`

b) `This is not an apple;`
`this is a paper.`

Sentence a) is more plausible in real life than sentence b). However, since apples and oranges share a lot in common, their vector or density matrix encodings would most likely not be orthogonal. Consequently, such a logical negation of apple would more likely indicate a paper than an orange.

Blunsom et al. (2013) propose that the encoding of a word should have a distinct “domain” and “value”, and its negation should only affect the “value”. In this way, *not blue* would still be in the domain of *color*. However, they do not provide any scalable way to generate such representation of “domain” and “value” from a corpus. We argue that this domain need not be encoded in the vector or density matrix itself. Instead, we propose a method to generate what we call *worldly context* directly from the word and its relationships to other words, computed a priori using worldly knowledge.

Furthermore, we want such conversational negation to generalize from words to sentences and to entire texts. DisCoCat (Coecke et al., 2010) provides a method to compose the meaning of words

Proceedings of the 2021 Workshop on Semantic Spaces at the Intersection of NLP, Physics, and Cognitive Science, pages 53–65
June 16th, 2021. ©2021 Association for Computational Linguistics

to get the meaning of sentences and DisCoCirc (Coecke, 2020) extends this to propagate knowledge throughout the text. Therefore, we propose our conversational negation in the DisCoCirc formalism, putting our framework in a rich expanse of grammatical types and sentence structures. Focusing on the conversational negation of single words, we leave the interaction of conversational negation with grammatical structures for future work.

Section 2 introduces the necessary background. Section 3 discusses the logical negation using subtraction from the identity matrix from Lewis (2020), and proposes and justifies a second, new form of logical negation using matrix inverse. Section 4 introduces methods for context creation based on worldly knowledge. Section 5 presents the general framework for performing conversational negation of a word by combining logical negation with worldly context. Section 6 experimentally verifies the proposed framework, comparing each combination of different logical negations, compositions, bases, and worldly context generation. We end our discussion with an overview of future work.

2 Background

2.1 Conversational negation

Kruszewski et al. (2016) point out a long tradition in formal semantics, pragmatics and psycholinguistics which has argued that negation—in human conversation—is not simply a denial of information; it also indicates the truth of an *alternative* assertion. They call this alternative-licensing view of negation *conversational negation*.

Another view on negation states that the effect of negation is merely one of information denial (Evans et al., 1996). However, Prado and Noveck (2006) explain that even under this view, the search for alternatives could happen as a secondary effort for interpreting negation in the sentence.

The likelihood of different alternatives to a negated word inherently admits a grading (Oaksford, 2002; Kruszewski et al., 2016). For example, something that is not a *car* is more likely to be a *bus* than a *pen*. They argue that the most plausible alternatives are the ones that are applicable across many varied contexts; *car* can be replaced by *bus* in many contexts, but it requires an unusual context to sensibly replace *car* with *pen*.

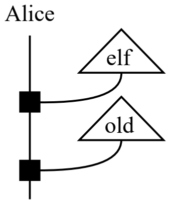


Figure 1: Graphical representation of meaning updating in DisCoCirc - read from top to bottom

2.2 Compositional semantics and DisCoCirc

Language comprehension depends on understanding the meaning of words as well as understanding how the words interact with each other in a sentence. While the former is an understanding of the definitions of words, the latter requires an understanding of grammar. Coecke et al. (2010) build on this intuition to propose DisCoCat, a compositional distributional model of meaning, making use of the diagrammatic calculus originally introduced for quantum computing (Abramsky and Coecke, 2004). In Coecke (2020), this model was extended to DisCoCirc which generalized DisCoCat from modeling individual sentences to entire texts. In DisCoCirc, the two sentences

```
Alice is an elf.
 Alice is old.
```

are viewed as two processes updating the state of Alice, about whom, at the beginning of the text, the reader knows nothing. Graphically this can be displayed as shown in Figure 1. The wire labeled by *Alice* represents the knowledge we have about Alice at any point in time. It is first updated by the fact that she is an elf and subsequently updated by the fact that she is old. We use a black square to represent a general meaning-update operation, which can be one of a variety of operators we discuss in the next section. DisCoCirc allows for more grammatically complex sentence and text structures not investigated in this work.

DisCoCirc allows for various ways of representing meaning such as vector spaces (Coecke et al., 2010; Grefenstette and Sadrzadeh, 2011), conceptual spaces (Bolt et al., 2017), and density matrices (Balkir et al., 2016; Lewis, 2019). A density matrix is a complex matrix, which is equal to its own conjugate transpose (Hermitian) and has non-negative eigenvalues (positive semidefinite). They can be viewed as an extension of vector spaces to allow for encoding lexical entailment structure (see Section 2.4), a property for which they were selected as the model of meaning for this paper.

2.3 Compositions for meaning update

We present four compositions for meaning update:

$$\mathit{spider}(\mathsf{A}, \mathsf{B}) := U_s(\mathsf{A} \otimes \mathsf{B})U_s^\dagger \tag{1}$$

- $U_s = \sum_i |i\rangle \langle ii|$ where $\{|i\rangle\}_i$ is B's eigenbasis
- non-linear AND in Coecke (2020)

$$\mathit{fuzz}(\mathsf{A}, \mathsf{B}) := \sum_i x_i P_i \circ \mathsf{A} \circ P_i \tag{2}$$

- $\mathsf{B} = \sum_i x_i P_i$
- in Coecke and Meichanetzidis (2020)
- Kmult in Lewis (2020)

$$\mathit{phaser}(\mathsf{A}, \mathsf{B}) := \mathsf{B}^{\frac{1}{2}} \mathsf{A} \mathsf{B}^{\frac{1}{2}} \tag{3}$$

- $\mathsf{B} = \sum_i x_i^2 P_i$ where $\mathsf{B}^{\frac{1}{2}} = \sum_i x_i P_i$
- in Coecke and Meichanetzidis (2020)
- Bmult in Lewis (2020)
- corresponds to quantum Bayesian update (van de Wetering, 2018)

$$\mathit{diag}(\mathsf{A}, \mathsf{B}) := \mathit{dg}(\mathsf{A}) \circ \mathit{dg}(\mathsf{B}) \tag{4}$$

- a Compr from De las Cuevas et al. (2020): lifts verbs and adjectives to completely positive maps matching their grammatical type

where A and B are density matrices, x_i is a real scalar between 0 and 1, P_i's are projectors, and the function *dg* sets all off-diagonal matrix elements to 0 giving a diagonal matrix.

Of the many Compr variants (De las Cuevas et al., 2020), we only consider *diag* and *mult* (elementwise matrix multiplication, which is an instance of *spider*) as candidates for composition. All other variants are scalar multiples of one input, the identity wire, or a maximally mixed state; therefore we do not consider them as they discard too much information about the inputs.

For *spider*, *fuzz*, and *phaser*, choosing the basis of the composition determines the basis the resulting density matrix takes on, and its meaning is interpreted in (Coecke and Meichanetzidis, 2020).

2.4 Lexical entailment via hyponymies

A word w_A is a hyponym of w_B if w_A is a type of w_B; then, w_B is a hypernym of w_A. For example, *dog* is a hyponym of *animal*, and *animal* is a hypernym of *dog*. Where there is a meaning relation between two words, there exists an entailment relation between two sentences containing those words. Measures to quantify these relations ought to be *graded*, as one would expect some entailment relations to be weaker than others. Furthermore, such measures should be *asymmetric* (a bee is an insect, but an insect is not necessarily a bee) and *pseudo-transitive* (a t-shirt is a shirt, a shirt can be formal, but a t-shirt is usually not formal).

One of the limitations of the vector space model of NLP is that it does not admit a natural non-trivial graded entailment structure (Balkir et al., 2016; Coecke, 2020). Bankova et al. (2019) utilize the richer setting of density matrices to define a measure called k-hyponymy, generalizing the Löwner order to have a grading for positive operators, satisfying the above three properties. They further lift from entailment between words to between two sentences of the same grammatical structure, using compositional semantics, and prove a lower bound on this entailment between sentences.

The k-hyponymy (k_{hyp}) between density matrices A and B is the maximum k such that

$$\mathsf{A} \sqsubseteq_k \mathsf{B} \iff \mathsf{B} - k\,\mathsf{A} \text{ is a positive operator} \tag{5}$$

where k is between 0 (no entailment) and 1 (full entailment).

Van de Wetering (2018) finds that the crisp Löwner ordering ($k_{\mathsf{hyp}} = 1$) is trivial when operators are normalized to trace 1. On the other hand, they enumerate highly desirable properties of the Löwner order when normalized to highest eigenvalue 1. In particular, the maximally mixed state is the bottom element; all pure states are maximal; and the ordering is preserved under any linear trace-preserving isometry (including unitaries), convex mixture, and the tensor product. In our experiments, we leverage these ordering properties following Lewis (2020)'s convention of normalizing operators to highest eigenvalue ≤ 1.

According to Bankova et al. (2019, Theorem 2), when $\mathit{supp}(\mathsf{A}) \subseteq \mathit{supp}(\mathsf{B})$, k_{hyp} is given by $1/\gamma$, where γ is the maximum eigenvalue of $\mathsf{B}^+\mathsf{A}$. Here B^+ denotes the Moore-Penrose inverse of B, which we refer to in the next section as support inverse. If $\mathit{supp}(\mathsf{A}) \not\subseteq \mathit{supp}(\mathsf{B})$, k_{hyp} is 0. This means that

k_{hyp} admits a grading, but is not robust to errors. In our experiments, to circumvent this issue of almost all of our calculated k_{hyp} being 0, we employ a generalized form of k_{hyp} equivalent to as originally defined in Bankova et al. (2019, Theorem 2), less checking whether $supp(\mathsf{A}) \subseteq supp(\mathsf{B})$.

To propose more robust measures, Lewis (2019) says A entails B with the error term E if there exists a D such that:

$$\mathsf{A} + \mathsf{D} = \mathsf{B} + \mathsf{E} \tag{6}$$

to define the following two entailment measures

$$k_{\mathsf{BA}} = \frac{\sum_i \lambda_i}{\sum_i |\lambda_i|} = \frac{\mathsf{Trace}(\mathsf{D} - \mathsf{E})}{\mathsf{Trace}(\mathsf{D} + \mathsf{E})} \tag{7}$$

$$k_{\mathsf{E}} = 1 - \frac{||\mathsf{E}||}{||\mathsf{A}||} \tag{8}$$

where the λ_i's are the eigenvalues of $\mathsf{B} - \mathsf{A}$. In Equations 7 and 8, the error term E satisfying Equation 6 is constructed by taking the diagonalization of $\mathsf{B} - \mathsf{A}$, setting all positive eigenvalues to zero, and changing the sign of all negative eigenvalues. k_{BA} ranges from -1 to 1, and k_{E} ranges from 0 to 1.

According to De las Cuevas et al. (2020), *diag*, *mult*, and *spider* preserve crisp Löwner order:

$$\mathsf{A}_1 \sqsubseteq \mathsf{B}_1, \mathsf{A}_2 \sqsubseteq \mathsf{B}_2 \iff \mathsf{A}_1 \blacksquare\ \mathsf{A}_2 \sqsubseteq \mathsf{B}_1 \blacksquare\ \mathsf{B}_2 \tag{9}$$

Fuzz and *phaser* do not satisfy Equation 9.

3 Logical negations

To construct conversational negation, we must first define a key ingredient – logical negation, denoted by $\neg$. The logical negation of a density matrix is a unary function that yields another density matrix.

The most important property of a logical negation is that it must interact well with hyponymy. Ideally, the interpretation of the contrapositive of an entailment must be sensible:

$$\mathsf{A} \sqsubseteq \mathsf{B} \iff \neg\mathsf{B} \sqsubseteq \neg\mathsf{A} \tag{10}$$

A weakened notion arises from allowing varying degrees of entailment:

$$\mathsf{A} \sqsubseteq_k \mathsf{B} \iff \neg\mathsf{B} \sqsubseteq_{k'} \neg\mathsf{A} \tag{11}$$

where $k = k'$ in the ideal case.

Equation 11 necessitates any candidate of logical negation to be *order-reversing*. However, van de Wetering (2018) proved that all unitary operations preserve Löwner order. Therefore, no quantum gates can reverse Löwner order, and the search for a logical negation compatible with quantum natural language processing (Coecke et al., 2020) (originally formulated in the category of **CPM(FHilb)** (Piedeleu et al., 2015)) remains an open question.

We now discuss two candidates for logical negation that have desirable properties and interaction with the hyponymies presented in Section 2.4.

3.1 Subtraction from identity negation

Lewis (2020) introduces a candidate logical negation which preserves positivity of density matrix X:

$$\neg_{sub}\mathsf{X} := \mathbb{I} - \mathsf{X} \tag{12}$$

In the case where X is a pure state, it maps X to the subspace orthogonal to it, as the identity matrix $\mathbb{I}$ is the sum of orthonormal projectors. This logical negation satisfies Equation 10 for the crisp Löwner order. It satisfies Equation 11 with $k = k'$ for k_{BA}, but not for k_{hyp} or k_{E}.

3.2 Matrix inverse negation

We introduce a new candidate for logical negation, the *matrix inverse*. This reverses Löwner order, i.e. satisfies Equation 11 with $k = k'$ (see Corollary 1 in Appendix). It additionally satisfies Equation 11 with $k = k'$ for k_{BA} if both density operators have same eigenbases (see Theorem 2 in Appendix).

As the matrix inverse of a non-invertible matrix is undefined, we define a logical negation from two generalizations of the matrix inverse acting upon the support and kernel subspaces, respectively.

Definition 1. *For any density matrix X with spectral decomposition $X = \sum_i \lambda_i |i\rangle\langle i|$,*

$$\neg_{supp}X := \sum_i \begin{cases} \frac{1}{\lambda_i} |i\rangle\langle i|, & \textit{if } \lambda_i > 0 \\ 0, & \textit{otherwise} \end{cases} \tag{13}$$

Definition 1 is the Moore-Penrose generalized matrix inverse and is equal to the matrix inverse when the kernel is empty. It has the property that Equation 11 with $k = k'$ is satisfied for k_{hyp} when $rank(\mathsf{A}) = rank(\mathsf{B})$ (see Theorem 1 in Appendix). We call it the *support inverse*, to contrast with what we call the *kernel inverse*:

Definition 2. *For any non-invertible density matrix X with spectral decomposition $X = \sum_i \lambda_i |i\rangle\langle i|$,*

$$\neg_{ker}X := \sum_i \begin{cases} 1\, |i\rangle\langle i|, & \textit{if } \lambda_i = 0 \\ 0, & \textit{otherwise} \end{cases} \tag{14}$$

The kernel inverse is the limit of matrix regularization by spectral filtering (i.e. setting all zero eigenvalues to an infinitesimal positive eigenvalue), then inverting the matrix and normalizing to highest eigenvalue 1. Its application discards all information about the eigenspectrum of the original matrix. Therefore, applying the kernel inverse twice results in a maximally mixed state over the support of the original matrix. Operationally speaking, $\neg_{ker}$ and $\neg_{sub}$ act upon the kernel of the original matrix identically.

We can think conceptually of a negated word as containing elements both "near" (in support) and "far" (in kernel) from the original word. Therefore, a logical negation should encompass nonzero values in the original matrix's support and in its kernel; it is up to conversational negation to then weight the values in the logical negation according to their contextual relevance.

On their own, neither the support inverse nor the kernel inverse are sensible candidates for logical negation. A convex mixture of the two, which we call *matrix inverse* and denote with $\neg_{inv}$, spans both support and kernel of the original matrix. In our experiments we weight support and kernel equally, but other weightings could be considered, for instance to take into account a noise floor or enforce the naively unsatisfied property that twice application is the identity operation.

When composing a density matrix X with $\neg_{inv}\mathsf{X}$ or $\neg_{supp}\mathsf{X}$ via *spider*, *fuzz*, or *phaser*, the resulting density matrix has the desired property of being a maximally mixed state on the support with zeroes on the kernel (see Theorem 3 and Corollary 2 in Appendix). In other words, this operation is the fastest "quantum (Bayesian, in the case of *phaser*) update" from a density matrix to the state encoding no information other than partitioning support and kernel subspaces. Interpreting composition as logical AND, this corresponds to the contradiction that a proposition (restricted to the support subspace) cannot simultaneously be true and not true.

3.3 Normalization

$\neg_{sub}$, $\neg_{supp}$, and $\neg_{inv}$ preserve eigenvectors (up to uniqueness for eigenvalues with multiplicity > 1). We ignore normalization for logical negation because in our conversational negation framework, which we introduce in Section 5, we can always normalize to largest eigenvalue ≤ 1 after the composition operation.

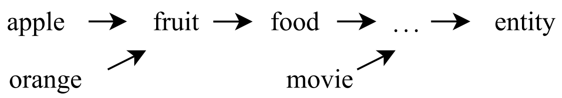


Figure 2: Example of hyponymy structure as can be found in entailment hierarchies

4 Context determination

Negation is intrinsically dependent on context. Context can be derived from two sources: 1) knowledge gained throughout the sentence or the text (textual context), and 2) worldly knowledge from experts or data such as a corpus (worldly context). While textual context depends on the specific text being analyzed, worldly context can be computed a priori. In this section, we introduce worldly context and propose two methods of computing it.

4.1 Worldly context

Worldly knowledge is a certain understanding of the world that most users of a language intuitively possess. We want to capture this worldly knowledge to provide a context for negation that is not explicit in the text. In this section, we propose two methods of generating a worldly context: 1) knowledge encoded in an entailment hierarchy such as WordNet, and 2) generalizing the ideas of the first method to context derivation from the entailment information encoded in density matrices.

4.1.1 Context from an entailment hierarchy

We consider an entailment hierarchy for words that leads to relations such as in Figure 2, where a directed edge can be understood as a hyponym relation. Such relational hierarchy can be obtained from human curated database like WordNet (Fellbaum, 1998) or using unsupervised methods such as Hearst patterns (Hearst, 1992; Roller et al., 2018).

We can use such a hierarchy of hyponyms to generate worldly context, as words usually appear in the implicit context of their hypernyms; for example, *apple* is usually thought of as a *fruit*. Now, to calculate the worldly context for the word *apple*, we take a weighted sum of the hypernyms of *apple*, with more direct hypernyms such as *fruit* weighted higher than more distant hypernyms such as *entity*. This corresponds to the idea that when we talk in the context of *apple*, we are more likely to talk about an *orange* (hyponym of *fruit*) than a *movie* (hyponym of *entity*). Hence, for a word w

with hypernyms $h_1, \ldots, h_n$ ordered from closest to furthest, we define the worldly context $\mathtt{wc}_w$ as:

$$\llbracket \mathtt{wc}_w \rrbracket := \sum_i p_i \llbracket h_i \rrbracket \tag{15}$$

where $p_i \geq p_{i+1}$ for all i.

For this approach, we assume that the density matrix of the word is a mixture containing its hyponyms; i.e. the density matrix of *fruit* is a mixture of all fruits such as *apple*, *orange* and *pears*.

4.1.2 Context using entailment encoded in the density matrices

As explained in Section 2.4, density matrix representation of words can be used to encode the information about entailment between words. Furthermore, this entailment can be graded; for example, *fruit* would entail *dessert* with a high degree, but not necessarily by 1. Such graded entailment is not captured in the human curated WordNet database. Although there have been proposals to extend WordNet (Boyd-Graber et al., 2006; Ahsaee et al., 2014), such semantic networks are not yet available.

We generalize the idea of entailment hierarchy by considering a directed weighted graph where each node is a word and the edges indicate how much one word entails the other. Once we have the density matrices for words generated from corpus data, we can build this graph by calculating the graded hyponymies (see Section 2.4) among the words, thereby extracting the knowledge gained from the corpus encoded in the density matrices, without requiring human narration.

Consider words x and y where $x \sqsubseteq_p y$ and $y \sqsubseteq_q x$. In the ideal case, there are three possibilities: 1) x and y are not related (both p and q are small), 2) one is a type of the other (one of p and q is large), or 3) they are very similar (both p and q are large). Hence, we need to consider both p and q when we generate the worldly context. To obtain the worldly context for a word w, we consider all nodes (words) connected to w along with their weightings. If $p_1, \ldots, p_n$ and $q_1, \ldots, q_n$ are the weights of the edges from w to words $h_1, \ldots, h_n$, then worldly context $\mathtt{wc}_w$ is given by

$$\llbracket \mathtt{wc}_w \rrbracket := \sum_i f(p_i, q_i) \llbracket h_i \rrbracket \tag{16}$$

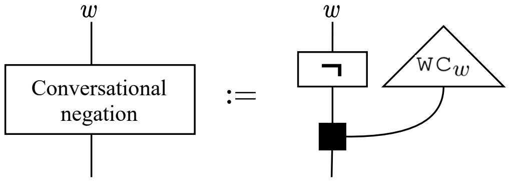
where f is some function of weights p_i and q_i.

5 Conversational negation in DisCoCirc

5.1 A framework for conversational negation

In this section, we present a framework to obtain conversational negation by composing logical negation with worldly context. As discussed in Section 2.1, negation—when used in conversation—can be viewed as not just a complement of the original word, but as also suggesting an *alternative* claim. Therefore, to obtain conversational negation, we need to adapt the logical negation to take into account the worldly context of the negated word.

In DisCoCirc (see Section 2.2), words are wires, and sentences are processes that update meaning of the words. Similarly, we view *conversational negation* as a process that updates the meaning of the words. We propose the general framework for conversational negation by defining it to be the logical negation of the word, updated through composition with the worldly context evoked by that word:



(17)

The framework presented here is general; i.e. it does not restrict the choice of logical negation, worldly context or composition. The main steps of conversational negation are:

1. Calculate the logical negation $\neg(\llbracket w \rrbracket)$.
2. Compute the worldly context $\llbracket \mathtt{wc}_w \rrbracket$.
3. Update the meaning of $\neg(\llbracket w \rrbracket)$ by composing with $\llbracket \mathtt{wc}_w \rrbracket$ to obtain $\neg(\llbracket w \rrbracket) \,\blacksquare\, \llbracket \mathtt{wc}_w \rrbracket$.

Further meaning updates can be applied to the output of conversational negation using compositional semantics as required from the structure of the text, although we do not investigate this in the current work.

5.2 See it in action

We present a toy example to develop intuition of how meaning provided by worldly context interacts with logical negation and composition to derive conversational negation. Suppose $\{apple,\ orange,\ fig,\ movie\}$ are pure states forming an orthonormal basis (ONB). In practice ONBs are far larger, but this example suffices to illustrate how the conversational negation accounts for which states are relevant. We take $\neg_{sub}$ as the choice of negation and *spider* in this ONB as the choice of composition.

Now, consider the sentence:

```
This is not an apple.
```

Although in reality the worldly context of $apple$ encompasses more than just $fruit$, for ease of understanding, assume the worldly context of apple is $\llbracket \mathtt{wc}_{apple} \rrbracket = \llbracket fruit \rrbracket$, given by

$$\llbracket fruit \rrbracket = \frac{1}{2}\llbracket apple \rrbracket + \frac{1}{3}\llbracket orange \rrbracket + \frac{1}{6}\llbracket fig \rrbracket$$

Applying $\neg_{sub}(\llbracket apple \rrbracket) = \mathbb{I} - \llbracket apple \rrbracket$, we get

$$\neg_{sub}(\llbracket apple \rrbracket) = \llbracket orange \rrbracket + \llbracket fig \rrbracket + \llbracket movie \rrbracket$$

Finally, to obtain conversational negation, logical negation is endowed with meaning through the application of worldly context.

$$\neg_{sub}(\llbracket apple \rrbracket) \curlyvee \llbracket fruit \rrbracket = \frac{1}{3}\llbracket orange \rrbracket + \frac{1}{6}\llbracket fig \rrbracket$$

This conversational negation example not only yields all *fruits* which are not *apples*, but also preserves the proportions of the non-apple fruits.

6 Experiments

To validate the proposed framework, we perform experiments on the data set of alternative plausibility ratings created by Kruszewski et al. (2016)[1]. In their paper, Kruszewski et al. (2016) predict plausibility scores for word pairs consisting of a negated word and its alternative using various methods to compare the similarity of the words. While achieving a high correlation with human intuition, they do not provide an operation to model the outcome of a conversational negation. Through the experiments, we test whether our operational conversational negation still has correlation with human intuition.

6.1 Data

The Kruszewski et al. (2016) data set consists of word pairs containing a noun to be negated and an alternative noun, along with a plausibility rating. We will denote the word pairs as (w_N, w_A). The authors transform these word pairs into simple sentences of the form: *This is not a* w_N*, it is a* w_A (e.g. This is not a radio, it is a dad.). These sentences are then rated by human participants on how plausible they are to appear in a natural conversation.

To build these word pairs, Kruszewski et al. (2016) randomly picked 50 common nouns as w_N and paired them with alternatives that have various relations to w_N. Then using a crowd-sourcing service, they asked the human participants to judge the plausibility of each sentence. The participants were told to rate each sentence on a scale of 1 to 5.

6.2 Methodology

We build density matrices from 50 dimensional GloVe (Pennington et al., 2014) vectors using the method described in Lewis (2019). Then for each word pair (w_N, w_A) in the data set, we use various combinations of operations to perform conversational negation on the density matrix of w_N and calculate similarity with the density matrix of w_A.

For conversational negation, we experiment with different combinations of logical negations, composition operations and worldly context. We use two types of logical negations: $\neg_{sub}$ and $\neg_{inv}$. For composition, we use *spider*, *fuzz*, *phaser*, *mult* and *diag*. With *spider*, *fuzz* and *phaser*, we perform experiments in two choices of basis: ‘w’, the basis of $\neg(\llbracket w_N \rrbracket)$, and ‘c’, the basis of $\llbracket \mathtt{wc}_{w_N} \rrbracket$. We use worldly context generated from the WordNet entailment hierarchy as per Section 4.1.1; we experiment with different methods to calculate the weights p_i along the hypernym path.

To find plausibility ratings, we calculate hyponymies k_{hyp}, k_{E} and k_{BA}, as well as *trace similarity* (the density operator analog of cosine similarity for vectors), between the density matrix of the conversational negation of w_N and $\llbracket w_A \rrbracket$. Note that in our experiments, unlike in the originally proposed formulation of k_{hyp}, we generalize k_{hyp} to not be 0 when $supp(\mathsf{A}) \not\subseteq supp(\mathsf{B})$, as described in Section 2.4. We calculate entailment in both directions for k_{E} and k_{hyp}, which are asymmetric. The entailment from w_N to w_A is denoted k_{E1} and k_{hyp1} while the entailment from w_A to w_N is denoted k_{E2} and k_{hyp2}. Finally, we calculate the Pearson correlation between our plausibility ratings and the mean human plausibility ratings from Kruszewski et al. (2016).

6.3 Results

Our experiments revealed that the best conversational negation is obtained by choosing $\neg_{sub}$ with *phaser* in the basis ‘w’. We achieve 0.635 correlation of the *trace similarity* plausibility rating with

[1] The data set is available at `http://marcobaroni.org/PublicData/alternatives_dataset.zip`

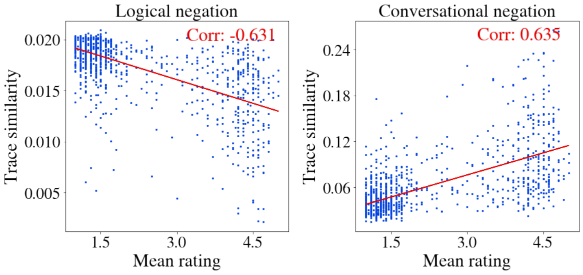


Figure 3: Correlation of logical (left) and conversational negation (right) with mean human rating

Logical negation	Composition	k_{E1}	k_{E2}	k_{hyp1}	k_{hyp2}	k_{BA}	trace
$\neg_{sub}$	spider$_c$	-0.152	0.068	0.287	0.357	0.241	0.355
	spider$_w$	-0.181	-0.176	0.282	0.217	0.243	-0.154
	phaser$_c$	-0.270	-0.279	0.305	0.153	0.256	-0.235
	phaser$_w$	0.432	0.602	0.289	0.495	0.311	0.635
	fuzz$_c$	-0.226	-0.064	0.298	0.175	0.246	0.449
	fuzz$_w$	-0.252	-0.125	0.304	0.191	0.259	0.047
$\neg_{inv}$	spider$_c$	0.197	0.455	0.263	0.419	0.261	0.455
	spider$_w$	0.006	-0.034	0.273	0.112	0.163	0.111
	phaser$_c$	-0.258	-0.129	0.285	0.139	0.183	-0.037
	phaser$_w$	0.279	0.432	0.285	0.285	0.241	0.519
	fuzz$_c$	-0.212	-0.050	0.296	0.034	0.188	0.135
	fuzz$_w$	-0.261	-0.070	0.299	0.180	0.232	0.033

Figure 4: Correlation of various conversational negations with mean plausibility ratings of human participants. Correlations above 0.4 are highlighted in green.

the human ratings, as shown in Figure 3 (right).

On the other hand, Figure 3 (left) shows *trace similarity* of $\neg_{sub}$ without applying any context. We observe that simply performing logical negation yields a negative correlation with human plausibility ratings. This is because logical negation gives us a density matrix furthest from the original word, going against the observation of Kruszewski et al. (2016) that an alternative to a negated word appears in similar contexts to it. Figure 3 (right) shows the results of combining this logical negation with worldly context to obtain meaning that positively correlates with how humans think of negation in conversation.

We tested many combinations for conversational negation enumerated in Section 6.2. The correlation between plausibility ratings for our conversational negation and the mean human plausibility rating is shown in Figure 4. We left out *mult* and *diag* from the table as they did not achieve any correlation above 0.3. Now, we will explore each variable of our experiments individually in the next sections.

6.3.1 Logical negation

We tested $\neg_{sub}$ and $\neg_{inv}$ logical negations. We found that the conversational negations built from $\neg_{sub}$ negation usually had a higher correlation with human plausibility ratings, with the highest being 0.635 as shown in Figures 3 and 4. One exception to this is when the $\neg_{inv}$ is combined with *spider* in the basis 'c', for which we get the correlation of 0.455 for both *trace similarity* and k_{E2}.

6.3.2 Composition

We investigated five kinds of composition operations: *spider*, *fuzz*, *phaser*, *mult*, and *diag*. We found that the results using *mult* and *diag* do not have any statistically significant correlation (<0.3) with human plausibility rating. On the other hand, *phaser* (in the basis 'w') has the highest correlation. It performs well with both logical negations. Plausibility ratings for *phaser* with $\neg_{sub}$ negation measured using k_{E2} and *trace similarity* has correlations of 0.602 and 0.635 respectively. *Spider* and *fuzz* have statistically relevant correlation for a few cases but never more than 0.5.

6.3.3 Basis

Spider, *fuzz*, and *phaser* necessitate a choice of basis for applying the worldly context in the conversational negation. We can interpret this choice as determining which input density matrix sets the eigenbasis of the output, and which modifies the other's spectrum. We found that *phaser* paired with the basis 'w' (the basis of the logically negated word) performs better than the basis 'c' (the basis of the worldly context) across both negations for most plausibility metrics. This lines up with our intuition that applying worldly context updates the eigenspectrum of $\neg(\llbracket w_N \rrbracket)$, leveraging worldly knowledge to increase/decrease the weights of more/less contextually relevant values of the logical negation of w_N. However, a notable exception to this reasoning is our result that for *spider* paired with $\neg_{inv}$, basis 'c' has statistically significant correlations with human ratings, while basis 'w' does not.

6.3.4 Worldly context

For these experiments, we create worldly context based on the hypernym paths provided by WordNet. As explained in Section 4.1.1, we need $p_i \geq p_{i+1}$ in Equation 15 for the more direct hypernyms to be more important than more distant hypernyms. Hence, we tried multiple monotonically decreasing functions for the weights $\{p_i\}_i$ of the hypernyms.

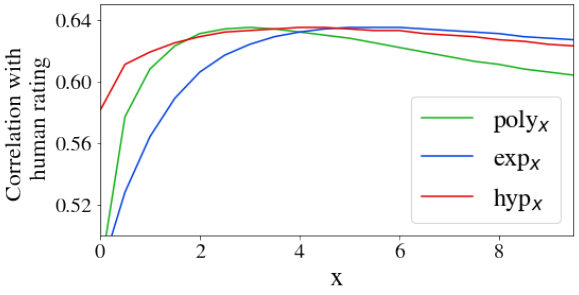


Figure 5: Correlation of results of different context functions with human rating

For a word w with n hypernyms $h_1, ..., h_n$ ordered from closest to furthest, we define the following functions to calculate p_i.

$$\texttt{poly}_x(i) := (n-i)^x \tag{18}$$

$$\texttt{exp}_x(i) := (1+\frac{x}{10})^{(n-i)} \tag{19}$$

$$\texttt{hyp}_x(i) := (n-i)^{\frac{x}{2}} k_{\mathsf{E}}(w, h_i) \tag{20}$$

Figure 5 shows on the y-axis the correlation of the human rating with the plausibility rating (*trace*) of our best conversational negation (*phaser* with $\neg_{sub}$ in the basis 'w') and the parameters of context functions on the x-axis. We observe that all three context functions achieve a maximal correlation of 0.635, therefore being equally good. All functions eventually drop in correlation as the value of x increases, showing that having the context too close to the word does not yield optimal results either. One important observation is that at $x = 0$, $\texttt{hyp}_x(i) = k_{\mathsf{E}}(w, h_i)$ still performs well with a correlation of 0.581, despite not taking the WordNet hypernym distance into account. This is an evidence for the potential of the context creation based on density matrix entailment proposed in Section 4.1.2.

6.3.5 Plausibility rating measures

On top of calculating the conversational negation, the experiments call for comparing the results of the conversational negation with w_A to give plausibility ratings. We compare the hyponymies k_{E}, k_{hyp}, and k_{BA}, as well as *trace similarity*. The results show that *trace similarity* and k_{E2} interact most sensibly with our conversational negation, attaining 0.635 and 0.602 correlation with mean human ratings respectively. For the asymmetric measures k_{E} and k_{hyp}, computing the entailment from w_A to the conversational negation of w_N performed better than the other direction. For all similarity measures (except k_{hyp1}), $\neg_{sub}$ paired with *phaser* in the basis 'w' performs the best.

7 Future work

The framework presented in this paper shows promising results for conversational negation in compositional distributional semantics. Given its modular design, additional work should be done exploring more kinds of logical negations, compositions and worldly contexts, as well as situations for which certain combinations are optimal. Since creating worldly context—as presented in this paper—is a new concept in the area of DisCoCirc, it leaves the most room for further exploration. In particular, our framework does not handle how to disambiguate different meanings of the same word; for example, the worldly context of the word *apple* should be different for the fruit *apple* versus the technology company *apple*.

Our conversational negation framework currently does not model a different kind of negation where the suggested alternative is an antonym rather than just any other word that appears in similar contexts. For instance, the sentence *Alice is not happy* suggests that Alice is *sad*—an antonym of *happy*—rather than *cheerful*, even though *cheerful* might appear in similar contexts as *happy*. We would like to extend the conversational negation framework to account for this.

We would like to implement the context generation method presented in Section 4.1.2 and test on the current experimental setup.[2] To further validate the framework, more data sets should be collected and evaluated on to explore, for each type of relation between words, what construction of conversational negation yields sensible plausibility ratings.

For the conversational negation to be fully applicable in the context of compositional distributional semantics, further theoretical work is required to generalize the model from negation of individual nouns to negation of other grammatical classes and complex sentences. Furthermore, we would like to analyze the interplay between conversational negation, textual context, and evolving meanings. Lastly, the interaction of conversational negation with logical connectives and quantifiers leaves open questions to explore.

[2] The code is available upon request.

Acknowledgements

We would like to give special thanks to Martha Lewis for the insightful conversation and for sharing her code for generating density matrices. We appreciate the guidance of Bob Coecke in introducing us to the field of compositional distributional semantics for natural language processing. We thank John van de Wetering for informative discussion about ordering density matrices. We thank the anonymous reviewers for their helpful feedback. Lia Yeh gratefully acknowledges funding from the Oxford-Basil Reeve Graduate Scholarship at Oriel College in partnership with the Clarendon Fund.

References

Samson Abramsky and Bob Coecke. 2004. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 2004.*, pages 415–425.

Mostafa Ghazizadeh Ahsaee, Mahmoud Naghibzadeh, and S Ehsan Yasrebi Naeini. 2014. Semantic similarity assessment of words using weighted wordnet. *International Journal of Machine Learning and Cybernetics*, 5(3):479–490.

Jerzy K. Baksalary, Friedrich Pukelsheim, and George P.H. Styan. 1989. Some properties of matrix partial orderings. *Linear Algebra and its Applications*, 119:57–85.

Esma Balkir, Mehrnoosh Sadrzadeh, and Bob Coecke. 2016. Distributional sentence entailment using density matrices. In *Topics in Theoretical Computer Science*, pages 1–22, Cham. Springer International Publishing.

Dea Bankova, Bob Coecke, Martha Lewis, and Dan Marsden. 2019. Graded hyponymy for compositional distributional semantics. *Journal of Language Modelling*, 6(2):225–260.

Phil Blunsom, Edward Grefenstette, and Karl Moritz Hermann. 2013. "not not bad" is not "bad": A distributional account of negation. In *Proceedings of the 2013 Workshop on Continuous Vector Space Models and their Compositionality*.

Joe Bolt, Bob Coecke, Fabrizio Genovese, Martha Lewis, Dan Marsden, and Robin Piedeleu. 2017. Interacting conceptual spaces I : Grammatical composition of concepts. *CoRR*, abs/1703.08314.

Jordan Boyd-Graber, Christiane Fellbaum, Daniel Osherson, and Robert Schapire. 2006. Adding dense, weighted connections to wordnet. In *Proceedings of the third international WordNet conference*, pages 29–36. Citeseer.

Bob Coecke. 2020. The mathematics of text structure.

Bob Coecke, Giovanni de Felice, Konstantinos Meichanetzidis, and Alexis Toumi. 2020. Foundations for near-term quantum natural language processing. *ArXiv*, abs/2012.03755.

Bob Coecke and Konstantinos Meichanetzidis. 2020. Meaning updating of density matrices. *FLAP*, 7:745–770.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *Lambek Festschrift Linguistic Analysis*, 36.

Gemma De las Cuevas, Andreas Klinger, Martha Lewis, and Tim Netzer. 2020. Cats climb entails mammals move: preserving hyponymy in compositional distributional semantics. In *Proceedings of SEMSPACE 2020*.

Jonathan St BT Evans, John Clibbens, and Benjamin Rood. 1996. The role of implicit and explicit negation in conditional reasoning bias. *Journal of Memory and Language*, 35(3):392–409.

Christiane Fellbaum. 1998. Wordnet: An electronic lexical database.

Edward Grefenstette and Mehrnoosh Sadrzadeh. 2011. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Coling 1992 volume 2: The 15th international conference on computational linguistics*.

Laurence Horn. 1972. On the semantic properties of logical operators in english. *Unpublished Ph.D. dissertation*.

Germán Kruszewski, Denis Paperno, Raffaella Bernardi, and Marco Baroni. 2016. There is no logical negation here, but there are alternatives: Modeling conversational negation with distributional semantics. *Computational Linguistics*, 42(4):637–660.

Martha Lewis. 2019. Compositional hyponymy with positive operators. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 638–647, Varna, Bulgaria. INCOMA Ltd.

Martha Lewis. 2020. Towards logical negation for compositional distributional semantics. *IfCoLoG Journal of Logics and their Applications*, 7(3).

Mike Oaksford. 2002. Contrast classes and matching bias as explanations of the effects of negation on conditional reasoning. *Thinking & Reasoning*, 8(2):135–151.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Robin Piedeleu, Dimitri Kartsaklis, Bob Coecke, and Mehrnoosh Sadrzadeh. 2015. Open system categorical quantum semantics in natural language processing.

Jérôme Prado and Ira A. Noveck. 2006. How reaction times can elucidate matching effects and the processing of negation. *Thinking and Reasoning*, 12(3).

Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora. *arXiv preprint arXiv:1806.03191*.

John van de Wetering. 2018. Ordering quantum states and channels based on positive bayesian evidence. *Journal of Mathematical Physics*, 59(10):102201.

Dominic Widdows and Stanley Peters. 2003. Word vectors and quantum logic: Experiments with negation and disjunction. *Mathematics of language*, 8(141-154).

A Proofs

A.1 Support inverse reverses k-hyponymy

Theorem 1. *For two density matrices A and B, k-hyponymy is reversed by support inverse when $rank(\mathsf{A}) = rank(\mathsf{B})$:*

$$\mathsf{A} \sqsubseteq_k \mathsf{B} \Longleftrightarrow \neg_{supp}\mathsf{B} \sqsubseteq_k \neg_{supp}\mathsf{A} \tag{21}$$

Proof. From (Baksalary et al., 1989), $\neg_{supp}$ reverses Löwner order when $rank(\mathsf{A}) = rank(\mathsf{B})$:

$$\mathsf{A} \sqsubseteq \mathsf{B} \Longleftrightarrow \neg_{supp}\mathsf{B} \sqsubseteq \neg_{supp}\mathsf{A} \tag{22}$$

Thus, letting "≥ 0" denote the operator is positive:

$$\begin{aligned} \mathsf{A} \sqsubseteq_k \mathsf{B} &\Longleftrightarrow \mathsf{B} - k\mathsf{A} \geq 0 && (23)\\ &\Longleftrightarrow (k\mathsf{A})^{-1} - \mathsf{B}^{-1} \geq 0 && (24)\\ &\Longleftrightarrow \frac{1}{k}\mathsf{A}^{-1} - \mathsf{B}^{-1} \geq 0 && (25)\\ &\Longleftrightarrow \mathsf{A}^{-1} - k\mathsf{B}^{-1} \geq 0 && (26)\\ &\Longleftrightarrow \mathsf{B}^{-1} \sqsubseteq_k \mathsf{A}^{-1} && (27) \end{aligned}$$

using Equations 5 and 22 from Equation 23 to 24. □

Corollary 1. *For two invertible density matrices A and B, k-hyponymy is reversed by matrix inverse:*

$$\mathsf{A} \sqsubseteq_k \mathsf{B} \Longleftrightarrow \mathsf{B}^{-1} \sqsubseteq_k \mathsf{A}^{-1} \tag{28}$$

A.2 Matrix inverse reverses k_{BA} in same basis case

Theorem 2. *For two density matrices A and B with the same eigenbasis, k_{BA} is reversed by matrix inverse:*

$$k_{\mathsf{BA}}(\mathsf{B}^{-1}, \mathsf{A}^{-1}) = k_{\mathsf{BA}}(\mathsf{A}, \mathsf{B}) \tag{29}$$

Proof.

$$\begin{aligned} k_{\mathsf{BA}}(\mathsf{B}^{-1}, \mathsf{A}^{-1}) &= \frac{\sum_i \lambda^i_{\mathsf{A}^{-1}} - \lambda^i_{\mathsf{B}^{-1}}}{\sum_i \left|\lambda^i_{\mathsf{A}^{-1}} - \lambda^i_{\mathsf{B}^{-1}}\right|} && (30)\\ &= \frac{\sum_i \frac{1}{\lambda^i_{\mathsf{A}}} - \frac{1}{\lambda^i_{\mathsf{B}}}}{\sum_i \left|\frac{1}{\lambda^i_{\mathsf{A}}} - \frac{1}{\lambda^i_{\mathsf{B}}}\right|} && (31)\\ &= \frac{\sum_i \lambda^i_{\mathsf{B}} - \lambda^i_{\mathsf{A}}}{\sum_i \left|\lambda^i_{\mathsf{B}} - \lambda^i_{\mathsf{A}}\right|} && (32)\\ &= k_{\mathsf{BA}}(\mathsf{A}, \mathsf{B}) && (33) \end{aligned}$$

using Equation 13 from Equation 30 to 31. □

A.3 Composing with $\neg_{sub}$ or $\neg_{inv}$ gives maximally mixed support

Theorem 3. *When composing a density matrix X with $\neg_{supp}\mathsf{X}$ via* spider, fuzz, *or* phaser, *the resulting density matrix has the desired property of being a maximally mixed state on the support with zeroes on the kernel.*

Proof. $\neg_{supp}\mathsf{X}$ and X have the same eigenbasis. From Equation 13, all nonzero eigenvalues of $\neg_{supp}\mathsf{X}$ are multiplicative inverses of the corresponding eigenvalue of X.
We use definitions of *spider*, *fuzz*, and *phaser* from Equations 1, 2, and 3. The summation indices are over eigenvectors with nonzero eigenvalue.

$$\begin{aligned} &spider(\mathsf{X}, \neg_{supp}\mathsf{X}) && (34)\\ &= U_s(\mathsf{X} \otimes \neg_{supp}\mathsf{X})U_s^\dagger && (35)\\ &= \Big(\sum_i |i\rangle \langle ii| \Big)(\mathsf{X} \otimes \neg_{supp}\mathsf{X})\Big(\sum_j |jj\rangle \langle j| \Big) && (36)\\ &= \sum_i |i\rangle \langle ii| \Big((\lambda\, |i\rangle \langle i|) \otimes (\frac{1}{\lambda_i} |i\rangle \langle i|) \Big) |ii\rangle \langle i| && (37)\\ &= \sum_i |i\rangle \langle i| && (38)\\ &= \mathbb{I}_{supp} && (39) \end{aligned}$$

$$\begin{aligned} fuzz(\mathsf{X}, \neg_{supp}\mathsf{X}) &= \sum_i x_i P_i \circ \mathsf{X} \circ P_i && (40)\\ &= \sum_i \frac{1}{\lambda_i} P_i \Big(\sum_j \lambda_i P_i\Big) P_i && (41)\\ &= \sum_i P_i && (42)\\ &= \mathbb{I}_{supp} && (43) \end{aligned}$$

$$\begin{aligned} &phaser(\mathsf{X}, \neg_{supp}\mathsf{X}) && (44)\\ &= \Big(\sum_i x_i P_i\Big) \circ \mathsf{X} \circ \Big(\sum_i x_i P_i\Big) && (45)\\ &= \Big(\sum_i \lambda_i{}^{-\frac{1}{2}} P_i\Big)\Big(\sum_j \lambda_j P_j\Big)\Big(\sum_k \lambda_k{}^{-\frac{1}{2}} P_k\Big) && (46)\\ &= \sum_i P_i && (47)\\ &= \mathbb{I}_{supp} && (48) \end{aligned}$$

□

Corollary 2. *When composing a density matrix* X *with* $\neg_{inv}\mathsf{X}$ *via* spider, fuzz*, or* phaser*, the resulting density matrix has the desired property of being a maximally mixed state on the support with zeroes on the kernel.*

Parsing Conjunctions in DisCoCirc

Tiffany Duneau
University of Oxford
fanny.duneau@cs.ox.ac.uk

Abstract

In distributional compositional models of meaning logical words require special interpretations, that specify the way in which other words in the sentence interact with each other. So far within the DisCoCat framework, conjunctions have been implemented as merging both conjuncts into a single output, however in the new framework of DisCoCirc merging between nouns is no longer possible. We provide an account of conjunction and an interpretation for the word *and* that solves this, and moreover ensures certain intuitively similar sentences can be given the same interpretations.

1 Introduction

The distributional semantics paradigm allows us to model the meanings of words in terms of their surrounding words, as well as the meaning of texts in terms of their constituent words. Meanwhile compositional semantics considers how meaning can arise from the grammatical form of a sentence. The distributional compositional (DisCo-) approach to modelling meaning introduced by Coecke et al. (2010) allows the meaning of a sentence to be computed as a function of both the distributional meaning of the words involved, as well as its grammatical form. Recent work by Lorenz et al. (2021) has implemented the approach on quantum hardware, and Kartsaklis and Sadrzadeh (2016); Lewis (2019) have shown the classical model to perform well at various natural language processing tasks.

However, there are certain words whose meaning cannot be expressed as a function of their surrounding words - logical words like *and, or* and *not*, as well as pronouns such as *whom* and *that* - since they tend to appear in all contexts and thus render distributional approaches to modelling them inadequate. On the other hand, such words typically have an impact on the way in which the other words in the sentence interact, and thus take on a more syntactic role, acting as an extension of the grammar. In this paper, we shall focus in particular on the logical connective *and*.

Previously, in the DisCo- formalisms conjunctions have been interpreted as simply mixing the conjuncts involved together to produce a single entity of the appropriate type: as described in Kartsaklis (2016), we can mix the adjectives *red* and *yellow* together to make a new adjective that describes things that are both red and yellow. This approach works less well however, when we attempt to mix nouns; when we discuss *a hat and a scarf*, we are not discussing a single hybrid object that is both a hat and a scarf, but *two* objects, one of which is a hat, the other of which is a scarf. This difference suggests that there are two different types of conjunctions at play. The DisCoCirc framework allows us to express this difference - in the introductory paper Coecke (2020), they are denoted as *linear* and *non-linear* forms of *and*.

The non-linearity can be seen as arising from an induced duplication:

1a Alice wears red and yellow socks.

b Alice wears red socks. Alice wears yellow socks.

2a Alice wears a hat and a scarf.

b Alice wears a hat. Alice wears a scarf.

Sentence (1a) employs a linear form of *and* - in this case both adjectives are applied to the socks which then feeds into the rest of the sentence. We can contrast this with the non-linear interpretation of the same sentence in (1b), which does not seem to convey the same meaning. On the other hand, (2a) seems to carry the same meaning as (2b) which exhibits the implicit duplication of *wear* induced by the conjunction. This distinction was previously

Proceedings of the 2021 Workshop on Semantic Spaces at the Intersection of NLP, Physics, and Cognitive Science, pages 66–75
June 16th, 2021. ©2021 Association for Computational Linguistics

lost as conjoined nouns were considered equivalent to a single noun - this meant that any duplication would have occurred at best implicitly.

Standard DisCo- approaches are often limited to linear operations only, as they have standardly been implemented using vector spaces and linear maps, however certain phenomena seem to require non-linear elements: in Lewis (2020), negation is modelled as a (non-linear) projection of positive maps onto their orthogonal subspaces, while (Wijnholds and Sadrzadeh, 2019a) shows how modelling duplication *explicitly* improves performance in sentence similarity and entailment tasks involving ellipsis and anaphora. Wijnholds and Sadrzadeh (2019b) describes how the duplication is introduced, by augmenting the basic pregroup based grammars with a non-linear reduction rule.

Here, we will also be introducing a non-linear element, modifying word meanings. In particular, this will allow us to model the duplication in sentences like (1) and (2), such that the interpretation of sentences we consider equivalent are the same when expressed using the DisCoCirc framework. We will first give a brief overview of this framework, then introduce the required structures and definitions for modelling *and*, finishing with some examples.

2 Mathematical Background

We will give a brief overview of the mathematical background for the DisCo- approaches to modelling meaning. For a detailed introduction to the category theory see Heunen (2020), while Coecke and Kissinger (2017); Selinger (2009) provide a more diagrammatic treatment. Coecke et al. (2010); Coecke (2020) introduce the DisCoCat and DisCoCirc frameworks respectively.

2.1 Compact Closed Categories

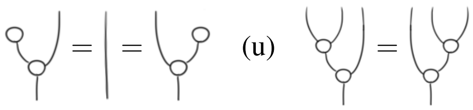
We will be encoding sentences as diagrams, in which wires will carry meanings, and boxes will represent processes that modify these meanings. These diagrams are representations of structures in a compact closed category; we will take the convention of reading the diagrams from top to bottom, and will sometimes add a direction to the wire to indicate the presence of a dual. The diagrams come with an associated calculus, framed as a set of rewrite rules. Diagrams and the structures they represent are equivalent if and only if there is a valid way to rewrite one into the other.

We can use such diagrams to express syntactic relations between the words of a sentence. In the present case, this is what we are most interested in, as conjunctions impart meaning mostly by imposing extra syntactic dependencies between other words in the sentence. In order to get back a representation of what the diagram *means*, however, we also need to supply a way to *interpret* diagrams, often within a specific compact closed category. Standardly, this has been as linear maps between vector spaces or relations between sets, though other categories have been used too (Bankova et al., 2016; Bolt et al., 2017). Here, we will not be concerned with the specific meanings, only the ways in which the words are connected. As such, we can take ourselves to be working in the category freely generated by a chosen set of types and boxes, along with the required cups, caps and swaps that make the category compact closed.

2.2 Monoids

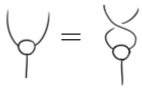
Monoids encapsulate the idea of combining and splitting objects, and so form a vital part of our theory of meaning, and are particularly relevant to the notion of conjunction.

Definition 1. *A* ***monoid*** $(A, ⊻, ○)$*, is structure over an object* A *that satisfies unitality (u) and associativity (a) axioms:*



(a)

Commutative monoids satisfy the additional condition (c):



(c)

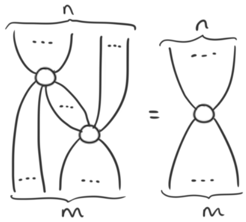
A **comonoid** structure $(A, ▲, ●)$, is a vertically flipped version of the monoid, satisfying the analogously flipped axioms.

2.2.1 Spiders

Certain monoid-comonoid pairs satisfy additional rules[1], which allows us to write the dots all in the same colour, as they may be interchanged. In particular, as associativity identifies all orders of composition, and we may write sequences of monoid

[1] Namely, when the comonoid is the dagger of the monoid and they are special and Frobenius: Heunen (2020) chapter 5

or comonoid applications as a single dot with many legs. These dots obey the 'spider fusion' rule:



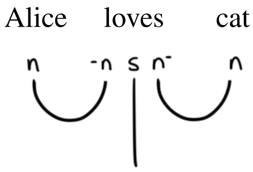
(SF)

2.3 Pregroups

Having created all the above structure to encode meaning, we now turn to grammar. A *pregroup* is a group where we may have distinct left and right inverses for each element. Extending this into pregroup grammar in the style of Lambek (1968) and Bar-Hillel (1953), we start with some generating elements, or basic grammatical types: 's' (sentence) and 'n' (noun), along with a unit element I, and their respective left and right inverses. More complex types are composed by stringing the basic types together via the multiplication operation: transitive verbs such as *loves* and *sits on* are of the form '${}^{-}n\ s\ n^{-}$', as they need a noun on either side to form a sentence. The multiplication of the pregroup is given by a partial order on the strings generated by the basic types:

$$x \cdot {}^{-}x \le I \qquad\qquad x^{-} \cdot x \le I$$
$$I \le {}^{-}x \cdot x \qquad\qquad 1 \le x \cdot x^{-}$$

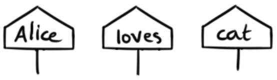
These equations correspond exactly to the cups and caps within a compact closed category if we take I to be the monoidal unit. Indeed the category formed by taking the grammatical types as objects, string concatenation as parallel composition, and the reductions to define morphisms between objects is compact closed. We can hence write grammatical reductions out graphically. '*Alice loves [the] cat*' has grammatical type '$n\ {}^{-}nsn^{-}\ n$', and reduces as follows:



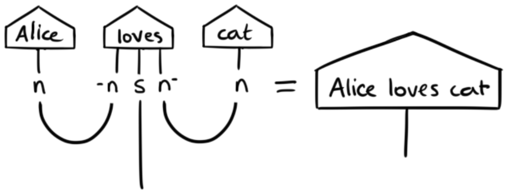
If, as with this example there is a way to reduce the grammatical type to a single sentence wire, then the sentence is grammatical. Of note is that there may sometimes be different ways in which the same string can be reduced. These correspond to different possible grammatical interpretations of the sentence. In forming reductions, we are not allowed to cross wires over each other, as this would allow some non grammatical strings such as '*loves cat Alice*' to be reduced too, by swapping word order as part of the reductions.

2.4 DisCoCat

DisCoCat is a *dis*tributional *co*mpositional *cat*egorical model of meaning, that is formulated in a compact closed category, introduced in Coecke et al. (2010). The distributional aspect of the model concerns word meaning, often encoded as vectors. The easiest way of combining our words into a sentence is to simply write them next to each other:



However this fails to capture any meaningful grammatical relationship between the words, other than perhaps word order. We hence add a grammatical type to each of our wires, and allow the grammar to mediate how the words *compose* to give the final sentence meaning, linking the wires together using cups and caps:

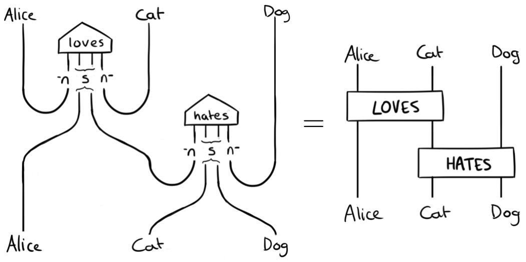


As mentionned above, there are some words for which we hard code the meaning - logical words like *and* and *not* (Kartsaklis, 2016), as well as relative pronouns like *which* and *whom* (Sadrzadeh et al., 2013, 2016). In many of these cases the 'copy' spider has been used, which copies or merges wires relative to a given basis. Importantly however, this spider is a *linear* operation so cannot truly duplicate anything.

Some words will also need to be assigned an ambiguous grammatical type, if they can occur in different contexts; the correct type to use can be informed by considering the choice that allows the surrounding sentence to be reduced into a sentence.

2.5 DisCoCirc

DisCoCat allows us to encode single sentences to obtain a meaning vector describing the entire sentence. Going one step further, DisCoCirc allows us to compose multiple sentences together, and model the meaning of an entire text. This involves shifting our perspective slightly, so that we are considering the way the meaning of words are altered by a sentence, for example the way a character's name changes meaning as we read a novel and learn more about them. Rather than having a single sentence type, we hence move to taking our types to be the *dynamic objects*, like *The cat* or *Alice*, that feature in the sentences we are concerned with. Each sentence is then still tied together in much the same way as in DisCoCat - according to the grammar - but the outgoing sentence type is now a composite of the dynamic objects. The overall sentence becomes a box that acts on these dynamic objects, rather than just a state, allowing it to be composed with other sentences. A sentence acts as the identity on any object that is not involved in it. For example, we can now encode '*Alice loves the cat. The cat hates the dog.*' as follows:



Since we are now composing sentences together, we allow wires to cross freely *between* sentences, though no extra crossings may occur in the grammatical reductions.

By encoding the meaning of texts in terms of the meanings of the characters within them, we sacrifice having a common space in which all texts are encoded, and all the associated benefits for sentence comparison. This remains an interesting move to make if we are interested in making a more involved analysis of the *content* of the texts in question, as we can retain a rather detailed representation of what is happening without needing to squeeze everything down into a fixed space. This has a particular impact for the modelling of and, as it will require us to introduce a notion of duplication.

3 New structure

In addition to the basic structures presented above, we shall make use of a few extra components in order to model and.

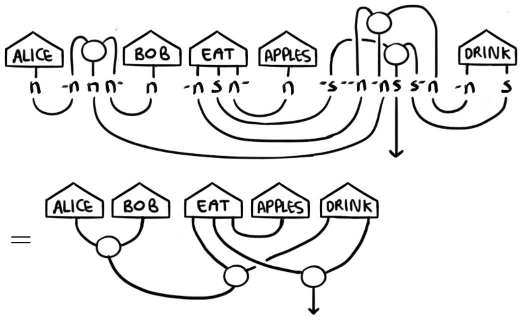
3.1 Fine-graining of the n type

Much of the work done with DisCoCat and DisCoCirc so far makes use of only n and s as generators for the grammar. In DisCoCirc, it becomes necessary to break the usual n type down into two different types, so that we can treat them differently when encoding them. The distinction we need is between *singular* (which we will keep as n) and *plural* (denoted N) nouns[2]. Singular nouns behave just as we expect, since they contain exactly one noun wire, whereas plural nouns are in fact a series of singular nouns that have been packaged together. Formally, this 'packaging together' occurs via the monoidal tensor, and indeed the sentence type in DisCoCirc corresponds to the plural noun type N. We will still make use of the sentence type s, to ensure that the grammar is not resolved differently, and so that no grammatically incorrect sentences are accepted as a result.

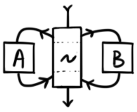
3.2 Merge box

In order to model and, we will need a notion of merging. Previously (Kartsaklis, 2016), a spider was used:

- Alice and Bob eat apples and drink.



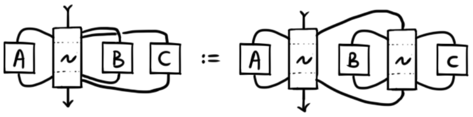
However, the notion of merging a pair of boxes is much more general than this particular choice. Indeed, a series of methods for combining density matrices has been surveyed in Cuevas et al. (2020); Coecke and Meichanetzidis (2020). The basic form of such a map is as follows:



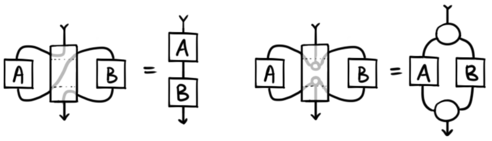
Here we have taken some liberties in drawing the diagram connections - concretely the wires coming

[2] In Lambek (1968), a different sort of plural noun is considered, typed n^*, which seems more aimed at dealing with generic nouns (*men*; *strawberries*) rather than multiple copies of n (*two strawberries*; *men and women*) as introduced here.

out of the sides of the box should be considered as coming out of the top or bottom (whichever is closest), however we draw them differently to highlight them as 'intermediate' wires, each of which is representing a particular branch that will be merged. Merging more than two wires together is defined recursively in terms of a two way merge:

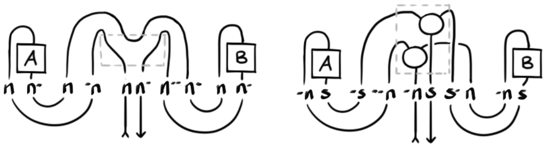


In order to obtain a useful merging operation, we will restrict our attention to associative binary operations - semigroups - and in particular consider two natural examples (where SPIDER corresponds to Mult in Cuevas et al. (2020), and SEQ is sequential composition.):

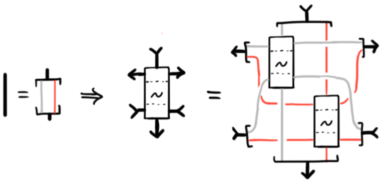


(SEQ) (SPIDER)

On top of being associative, these operations are actually based on monoids - the copy spider and 'pair of pants' monoid respectively. This form is most clear when we draw the merge with it's wires paired in a particular way:



We extend the merging to plural wires by merging each component wire separately[3]:

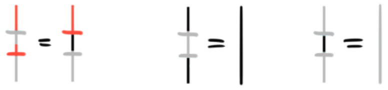


3.3 Duplication 2-Morphism

The main ingredients that will help us formulate conjunctions are a 2-morphism, or meta operation which we will denote $[//]$, along with a marker morphism $//$, and colour typing morphisms $\vdash$, $\dashv$.

3.3.1 Wire colours

In order to control the 'footprint' of the $[//]$ morphism, we make use of certain extra typing annotations, which we will represent using wire colours. These annotations stack rather than mix - a red and blue wire is not the same as a purple wire. We take 'uncoloured' wires to be black, with the colours controlled by 'start' ($\vdash$), and 'stop' ($\dashv$) morphisms.

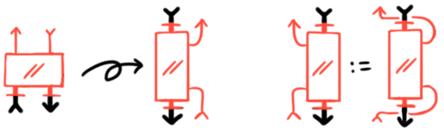


Commute Identity

As these colours have nothing to do with the meanings carried by the wire, we can view them as living in a separate sub-wire, conjoined to the main meaning wire. Supposing that the compact closed category $\boldsymbol{Col}$ admits colour morphisms like the above, and that our main 'semantic' category is $\boldsymbol{S}$, we can define a new category of semantics with colour annotations: $\boldsymbol{S} \times \boldsymbol{Col}$. This new category will also be compact closed, inheriting the relevant structure from the compact closure of both $\boldsymbol{S}$ and $\boldsymbol{Col}$, and can hence replace $\boldsymbol{S}$ as the category in which we are expressing our meanings.

3.3.2 Marker morphism

Each time we want to duplicate a wire, we will introduce a 'marker' morphism, $//$, to indicate where the duplication should occur. Such a morphism will have an associated colour, and type signature to indicate how the wires are to be split by the duplication operation. We will draw it as follows[4]:



The thick N or S type wires contain the series of n wires to duplicate over, while the thin coloured wire highlights the part of the diagram in need of duplication, standing in for a given n wire.

3.3.3 The duplication 2-morphism

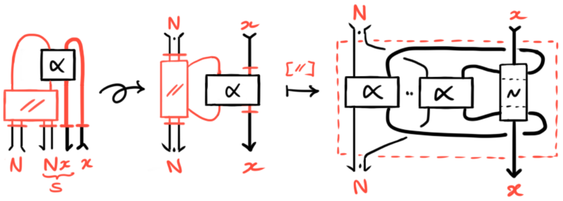
Informally, a **2-morphism** is a morphism defined between the *morphisms* (rather than objects) of a category. Analogously, it can be viewed as specifying an extra rewrite rule for our diagrams - the new diagram obtained may not be strictly equal to the

[3] The N typed wires have been drawn thicker to highlight that they are actually a collection of wires. The bracketing or gathering morphism should only be taken as a notational tool that highlights which specific n wires contribute to the plural N wire.

[4] When expressing word states using $//$ we will tend to write it horizontally, whereas within a full DisCoCirc diagram it will be more convenient to write the morphism vertically.

previous one, as per the usual rewrite rules, however the 2-morphism defines a sense in which the diagrams should be viewed as equivalent. In our case, we will be introducing a 2-morphism that will deal with duplication, as there is no corresponding (1-)morphism in a compact closed category that can do the job.

Definition 2. *The 2-morphism [//] copies boxes and introduces a mixing operation as follows:*



Importantly, we apply [//] centered on a marker, exclusively to the largest sub-diagram that has wires coloured the same way as the marker. For this to be well defined, we hence require that each instance of // be uniquely coloured. The resulting diagram applies the coloured sub-diagram to each marked wire, merging any other wires together according to a chosen merge operation.

If the underlying mixing operation is both associative and commutative, the order in which we apply the [//] is associative also (for a proof see the appendix). In such a case, there will be a unique normal form for the diagram that contains no marker boxes.

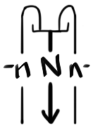
A full expansion of the marker boxes in a diagram renders any leftover colour annotations irrelevant to the meaning of the sentence. The final step in computing the DisCoCirc diagram corresponding to a given sentence involving such markers will then be to apply a forgetful functor from $\boldsymbol{S} \times \boldsymbol{Col}$ into $\boldsymbol{S}$, removing the now unnecessary colour subwire.

4 Modelling *and*

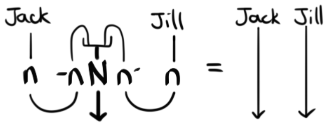
Having set the scene, we are now equipped to start modelling the word *and*. The first aspect to note, is that *and* has a variable type, of the form ${}^{-}x\ x\ x^{-}$, where x can stand for any (possibly composite) grammatical type. The account we give will hence be equally generic, though we proceed mostly via example.

4.1 Looking inside the box

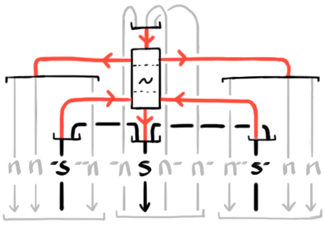
In the most basic case, conjoining two nouns is just a case of packaging them next to each other:



- Jack and Jill.

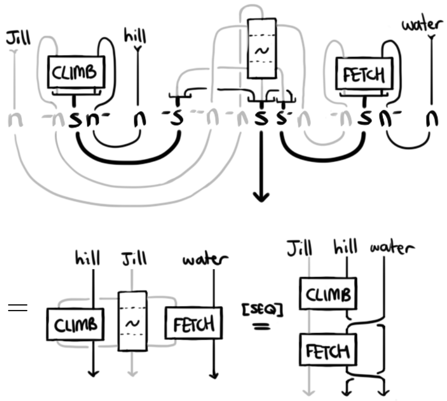


Next, conjoining boxes is just a case of merging them together. In some cases, the components we want to merge may not involve the same nouns - in order for the types to match we cannot merge them, so must combine them into the outgoing sentence wire directly. These extra nouns are drawn in black below:

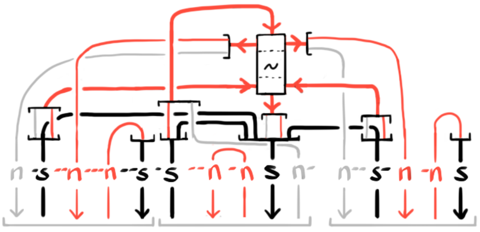


By varying the number of noun wires coming in or out appropriately, we can use the above to construct forms for adjectives ($n\ n^{-}$), simple verbs (${}^{-}n\ s$), transitive verbs (${}^{-}n\ s\ n^{-}$), and di-transitive verbs (${}^{-}n\ s\ n^{-}\ \ n^{-}$). Removing all the nouns results in a sentence combiner that acts analogously to the simple noun case. To illustrate, consider:

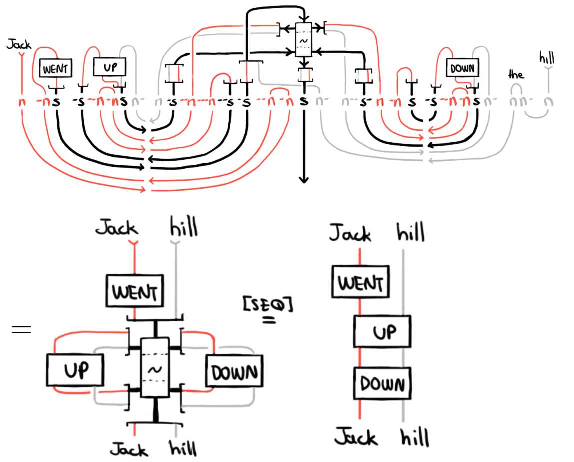
- Jill climbs hills and fetches water.



Another important structure is prepositions, typed $^{-}s \;\; ^{--}n \;\; ^{-}n\ s\ n^{-}$. In this case, a specific subject noun (in red), and an indirect object (in light grey) are identified in advance, and are subsequently passed to the two conjuncts to be mixed. Again, we collect any extra nouns picked up along the way and ensure they bypass the merging. Similarly to above, we can also add or remove as many light grey nouns as necessary to type match, following the schema given.



- Jack went up and down the hill.



The internal wirings for other types will largely follow the same format, merging together wires that are involved on both sides of the conjunct and combining the rest into an outgoing N or s plural noun.

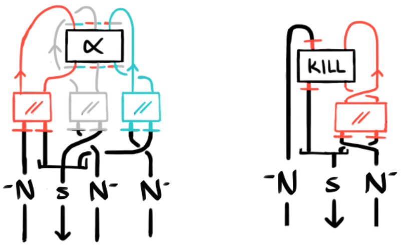
4.2 Dealing with N

The way in which we are conjoining nouns is introducing a new plural noun type N. In order for our sentence parsings to keep working, we hence need to introduce new forms for our other boxes that include N where we would usually have n. Bearing in mind that our goal is for the grammar wirings to look the same, this means that we are looking to convert N into n internally to the box in question.

The intuition behind this split is that usually, the action described by a box is not shared between the parties involved - instead each character does the given action independently. For example, consider the following sentences:

1. Alice and Bob eat cucumber.
2. Alice eats cucumber. Bob eats cucumber.
3. Alice and Bob murder Charlie.
4. Alice murders Charlie. Bob murders Charlie.

In (1), Alice and Bob are not sharing the same '*eating*', and the sentence seems equivalent to (2), suggesting that the verb is simply duplicated to accommodate the multiple subjects. On the other hand, in (3), there is only one murder, so Alice and Bob must be doing it together, suggesting that in this case we cannot duplicate the single-subject form. Indeed, (4) seems somewhat contradictory - when we learn that Alice murders Charlie, we assume that Charlie is then dead. Subsequently learning that Bob murders Charlie too appears odd, since as far as we are concerned there is no Charlie left for Bob to kill. We can express this difference with the following expansions of the verb boxes:



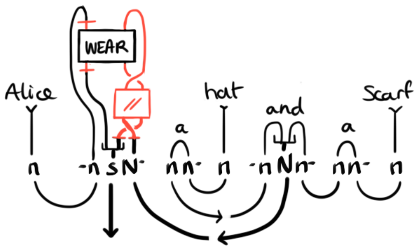
Generic verb

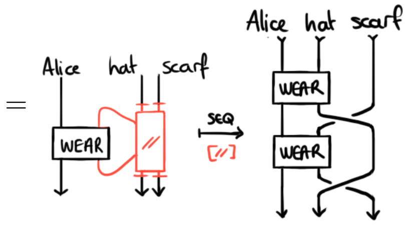
Kill

4.3 Examples

Having established the theory, we can now illustrate how this works with some actual sentences. First, consider the non-linear example given at the start:

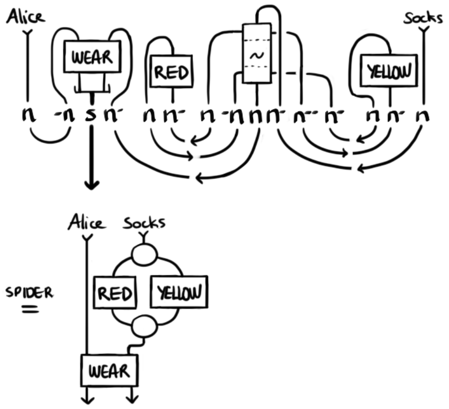
- Alice wears a hat and a scarf.





In this case, the hat and scarf are first packaged into a plural noun, which when fed into the verb *wears* induces a duplication. Taking the merge operation to be given by (SEQ), the final diagram obtained is equivalent to that of '*Alice wears a hat. Alice wears a scarf*'.

- Alice wears red and yellow socks.



In this linear case, the key operation is the merge - no duplication is necessary. The natural choice in this case seems to be (SPIDER), which puts both sides in parallel, suggesting that perhaps (SPIDER) should be used within *and* boxes, whilst (SEQ) should be the merge introduced by the duplication.

In taking the non-commutative (SEQ) as our merge operation, however, we will sometimes need to make an arbitrary choice about the order in which we apply [//]. This problem notably occurs with di-transitive verbs:

- Alice and Bob show Mary apples and pears.

In such cases, the various expansions are arguably different, as they seem to introduce a temporal ordering on the events described, which is not present when there is no common wire connecting the duplicates:

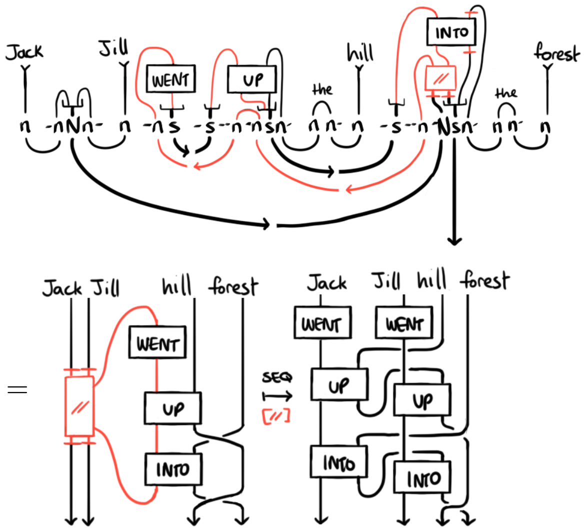
- Alice shows Mary apples. Alice shows Mary pears. Bob shows Mary apples. Bob shows Mary pears.

- Alice shows Mary apples. Bob shows Mary apples. Alice shows Mary pears. Bob shows Mary pears.

The solution would then either be to use a commutative merge like (SPIDER), which would result three different sentences; or to keep (SEQ), but attribute the non-commutativity to the verb *show* instead. In this case, it seems we should model *show* like *kill*, suggesting a link between whether an expansion of the verb is induced by *and*, and whether the verb's meaning is closely linked to the relative time at which it occurs.

Next, we consider a slightly more involved sentence which exhibits the way in which the plural nouns are fed through the sentence:

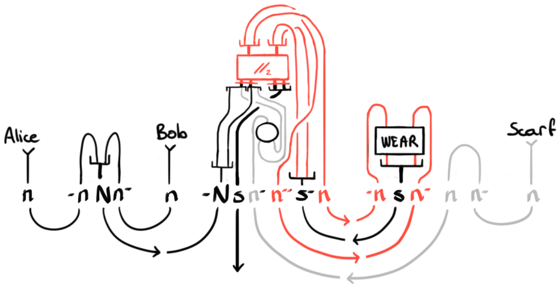
- Jack and Jill went up the hill into the forest.

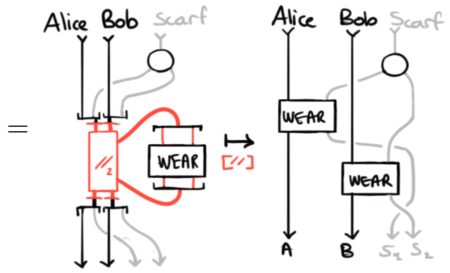


In this case, we can see that the plural noun is fed into the sub-phrase in which it occurs via the marker morphism, so that the sentence expands into the same diagram that would be given by '*Jack went up the hill into the forest. Jill went up the hill into the forest*', as expected.

The duplication operation can also be used with paired wires. In this case, the pairs will be fed through the inner phrase together, rather than independently. For example, consider:

- Alice and Bob *each* wear a scarf.

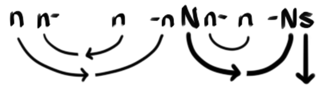




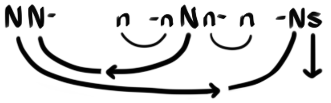
In this case, we have provided a particular interpretation of *each* that coordinates the subject nouns with a copy of the object noun before being fed through the marker morphism. This ensures that Alice and Bob are wearing different copies of the scarf, rather than both wearing the same scarf, or both wearing both scarves. Taking advantage of this distinction also seems like a promising way to model words with similar effects, such as *respectively*.

Finally, in the framework presented we can also account for conjunctive ambiguity, which arises as different ways of bracketing the conjuncts:

(Clumsy Jack) and Jill fell.



Clumsy (Jack and Jill) fell.



The ambiguity here is resolved by our choice of where to place the plural N types. In the first case, we assign *clumsy* a singular adjective type ($n\ n^{-}$), whereas in the second sentence it is plural ($N\ N^{-}$). Making sure to index the N types with the number of wires they are encapsulating, the grammatical ambiguity present can hence be isolated to the type assignment rather than the parsing tree.

5 Related Work

The non-linearity introduced here has very similar effects to the non-linearity Wijnholds and Sadrzadeh (2019b) introduce to deal with ellipsis, however arises rather differently. In particular, they introduce the non-linearity to the grammar parsing, effectively by allowing parts of the derivation to freely be reused elsewhere. In this way, arbitrary parts of the sentence can get copied over to a new location. In contrast, the duplication introduced here is via the words involved, such that the duplication can only occur along the wires given by the (linear) pregroup grammar. The duplication also does not apply to wires themselves - only boxes get copied, making this a strictly weaker notion.

Though we have not discussed a particular choice of meaning embedding, the approach relates to work encoding logical words using vector-based semantics. Within DisCoCirc, we are primarily concerned with *characters* - treating the nouns involved as individual entities. In some cases, however we might be interested in nouns as *concepts*, for which conjunction does not seem to imply the presence of multiple characters, but instead suggests a merging of the component concepts. Aerts (2009) discusses such concepts, in particular considering the so-called *Guppy effect* (*Guppy* is considered a good example of the concept *pet fish* despite being a bad example of both *fish* and *pet* when the concepts are considered separately). He suggests representing the joint concept *pet and fish* as a superposition in a Fock space to account for this: $|pet\ and\ fish\rangle := (|pet\rangle \otimes |fish\rangle) \oplus (|pet\rangle + |fish\rangle)$ In the account of conjunction given here, we effectively split the two ways of combining concepts: the tensor is used on 'characters', while everything else is merged - potentially as a superposition.

6 Conclusion

In summary, we have provided an account of *and* within the framework of DisCoCirc, that allows us to generate diagrams capturing the intuitive meaning of sentences that involve conjunctions. Due to the non-linear nature of *and*, we associate it to a diagram rewriting operation or 2-morphism, that duplicates the relevant parts of our diagrams.

The treatment provided allows us to parse the common usage of *and*, however there is more work to be done when it comes to certain more complex cases. In particular, there are many related words that control how and whether to duplicate words, such as *respectively* and *each*, as well as certain phrases like *three times* which interact with duplication in more complex ways. Ellipsis is also a closely related grammatical notion, and it would be interesting to see if the duplication approach explored here can provide a suitable solution.

References

Diederik Aerts. 2009. Quantum Structure in Cognition. *Journal of Mathematical Psychology*, 53(5):314–348. ArXiv: 0805.3850.

Desislava Bankova, Bob Coecke, Martha Lewis, and Daniel Marsden. 2016. Graded Entailment for Compositional Distributional Semantics. *arXiv:1601.04908 [quant-ph]*. ArXiv: 1601.04908.

Yehoshua Bar-Hillel. 1953. A Quasi-Arithmetical Notation for Syntactic Description. *Language*, 29(1):47.

Joe Bolt, Bob Coecke, Fabrizio Genovese, Martha Lewis, Dan Marsden, and Robin Piedeleu. 2017. Interacting Conceptual Spaces I : Grammatical Composition of Concepts. *arXiv:1703.08314 [cs]*. ArXiv: 1703.08314.

Bob Coecke. 2020. The Mathematics of Text Structure. *arXiv:1904.03478 [quant-ph]*. ArXiv: 1904.03478.

Bob Coecke and Aleks Kissinger. 2017. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, Cambridge.

Bob Coecke and Konstantinos Meichanetzidis. 2020. Meaning updating of density matrices. *arXiv:2001.00862 [quant-ph]*. ArXiv: 2001.00862.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical Foundations for a Compositional Distributional Model of Meaning. *arXiv:1003.4394 [cs, math]*. ArXiv: 1003.4394.

Gemma De las Cuevas, Andreas Klingler, Martha Lewis, and Tim Netzer. 2020. Cats climb entails mammals move: preserving hyponymy in compositional distributional semantics. *arXiv:2005.14134 [cs, math]*. ArXiv: 2005.14134.

Christiaan Johan Marie Heunen. 2020. *Categories for quantum theory: an introduction [electronic resource]*, first edition. edition. Oxford graduate texts in mathematics. University Press, Oxford.

Dimitri Kartsaklis. 2016. Coordination in Categorical Compositional Distributional Semantics. *Electronic Proceedings in Theoretical Computer Science*, 221:29–38. ArXiv: 1606.01515.

Dimitri Kartsaklis and Mehrnoosh Sadrzadeh. 2016. Distributional Inclusion Hypothesis for Tensor-based Composition. *arXiv:1610.04416 [cs]*. ArXiv: 1610.04416.

Joachim Lambek. 1968. The Mathematics of Sentence Structure. *Journal of Symbolic Logic*, 33(4):627–628.

Martha Lewis. 2019. Modelling hyponymy for DisCoCat.

Martha Lewis. 2020. Towards logical negation for compositional distributional semantics. *arXiv:2005.04929 [cs, math]*. ArXiv: 2005.04929.

Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis, and Bob Coecke. 2021. QNLP in Practice: Running Compositional Models of Meaning on a Quantum Computer. *arXiv:2102.12846 [quant-ph]*. ArXiv: 2102.12846.

Mehrnoosh Sadrzadeh, Stephen Clark, and Bob Coecke. 2013. The Frobenius anatomy of word meanings I: subject and object relative pronouns. *Journal of Logic and Computation*, 23(6):1293–1317. ArXiv: 1404.5278.

Mehrnoosh Sadrzadeh, Stephen Clark, and Bob Coecke. 2016. The Frobenius anatomy of word meanings II: possessive relative pronouns. *Journal of Logic and Computation*, 26(2):785–815. ArXiv: 1406.4690.

Peter Selinger. 2009. A survey of graphical languages for monoidal categories. *arXiv:0908.3347 [math]*. ArXiv: 0908.3347.

Gijs Wijnholds and Mehrnoosh Sadrzadeh. 2019a. Evaluating Composition Models for Verb Phrase Elliptical Sentence Embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 261–271, Minneapolis, Minnesota. Association for Computational Linguistics.

Gijs Wijnholds and Mehrnoosh Sadrzadeh. 2019b. A Type-Driven Vector Semantics for Ellipsis with Anaphora Using Lambek Calculus with Limited Contraction. *Journal of Logic, Language and Information*, 28(2):331–358.

Should Semantic Vector Composition be Explicit? Can it be Linear?

Dominic Widdows
LivePerson Inc.
dwiddows@liveperson.com

Kristen Howell
LivePerson Inc.
khowell@liveperson.com

Trevor Cohen
University of Washington
cohenta@uw.edu

Abstract

Vector representations have become a central element in semantic language modelling, leading to mathematical overlaps with many fields including quantum theory. Compositionality is a core goal for such representations: given representations for 'wet' and 'fish', how should the concept 'wet fish' be represented?

This position paper surveys this question from two points of view. The first considers the question of whether an explicit mathematical representation can be successful using only tools from within linear algebra, or whether other mathematical tools are needed. The second considers whether semantic vector composition should be explicitly described mathematically, or whether it can be a model-internal side-effect of training a neural network.

A third and newer question is whether a compositional model can be implemented on a quantum computer. Given the fundamentally linear nature of quantum mechanics, we propose that these questions are related, and that this survey may help to highlight candidate operations for future quantum implementation.

1 Introduction

Semantic composition has been noted and studied since ancient times, including questions on which parts of language should be considered atomic, how these are combined to make new meanings, and how explicitly the process of combination can be modelled.[1]

As vectors have come to play a central role in semantic representation, these questions have naturally become asked of semantic vector models. Early examples include the weighted summation of term vectors into document vectors in information retrieval (Salton et al. 1975) and the modelling of variable-value bindings using the tensor product[2] in artificial intelligence (Smolensky 1990). The use of vectors in the context of natural language processing grew from such roots, landmark papers including the introduction of Latent Semantic Analyis (Deerwester et al. 1990), where the vectors are created using singular value decomposition, and Word Embeddings (Mikolov et al. 2013), where the vectors are created by training a neural net to predict masked tokens.

During the 20th century, logical semantics was also developed, based very much upon the discrete mathematical logic tradition of Boole (1854) and Frege (1884) rather than the continuous vector spaces that developed from the works of Hamilton (1847) and Grassmann (1862). Thus, by the beginning of this century, compositional semantics was developed mathematically, provided frameworks such as Montague semantics for connecting the truth value of a sentence with its syntactic form, but provided little insight on how the atomic parts themselves should be represented. Good examples in this tradition can be found in Gamut (1991) and Partee et al. (1993). In summary, by the year 2000, there were distributional language models with vectors, symbolic models with composition, but little in the way of distributional vector models with composition.

[1] Early examples are given by Aristotle, such as (*De Interpretatione, Ch IV):*

> *The word 'human' has meaning, but does not constitute a proposition, either positive or negative. It is only when other words are added that the whole will form an affirmation or denial.*

[2] Familiarity with tensor products is assumed throughout this paper. Readers familiar with the linear algebra of vectors but not the multilinear algebra of tensors are encouraged to read the introduction to tensors in Widdows et al. (2021, §5).

Proceedings of the 2021 Workshop on Semantic Spaces at the Intersection of NLP, Physics, and Cognitive Science, pages 76–86
June 16th, 2021. ©2021 Association for Computational Linguistics

2 Explicit Composition in Semantic Vector Models

The most standard operations used for composition and comparison in vector model information retrieval systems have, for many decades, been the vector sum and cosine similarity (see Salton et al. 1975), and for an introduction, Widdows (2004, Ch 5)). Cosine similarity is normally defined and calculated in terms of the natural scalar product induced by the coordinate system, i.e.,

$$\cos(a, b) = \frac{a \cdot b}{\sqrt{(a \cdot a)(b \cdot b)}}.$$

While the scalar product is a linear operator because $\lambda a \cdot \mu b = \lambda\mu(a \cdot b)$, cosine similarity is deliberately designed so that $cos(\lambda a, \mu b) = \cos(a, b)$, so that normalizing or otherwise reweighting vectors does not affect their similarity, which depends only on the angle between them.

More sophisticated vector composition in AI was introduced with cognitive models and connectionism. The work of Smolensky (1990) has already been mentioned, and the holographic reduced representations of Plate (2003) is another widely-cited influence (discussed again below as part of Vector Symbolic Architectures). While Smolensky's work is often considered to be AI rather than NLP, the application to language was a key consideration:

> Any connectionist model of natural language processing must cope with the questions of how linguistic structures are represented in connectionist models. (Smolensky 1990, §1.2)

The use of more varied mathematical operators to model natural language operations with vectors accelerated considerably during the first decade of this century. In information retrieval, van Rijsbergen (2004) explored conditional implication and Widdows (2003) developed the use of orthogonal projection for negation in vector models.

Motivated partly by formal similarities with quantum theory, Aerts & Czachor (2004) proposed modelling a sentence $w_1, \ldots, w_n$ with a tensor product $w_1 \otimes \ldots \otimes w_n$ in the Fock space $\bigoplus_{k=1}^{\infty} V^{\otimes^k}$. The authors noted that comparing the spaces $V^{\otimes^k}$ and $V^{\otimes^j}$ when $k \neq j$ is a difficulty shared by other frameworks, and of course the prospect of summing to $k = \infty$ is a mathematical notation that motivates the search for a tractable implementation proposal.

By the middle of the decade, Clark & Pulman (2007) and Widdows (2008) proposed and experimented with the use of tensor products for semantic composition, and the parallelogram rule for relation extraction. Such methods were used to obtain strong empirical results for (intransitive) verb-noun composition by Mitchell & Lapata (2008) and for adjective-noun composition by Baroni & Zamparelli (2010). One culmination of this line of research is the survey by Baroni et al. (2014), who also addressed the problem of comparing tensors $V^{\otimes^k}$ and $V^{\otimes^j}$ when $k \neq j$. For example, if (as in Baroni & Zamparelli 2010)), nouns are represented by vectors and adjectives are represented by matrices, then the space of matrices is isomorphic to $V \otimes V$ which is not naturally comparable to V, and the authors note (Baroni & Zamparelli 2010, §3.4):

> As a result, Rome and Roman, Italy and Italian cannot be declared similar, which is counter-intuitive. Even more counter-intuitively, Roman used as an adjective would not be comparable to Roman used as a noun. We think that the best way to solve such apparent paradoxes is to look, on a case-by-case basis, at the linguistic structures involved, and to exploit them to develop specific solutions.

Another approach is to use a full syntactic parse of a sentence to construct vectors in a sentence space S from nouns and verbs as constituents in their respective spaces. This features prominently in the model of Coecke et al. (2010), which has become affectionately known as DisCoCat, from '*Dis*tributional *Co*mpositional *Cat*egorical'. The mathematics is at the same time sophisticated but intuitive: its formal structure relies on pregroup grammars and morphisms between compact closed categories, and intuitively, the information from semantic word vectors flows through a network of tensor products that parallels the syntactic bindings and produces a single vector in the S space.

Various papers have demonstrated empirical successes for the DisCoCat and related models. Grefenstette & Sadrzadeh (2011) were among the first, showing comparable and sometimes improved results with those of Mitchell & Lapata (2008). By 2014, several tensor composition operations were compared by Milajevs et al. (2014), and Sadrzadeh et al. (2018) showed that word, phrase, and sentence entailment can be modelled using vectors and density matrices. (The use of density matrices to model probability distributions for entailment was pioneered partly by van Rijsbergen (2004, p. 80) and will be discussed further in Section 4.) Further mathematical tools used in DisCoCat research include *copying* a vector $v \in V$ into a tensor in $V \otimes V$ where the coordinates of v become the diagonal entries in the matrix representation of the corresponding tensor, and *uncopying* which takes the diagonal elements of a matrix representing a tensor in $V \otimes V$ and uses these as the coordinates of a vector. With these additional operators, the tensor algebra becomes more explicitly a *Frobenius algebra*.[3] These are used in DisCoCat models by Sadrzadeh et al. (2014*a*,*b*) to represent relative and then possessive pronoun attachments (for example, representing the affect of the phrase "that chased the mouse" as part of the phrase "The cat that chased the mouse"). The method involves detailed tracking of syntactic types and their bindings, and certainly follows the suggestion from Baroni & Zamparelli (2010) to look at linguistic structures on a case-by-case basis.

There are practical concerns with tensor product formalisms. The lack of any isomorphism between $V^{\otimes^k}$ and $V^{\otimes^j}$ when $k \neq j$ and $\dim V > 1$ has already been noted, along with the difficulty this poses for comparing elements of each for similarity. Also, there is an obvious computational scaling problem: if V has dimension n, then $V^{\otimes^k}$ has dimension n^k, which leads to exponential memory consumption with classical memory registers. Taking the example of relative pronouns in the DisCoCat models of Sadrzadeh et al. (2014*a*) — these are represented as rank-4 tensors in spaces such as $N \otimes S \otimes N \otimes N$ and variants thereof, and if the basic noun space N and sentence space S have dimension 300 (a relatively standard number used e.g., by FastText vectors) then the relative pronouns would have dimension 8.1 billion. If this was represented densely and the coordinates are 4-byte floating point numbers, then just representing one pronoun would require over 30GB of memory, which is intractable even by today's cloud computing standards.

The development of Vector Symbolic Architectures (VSAs) (Gayler 2004) was partly motivated by these concerns. VSAs grew from the holographic reduced representations of Plate (2003): notable works in this intersection of cognitive science and artificial intelligence include those of Eliasmith (2013) and Kanerva (2009). At its core, a VSA is a vector space with an addition operator and a scalar product for computing similarity, along with a multiplication or *binding* operator (sometimes written as $*$, or $\otimes$ like the tensor product) which takes a product of two vectors and returns a new vector that it typically *not* similar to either of its inputs, so that $(a * b) \cdot a$ is small, but which is 'approximately reversible' — so there is an approximate inverse operator $\oslash$ where $(a * b) \oslash b$ is close to a.[4] The term 'binding' was used partly for continuity with the role-filler binding of Smolensky (1990).[5]

The VSA community has tended to avoid the full tensor product, for the reasons given above. In order to be directly comparable, it is desirable that $a * b$ should be a vector in the space V. Plate (2003) thoroughly explored the use of *circular correlation* and *circular convolution* for these operations, which involves summing the elements of the outer product matrix along diagonals. This works as a method to map $V \otimes V$ back to V, though the mapping is of course basis dependent. Partly to optimize the binding operation to $O(n)$ time, Plate (2003, Ch 5) introduces *circular vectors*, whose coordinates are unit complex numbers. There is some stretching of terminology here, because the circle group $U(1)$ of unit complex numbers is not, strictly speaking, a vector space. Circular vectors are added by adding their rectangular coordinates in a linear fashion, and then normalizing back to the unit circle by discarding the magnitude, which Plate notes is not an associative operation. Kanerva (2009) departs perhaps even further from the vector space mainstream, using binary-valued vectors throughout, with binding implemented as pairwise exclusive OR (XOR).

VSA binding operations have been used for composition of semantic vector representations, both during the training process to generate composite vector representations of term or character n-grams,

[3] Named after Georg Frobenius (1849–1917), a group theorist who contributed particularly to group representation theory. See Kartsaklis (2015) for a thorough presentation.

[4] This also explains why it is tempting to reuse or abuse the tensor product notation and use the symbol $\otimes$ for binding and $\oslash$ for the inverse or release operator, as in Widdows & Cohen (2015).

[5] The requirement that the binding $a*b$ be dissimilar to both a and b makes the Frobenius uncopying of Kartsaklis (2015) operator unsuitable for a VSA, because the coordinates of $v*w$ are the products of the corresponding coordinates in v and w, which typically makes the scalar product with either factor quite large. This is however a rather shallow observation, and the relationship between VSAs and Frobenius algebras may be a fruitful topic to investigate more thoroughly.

or semantic units such as predicate-argument pairs or syntactic dependencies, that are then further assembled to construct representations of larger units (Jones & Mewhort 2007, Kachergis et al. 2011, Cohen & Widdows 2017, Paullada et al. 2020); and to compose larger units from pretrained semantic vectors for downstream machine learning tasks (Fishbein & Eliasmith 2008, Mower et al. 2016). However, a concern with several of the standard VSA binding operators for the representation of sequences in particular is that they are commutative in nature: $x * y = y * x$. To address this concern, permutations of vector coordinates have been applied across a range of VSA implementations to break the commutative property of the binding operator, for example by permuting the second vector in sequence such that $\overrightarrow{wet} * \prod(\overrightarrow{fish})$ and $\overrightarrow{fish} * \prod(\overrightarrow{wet})$ result in different vectors (Kanerva 2009, Plate 2003, p. 121).

Thanks to their general nature and computational simplicity, permutations have been used for several other encoding and composition experiments. The use of permutations to encode positional information into word vector representations was introduced by Sahlgren et al. (2008). In this work a permutation (coordinate shuffling) operator was used to rearrange vector components during the course of training, with a different random permutation assigned to each sliding window position such that a context vector would be encoded differently depending upon its position relative to a focus term of interest. A subsequent evaluation of this method showed advantages in performance over the BEAGLE model (Jones & Mewhort 2007), which uses circular convolutions to compose representations of word n-grams, on a range of intrinsic evaluation tasks — however these advantages were primarily attributable to the permutation-based approach's ability to scale to a larger training corpus (Recchia et al. 2015). Random permutations have also been used to encode semantic relations (Cohen et al. 2009) and syntactic dependencies (Basile et al. 2011) into distributional models.

In high-dimensional space, the application of two different random permutations to the same vector has a high probability of producing vectors that are close-to-orthogonal to one another (Sahlgren et al. 2008). A more recent development has involved deliberately constructing 'graded' permutations by randomly permuting part of a parent permutation (Cohen & Widdows 2018). When this process is repeated iteratively, it results in a set of permutations that when applied to the same vector will produce a result with similarity to the parent vector that decreases in an ordinal fashion. This permits the encoding of proximity rather than position, in such a way that words in proximal positions within a sliding window will be similarly but not identically encoded. The resulting proximity-based encodings have shown advantages over comparable encodings that are based on absolute position (at word and sentence level) or are position-agnostic (at word and character level) across a range of evaluations (Cohen & Widdows 2018, Schubert et al. 2020, Kelly et al. 2020).

Note that coordinate permutations are all Euclidean transformations: odd permutations are reflections, and even permutations are rotations. Thus all permutation operations are also linear.

This survey of explicit composition in semantic vector models is not exhaustive, but gives some idea of the range of linear and nonlinear operations.

3 Compositional Semantics in Neural Networks

During the past decade, many of the most successful and well-known advances in semantic vector representations have been developed using neural networks.[6] In general, such networks are trained with some objective function designed to maximize the probability of predicting a given word, sentence, or group of characters in a given context. Various related results such as those of Scarselli & Tsoi (1998) are known to demonstrate that, given enough computational resources and training data, neural networks can be used to approximate any example from large classes of functions. If these target functions are nonlinear, this cannot be done with a network of entirely linear operations, because the composition of two linear maps is another linear map — "The hidden units should be nonlinear because multiple layers of linear units can only produce linear functions." (Wichert 2020, §13.5). Thus, part of the appeal of neural networks is that they are *not* bound by linearity: though often at considerable computational cost.

The skip gram with negative sampling method was introduced by Mikolov et al. (2013), imple-

[6] An introduction to this huge topic is beyond the scope of this paper. Unfamiliar readers are encouraged to start with a general survey such as that of Géron (2019), Chapter 16 being particularly relevant to the discussion here.

mentations including the word2vec[7] package from Google and the FastText package from Facebook.[8] The objective function is analyzed more thoroughly by Goldberg & Levy (2014), and takes the form:

$$\sum_{(w,c)\in D} \log \sigma(\vec{w} \cdot \vec{c}) + \sum_{(w,\neg c)\in D'} \log \sigma(-\vec{w} \cdot \vec{\neg c})$$

Here w is a word, c is a context feature (such as a nearby word), D represents observed term/context pairs in the document collection, D' represents randomly drawn counterexamples, and $\vec{w}$ and $\vec{c}$ are word and context vectors (input and output weights of the network, respectively). σ is the sigmoid function, $\frac{1}{1+e^{-x}}$. The mathematical structure here is in the family of logistic and softmax functions — the interaction between the word and context vectors involves exponential / logarithmic concepts, not just linear operations.

There have been efforts to incorporate syntactic information explicitly in the training process of neural network models. In the specific case of adjectives, Maillard & Clark (2015) use the skip gram technique to create matrices for adjectives following the pattern of Baroni & Zamparelli (2010) discussed in Section 2. The most recent culmination of this work is its adaptation to cover a much more comprehensive collection of categorial types by Wijnholds et al. (2020). Another early example comes from Socher et al. 2012, who train a Recursive Neural Network where each node in a syntactic parse tree becomes represented by a matrix that operates on a pair of inputs. Research on tree-structured LSTMs (see inter alia Tai et al. 2015, Maillard et al. 2019) leverages syntactic parse trees in the input and composes its hidden state using an arbitrary number of child nodes, as represented in the syntax tree. Syntax-BERT (Bai et al. 2021) uses syntactic parses to generate masks that reflect different aspects of tree structure (parent, child, sibling). KERMIT (Zanzotto et al. 2020) uses compositional structure explicitly by embedding syntactic subtrees in the representation space. In both cases, the use of explicit compositional syntactic structure leads to a boost in performance on various semantic tasks.

In KERMIT, the embedding of trees and (recursively) their subtrees follows a well-developed line of research on representing discrete structures as vectors, in particular combining circular convolution and permutations to introduce *shuffled circular convolution* (Ferrone & Zanzotto 2014). Even when combined in a recursive sum over constituents called a Distributed Tree Kernel operation, this is still a sum of linear inputs, so this form of composition is still linear throughout. In such methods, the result may be a collection of related linear operators representing explicit syntactic bindings, but the training method is typically not linear due to the activation functions.

What these neural net methods and the models described in the previous section have in common is that they encode some *explicit* compositional structure: a weighted sum of word or character n-grams, a role / value binding, or a relationship in a grammatical parse tree. This raises the question: can neural language models go beyond the bag-of-words drawbacks and encode more order-dependent language structures without using traditional logical compositional machinery?

A recent and comprehensive survey of this topic is provided by Hupkes et al. (2020). This work provides a valuable survey of the field to date, and conducts experiments with compositional behavior on artificial datasets designed to demonstrate various aspects of compositionality, such as productivity (can larger unseen sequences be produced?) and substitutivity (are outputs the same when synonymous tokens are switched?). This systematic approach to breaking compositionality into many tasks is a useful guide in itself.

Since then, attention-based networks were developed and have come to the forefront of the field (Vaswani et al. 2017). The attention mechanism is designed to learn when pairs of inputs depend crucially on one another, a capability that has demonstrably improved machine translation by making sure that the translated output represents all of the given input even when their word-orders do not correspond exactly. The 'scaled dot-product attention' used by Vaswani et al. (2017) for computing the attention between a pair of constituents uses softmax normalization, another nonlinear operation.

The use of attention mechanisms has led to rapid advances in the field, including the contextualized BERT (Devlin et al. 2018) and ELMo (Peters et al. 2018) models. For example, the ELMo model reports good results on traditional NLP tasks including question answering, coreference resolution, semantic role labelling, and part-of-speech tagging,

[7] `https://pypi.org/project/word2vec/`

[8] `https://fasttext.cc/`

and the authors speculate that this success is due to the model's different neural-network layers implicitly representing several different kinds of linguistic structure. This idea is further investigated by Hewitt & Manning (2019) and Jawahar et al. (2019), who probe BERT and ELMo models to find evidence that syntactic structure is implicitly encoded in their vector representations. The survey and experiments of Hupkes et al. (2020) evaluate three such neural networks on a range of tasks related to composition, concluding that each network has strengths and weaknesses, that the results are a stepping stone rather than an endpoint, and that developing consensus around how such tasks should be designed, tested and shared is a crucial task in itself.

At the time of writing, such systems are contributors to answering a very open research question: do neural networks need extra linguistic information in their input to properly work with language, or can they actually *recover* such information as a byproduct of training on raw text input? For example, a DisCoCat model requires parsed sentences as input — so if another system performed as well without requiring grammatical sentences as input and the support of a distinct parsing component in the implementation pipeline, that would be preferable in most production applications. (Running a parser is a requirement than today can often be satisfied, albeit with an implementational and computational cost. Requiring users to type only grammatical input is a requirement that cannot typically be met at all.) At the same time, does performance on the current NLP tasks used for evaluation directly indicate semantic composition at play? If the performance of a model without linguistic information in the input is up to par, would the internal operations of such an implicit model be largely inscrutable, or can we describe the way meaningful units are composed into larger meaningful structures explicitly?

Tensor networks are one of the possible mathematical answers to this question, and continue to build upon Smolensky's introduction of tensors to AI. For example McCoy et al. (2020) present evidence that the sequence-composition effects of Recurrent Neural Networks (RNNs) can be approximated by Tensor Product Decomposition Networks, at least in cases where using this structure provides measurable benefits over bag-of-words models. It has also been shown that Tensor Product Networks can be used to construct an attention mechanism from which grammatical structure can be recovered by unbinding role-filler tensor compositions (Huang et al. 2019).

While there are many more networks that could be examined in a survey like this, those described in this section illustrate that neural networks have been used to improve results with many NLP tasks, and the training of such networks often crucially depends on nonlinear operations on vectors. Furthermore, while tensor networks have been developed as a proposed family of techniques for understanding and exploiting compositional structures more explicitly, in some of the most state-of-the-art models, relationships between such operations to more traditional semantic composition are often absent or at least not well-understood.

4 Operators from Quantum Models

Mathematical correspondences between vector models for semantics and quantum theory have been recognized for some years (van Rijsbergen 2004), and are surveyed by Widdows et al. (2021). The advent of practical quantum computing makes these correspondences especially interesting, and constructs from quantum theory have been used increasingly deliberately in NLP. In quantum computing, tensor products no longer incur quadratic costs: instead, the tensor product $A \otimes B$ is the natural mathematical representation of the physical state that arises when systems in states A and B are allowed to interact. Heightened interest in quantum computing and quantum structures in general has led to specific semantic contributions already.

Mathematically, there is a historic relationship between linearity and quantum mechanics: the principle of superposition guarantees that for any state A, the vector cA corresponds to the same physical state for any complex number c (Dirac 1930, §5).[9] Hence the question of whether a compositional operator is linear or not is particularly relevant when we consider the practicality of implementation on a quantum computer.[10]

Many developments have followed from the Dis-

[9] This itself could lead to a mathematical discussion — the magnitude of a state vector in quantum mechanics is ignored, just like cosine similarity ignores the scale factors of the scalar product, and its resilience to scale factors makes the cosine similarity technically *not* a linear operator.

[10] The dependence of quantum computing on linearity should not go unquestioned — for example, the use of quantum harmonic oscillators rather than qubits has been proposed as a way to incorporate nonlinearity into quantum hardware by Goto (2016).

CoCat framework, whose mathematical structure is closely related to quantum mechanics through category theory (Coecke et al. 2017). As of 2021, the tensor network components of two DisCoCat models have even been implemented successfully on a quantum computer (Lorenz et al. 2021), and there are proposals for how to implement the syntactic parse on quantum hardware as well (Wiebe et al. 2019, Bausch et al. 2021). Of particular semantic and mathematical interest, topics such as hyponymy (Bankova et al. 2019) and negation (Lewis 2020) have been investigated, using density matrices and positive operator-valued measures, which are mathematical generalizations of state vectors and projection operators that enable the theory to describe systems that are not in 'pure' states. Density matrices have also been used to model sentence entailment (Sadrzadeh et al. 2018) and recently lexical ambiguity (Meyer & Lewis 2020).

A comprehensive presentation of the use of density matrices to model joint probability distributions is given by Bradley (2020). This work deliberately takes a quantum probability framework and applies it to language modelling, by way of the lattice structures of Formal Concept Analysis (Ganter & Wille 1999). This work uses the *partial trace* of density operators (which are tensors in $V \otimes V$) to project tensors in $V \otimes V$ to vectors in V. This is analogous to summing the rows or columns of a two-variable joint distribution to get a single-variable marginal distribution. This captures interference and overlap between the initial concepts, and in a framework such as DisCoCat, this might be used to model transitive verb-noun composition (as in Grefenstette & Sadrzadeh 2011, Sadrzadeh et al. 2018, and others).

Another mathematical development is the quantum Procrustes alignment method of Lloyd et al. (2020), where Procrustes alignment refers to the challenge of mapping one vector space into another preserving relationships as closely as possible. Procrustes techniques have been used to align multilingual FastText word vectors (Joulin et al. 2018), and it is possible that one day these methods may be combined to give faster and more noise-tolerant multilingual concept alignment.

This again is not a complete survey, but we hope it demonstrates that the interplay between quantum theory, semantic vector composition, and practical implementation has much to offer, and that work in this area is accelerating.

5 Summary, Conclusion, and Future Work

This paper has surveyed vector composition techniques used for aspects of semantic composition in explicit linguistic models, neural networks, and quantum models, while acknowledging that these areas overlap. The operations considered are gathered and summarized in Table 1.

Some of the most successful neural network models to date have used operations that are nonlinear and implicit. Though models such as BERT and ELMo have performed exceptionally well on several benchmark tasks, they are famously difficult to explain and computationally expensive. Therefore, scientific researchers and commercial user-facing enterprises have good reason to be impressed, but still to seek alternatives that are clearer and cheaper. At the same time, progress in quantum computing raises the possibility that the practical cost of different mathematical operations may be considerably revised over the coming year. For example, if the expense of tensor products becomes linear rather than quadratic, tensor networks may find a position at the forefront of 'neural quantum computing'.

In addition, there is emerging evidence that such models can be augmented by approaches that draw on structured semantic knowledge (Michalopoulos et al. 2020, Colon-Hernandez et al. 2021), suggesting the combination of implicit and explicit approaches to semantic composition as a fruitful area for future methodological research. We hope that this approach of surveying and comparing the semantic, mathematical and computational elements of various vector operations will serve as a guide to territory yet to be explored at the intersection of compositional operators and vector representations of language.

6 Acknowledgements

The authors would like to thank the anonymous reviewers for helping to improve the coverage of this survey, and the conference organizers for allowing extra pages to accommodate these additions.

References

Aerts, D. & Czachor, M. (2004), 'Quantum aspects of semantic analysis and symbolic artificial intelligence', *J. Phys. A: Math. Gen.* **37**, L123–L132.

Bai, J., Wang, Y., Chen, Y., Yang, Y., Bai, J., Yu, J. & Tong, Y. (2021), 'Syntax-BERT: Improving

Table 1: Survey Summary of Mathematical Methods Used for Semantic Vector Composition

Mathematical Method	**Use Case**	**Inputs**	**Outputs**	**Explicitness**	**Linearity**	**Described By**
Vector sum	Document retrieval and many others	Word vectors	Document vectors	Explicit	Linear	Widespread including Salton et al. (1975)
Scalar product	Similarity scoring	Any vector	Any vector	Explicit	Linear	Various
Cosine similarity	Similarity scoring	Any vector	Any vector	Explicit	Nonlinear — deliberately ignores magnitude	Various
Tensor product	Role-filler binding	Variable name	Variable value	Explicit	Linear	Smolensky (1990)
Circular vector sum	Holographic reduced representations	Circular vectors	Circular vectors	Explicit	Nonlinear, and non-associative	Plate (2003)
(Stalnaker) Conditional	Implication	Vector / subspace representing propositions	Subspace representing truth conditions	Explicit	Linear (though for subspaces it doesn't matter)	van Rijsbergen (2004, Ch 5)
Orthogonal projection	Negation	Vector or subspace	Vector	Explicit	Linear	Widdows (2003)
Sum of subspaces	Disjunction	Vectors or subspaces	Subspace	Explicit	Linear	Widdows (2003)
Parallelogram rule	Proportional analogy	Three vectors	Fourth vector	Explicit	Linear	Various incl. Widdows (2008), Mikolov et al. (2013)
Tensor product	Word vector composition	Word vector	Sentence-fragment tensor	Explicit	Linear	Various since Aerts & Czachor (2004), Clark & Pulman (2007)
Tensor and monoidal product	Parallel semantic, syntactic composition	(vector, syntactic type) pairs	Sentence vectors	Explicit	Linear	Various since Coecke et al. (2010)
Matrix multiplication	Adjective / noun composition	Matrix and vector	Vector	Explicit	Linear	Baroni & Zamparelli (2010)
Circular convolution	Vector binding	VSA vector	VSA vector	Explicit	Sometimes	Plate (2003), options in Widdows & Cohen (2015)
Binary XOR	Binary vector binding	VSA vector	VSA vector	Explicit	Binary vectors warrant more discussion!	Kanerva (2009)
Permutation of coordinates	Non-additive composition	Vector	Vector	Explicit, though often random	Linear (because rotation or reflection)	Sahlgren et al. (2008) and various
Skipgram objective	Vector interation in training	Word and context vector	Update to both	Explicit though internal	Nonlinear	Mikolov et al. (2013)
$\tanh$, Sigmoid, ReLU, Softmax, etc.	Activation functions in neural networks	Input weights	Output weights	Typically implicit	Nonlinear	Many including Géron (2019, Ch 10)
Scaled dot-product attention	Learning pairwise dependence	Vectors	Updated vectors	Typically internal	Nonlinear	Vaswani et al. (2017)
Distributed tree kernel / shuffled circular convolution	Embedding syntactic tree in vector space	Parse tree	Sentence vector	Explicit	Linear	(Ferrone & Zanzotto 2014)
Density matrices, POVMs	More general distributions over vector spaces, e.g., representing categories, implication	Several, e.g., superpositions of pairs of vectors	Projected vectors and / or probabilities	Often explicit	Linear	van Rijsbergen (2004), Sadrzadeh et al. (2018), Lewis (2020), Bradley (2020)
Procrustes alignment	Aligning vector models U and V	Pairs of source, target vectors	Linear mapping from U to V	Explicit	Linear	Bojanowski et al. (2017), Lloyd et al. (2020)

pre-trained transformers with syntax trees', *arXiv preprint arXiv:2103.04350* .

Bankova, D., Coecke, B., Lewis, M. & Marsden, D. (2019), 'Graded hyponymy for compositional distributional semantics', *Journal of Language Modelling* **6**(2), 225–260.

Baroni, M., Bernardi, R., Zamparelli, R. et al. (2014), 'Frege in space: A program for compositional distributional semantics', *Linguistic Issues in language technology* **9**(6), 5–110.

Baroni, M. & Zamparelli, R. (2010), Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space, *in* 'Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)'.

Basile, P., Caputo, A. & Semeraro, G. (2011), Encoding syntactic dependencies by vector permutation, *in* 'Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics', pp. 43–51.

Bausch, J., Subramanian, S. & Piddock, S. (2021), 'A quantum search decoder for natural language processing', *Quantum Machine Intelligence* **3**(1), 1–24.

Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. (2017), 'Enriching word vectors with subword information', *Transactions of the Association for Computational Linguistics* **5**, 135–146.

Boole, G. (1854), *An Investigation of the Laws of Thought*, Macmillan. Dover edition, 1958.

Bradley, T.-D. (2020), 'At the interface of algebra and statistics', *PhD dissertation, arXiv preprint arXiv:2004.05631* .

Clark, S. & Pulman, S. (2007), Combining symbolic and distributional models of meaning., *in* 'AAAI Spring Symposium: Quantum Interaction', pp. 52–55.

Coecke, B., Genovese, F., Gogioso, S., Marsden, D. & Piedeleu, R. (2017), 'Uniqueness of composition in quantum theory and linguistics', *14th International Conference on Quantum Physics and Logic (QPL)* .

Coecke, B., Sadrzadeh, M. & Clark, S. (2010), 'Mathematical foundations for a compositional distributional model of meaning', *CoRR* **abs/1003.4394**.

Cohen, T., Schvaneveldt, R. W. & Rindflesch, T. C. (2009), Predication-based semantic indexing: Permutations as a means to encode predications in semantic space, *in* 'AMIA Annual Symposium Proceedings', Vol. 2009, American Medical Informatics Association, p. 114.

Cohen, T. & Widdows, D. (2017), 'Embedding of semantic predications', *Journal of biomedical informatics* **68**, 150–166.

Cohen, T. & Widdows, D. (2018), Bringing order to neural word embeddings with embeddings augmented by random permutations (earp), *in* 'Proceedings of the 22nd Conference on Computational Natural Language Learning', pp. 465–475.

Colon-Hernandez, P., Havasi, C., Alonso, J., Huggins, M. & Breazeal, C. (2021), 'Combining pre-trained language models and structured knowledge', *arXiv preprint arXiv:2101.12294* .

Deerwester, S., Dumais, S., Furnas, G., Landauer, T. & Harshman, R. (1990), 'Indexing by latent semantic analysis', *Journal of the American Society for Information Science* **41(6)**, 391–407.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2018), 'BERT: Pre-training of deep bidirectional transformers for language understanding', *arXiv preprint arXiv:1810.04805* .

Dirac, P. (1930), *The Principles of Quantum Mechanics*, 4th edition, 1958, reprinted 1982 edn, Clarendon Press, Oxford.

Eliasmith, C. (2013), *How to Build a Brian: A Neural Architecture for Biological Cognition*, Oxford University Press.

Ferrone, L. & Zanzotto, F. (2014), Towards syntax-aware compositional distributional semantic models, *in* 'COLING 2014, 25th International Conference on Computational Linguistics'.

Fishbein, J. M. & Eliasmith, C. (2008), Integrating structure and meaning: A new method for encoding structure for text classification, *in* 'European Conference on Information Retrieval', Springer, pp. 514–521.

Frege, G. (1884), *The Foundations of Arithmetic (1884)*, 1974 (translated by J. L. Austin) edn, Blackwell.

Gamut, L. (1991), *Logic, Language, and Meaning*, University of Chicago Press.

Ganter, B. & Wille, R. (1999), *Formal Concept Analysis: Mathematical Foundations*, Springer.

Gayler, R. W. (2004), Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience, *in* 'In Peter Slezak (Ed.), ICCS/ASCS International Conference on Cognitive Science', Sydney, Australia. University of New South Wales., pp. 133–138.

Géron, A. (2019), *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media.

Goldberg, Y. & Levy, O. (2014), 'Word2Vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method', *arXiv preprint arXiv:1402.3722* .

Goto, H. (2016), 'Bifurcation-based adiabatic quantum computation with a nonlinear oscillator network', *Scientific reports* **6**(1), 1–8.

Grassmann, H. (1862), *Extension Theory*, History of Mathematics Sources, American Mathematical Society, London Mathematical Society. Translated by Lloyd C. Kannenberg (2000).

Grefenstette, E. & Sadrzadeh, M. (2011), 'Experimental support for a categorical compositional distributional model of meaning', *EMNLP* .

Hamilton, S. W. R. (1847), 'On quaternions', *Proc. Royal Irish Acad.* **3**, 1–16.

Hewitt, J. & Manning, C. D. (2019), A structural probe for finding syntax in word representations, *in* 'ACL', pp. 4129–4138.

Huang, Q., Deng, L., Wu, D., Liu, C. & He, X. (2019), Attentive tensor product learning, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 33, pp. 1344–1351.

Hupkes, D., Dankers, V., Mul, M. & Bruni, E. (2020), 'Compositionality decomposed: how do neural networks generalise?', *Journal of Artificial Intelligence Research* **67**, 757–795.

Jawahar, G., Sagot, B. & Seddah, D. (2019), What does bert learn about the structure of language?, *in* 'ACL 2019-57th Annual Meeting of the Association for Computational Linguistics'.

Jones, M. N. & Mewhort, D. J. K. (2007), 'Representing word meaning and order information in a composite holographic lexicon', *Psychological Review* **114**, 1–37.

Joulin, A., Bojanowski, P., Mikolov, T., Jégou, H. & Grave, E. (2018), Loss in translation: Learning bilingual word mapping with a retrieval criterion, *in* 'EMNLP'.

Kachergis, G., Cox, G. E. & Jones, M. N. (2011), Orbeagle: integrating orthography into a holographic model of the lexicon, *in* 'International conference on artificial neural networks', Springer, pp. 307–314.

Kanerva, P. (2009), 'Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors', *Cognitive Computation* **1**(2), 139–159.

Kartsaklis, D. (2015), 'Compositional distributional semantics with compact closed categories and frobenius algebras', *PhD thesis, Wolfson College Oxford. arXiv preprint arXiv:1505.00138* .

Kelly, M. A., Xu, Y., Calvillo, J. & Reitter, D. (2020), 'Which sentence embeddings and which layers encode syntactic structure?'.

Lewis, M. (2020), 'Towards logical negation for compositional distributional semantics', *arXiv preprint arXiv:2005.04929* .

Lloyd, S., Bosch, S., De Palma, G., Kiani, B., Liu, Z.-W., Marvian, M., Rebentrost, P. & Arvidsson-Shukur, D. M. (2020), 'Quantum polar decomposition algorithm', *arXiv preprint arXiv:2006.00841* .

Lorenz, R., Pearson, A., Meichanetzidis, K., Kartsaklis, D. & Coecke, B. (2021), 'QNLP in practice: Running compositional models of meaning on a quantum computer'.

Maillard, J. & Clark, S. (2015), Learning adjective meanings with a tensor-based skip-gram model, *in* 'Proceedings of the Nineteenth Conference on Computational Natural Language Learning', pp. 327–331.

Maillard, J., Clark, S. & Yogatama, D. (2019), 'Jointly learning sentence embeddings and syntax with unsupervised tree-lstms', *Natural Language Engineering* **25**(4), 433–449.

McCoy, R. T., Linzen, T., Dunbar, E. & Smolensky, P. (2020), 'Tensor product decomposition networks: Uncovering representations of structure learned by neural networks', *Proceedings of the Society for Computation in Linguistics* **3**(1), 474–475.

Meyer, F. & Lewis, M. (2020), 'Modelling lexical ambiguity with density matrices', *arXiv preprint arXiv:2010.05670* .

Michalopoulos, G., Wang, Y., Kaka, H., Chen, H. & Wong, A. (2020), 'Umlsbert: Clinical domain knowledge augmentation of contextual embeddings using the unified medical language system metathesaurus', *arXiv preprint arXiv:2010.10391* .

Mikolov, T., Chen, K., Corrado, G. & Dean, J. (2013), 'Efficient estimation of word representations in vector space', *arXiv preprint arXiv:1301.3781* .

Milajevs, D., Kartsaklis, D., Sadrzadeh, M. & Purver, M. (2014), Evaluating neural word representations in tensor-based compositional settings, *in* 'EMNLP', pp. 708–719.

Mitchell, J. & Lapata, M. (2008), Vector-based models of semantic composition., *in* 'ACL', pp. 236–244.

Mower, J., Subramanian, D., Shang, N. & Cohen, T. (2016), Classification-by-analogy: using vector representations of implicit relationships to identify plausibly causal drug/side-effect relationships, *in* 'AMIA annual symposium proceedings', Vol. 2016, American Medical Informatics Association, p. 1940.

Partee, B. H., ter Meulen, A. & Wall, R. E. (1993), *Mathematical Methods in Linguistics*, Kluwer.

Paullada, A., Percha, B. & Cohn, T. (2020), Improving biomedical analogical retrieval with embedding of structural dependencies, *in* 'Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing', pp. 38–48.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. & Zettlemoyer, L. (2018), 'Deep contextualized word representations', *arXiv preprint arXiv:1802.05365* .

Plate, T. A. (2003), *Holographic Reduced Representations: Distributed Representation for Cognitive Structures*, CSLI Publications.

Recchia, G., Sahlgren, M., Kanerva, P. & Jones, M. N. (2015), 'Encoding sequential information in semantic space models: comparing holographic reduced representation and random permutation.', *Computational intelligence and neuroscience* .

Sadrzadeh, M., Clark, S. & Coecke, B. (2014*a*), 'The frobenius anatomy of word meanings ii: possessive relative pronouns', *Journal of Logic and Computation* **26**(2), 785–815.

Sadrzadeh, M., Clark, S. & Coecke, B. (2014*b*), 'The frobenius anatomy of word meanings ii: possessive relative pronouns', *Journal of Logic and Computation* **26**(2), 785–815.

Sadrzadeh, M., Kartsaklis, D. & Balkır, E. (2018), 'Sentence entailment in compositional distributional semantics', *Annals of Mathematics and Artificial Intelligence* **82**(4), 189–218.

Sahlgren, M., Holst, A. & Kanerva, P. (2008), Permutations as a means to encode order in word space., *in* 'Proceedings of the 30th Annual Meeting of the Cognitive Science Society (CogSci'08), July 23-26, Washington D.C., USA.'.

Salton, G., Wong, A. & Yang, C.-S. (1975), 'A vector space model for automatic indexing', *Communications of the ACM* **18**(11), 613–620.

Scarselli, F. & Tsoi, A. C. (1998), 'Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results', *Neural networks* **11**(1), 15–37.

Schubert, T. M., Cohen, T. & Fischer-Baum, S. (2020), 'Reading the written language environment: Learning orthographic structure from statistical regularities', *Journal of Memory and Language* **114**, 104148.

Smolensky, P. (1990), 'Tensor product variable binding and the representation of symbolic structures in connectionist systems', *Artificial intelligence* **46**(1), 159–216.

Socher, R., Huval, B., Manning, C. D. & Ng, A. Y. (2012), Semantic compositionality through recursive matrix-vector spaces, *in* 'EMNLP', pp. 1201–1211.

Tai, K. S., Socher, R. & Manning, C. D. (2015), 'Improved semantic representations from tree-structured long short-term memory networks', *arXiv preprint arXiv:1503.00075* .

van Rijsbergen, K. (2004), *The Geometry of Information Retrieval*, Cambridge University Press.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017), Attention is all you need, *in* 'Advances in neural information processing systems', pp. 5998–6008.

Wichert, A. (2020), *Principles of quantum artificial intelligence: Quantum Problem Solving and Machine Learning (Second Edition)*, World scientific.

Widdows, D. (2003), Orthogonal negation in vector spaces for modelling word-meanings and document retrieval, *in* 'ACL 2003', Sapporo, Japan.

Widdows, D. (2004), *Geometry and Meaning*, CSLI Publications.

Widdows, D. (2008), Semantic vector products: Some initial investigations, *in* 'Proceedings of the Second International Symposium on Quantum Interaction'.

Widdows, D. & Cohen, T. (2015), 'Reasoning with vectors: a continuous model for fast robust inference', *Logic Journal of IGPL* **23**(2), 141–173.

Widdows, D., Kitto, K. & Cohen, T. (2021), 'Quantum mathematics in artificial intelligence', *arXiv preprint arXiv:2101.04255* .

Wiebe, N., Bocharov, A., Smolensky, P., Troyer, M. & Svore, K. M. (2019), 'Quantum language processing', *arXiv preprint arXiv:1902.05162* .

Wijnholds, G., Sadrzadeh, M. & Clark, S. (2020), Representation learning for type-driven composition, *in* 'Proceedings of the 24th Conference on Computational Natural Language Learning', pp. 313–324.

Zanzotto, F. M., Santilli, A., Ranaldi, L., Onorati, D., Tommasino, P. & Fallucchi, F. (2020), KERMIT: Complementing transformer architectures with encoders of explicit syntactic interpretations, *in* 'EMNLP', pp. 256–267.

Author Index