SEMANTIC DIMENSIONALITY REDUCTION DURING LANGUAGE ACQUISITION: A WINDOW INTO CONCEPT REPRESENTATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

ROJDA ÖZCAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF COGNITIVE SCIENCE

AUGUST 2022

Approval of the thesis:

SEMANTIC DIMENSIONALITY REDUCTION DURING LANGUAGE ACQUISITION: A WINDOW INTO CONCEPT REPRESENTATION

submitted by **ROJDA ÖZCAN** in partial fulfillment of the requirements for the degree of **Master of Science in Cognitive Science Department, Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozşahin
Dean, **Graduate School of Informatics** _____

Dr. Ceyhan Temürcü
Head of Department, **Cognitive Science** _____

Prof. Dr. Cem Bozşahin
Supervisor, **Cognitive Science, METU** _____

Examining Committee Members:

Assoc. Prof. Dr. Duygu Özge
Foreign Language Education, METU _____

Prof. Dr. Cem Bozşahin
Cognitive Science, METU _____

Assist. Prof. Dr. Özkan Aslan
Computer Engineering, Afyon Kocatepe University _____

Date: 31.08.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Rojda Özcan

Signature :

ABSTRACT

SEMANTIC DIMENSIONALITY REDUCTION DURING LANGUAGE ACQUISITION: A WINDOW INTO CONCEPT REPRESENTATION

Özcan, Rojda

MSc, Department of Cognitive Science

Supervisor: Prof. Dr. Cem Bozşahin

August 2022, 75 pages

We explore the dimensionality in the semantic representations derived from the Eve fragment of the CHILDES database to gain insights into whether or not semantic dimensionality reduction (DR) occurs during language acquisition, and if so to gain insights into how this reduction of dimensions could look like. We start exploring these representations that are in the form of lambda terms (LTs) by trying to find different representations for them which would be more suitable for the use of DR techniques on them. Seeing as they make the data set more suitable for the application of DR, we prepare a version of the original data set where the LTs are turned into De Bruijn terms (DBTs). After coming up with vectorial representations for the LTs and DBTs (For DBTs we prepare 2 different versions of the data set.), we study the dimensionality in these 3 data sets by utilizing Principal Component Analysis, Cosine Similarity Comparisons, and Affinity Propagation Clustering, and report the results.

Keywords: language acquisition, semantics, dimensionality reduction, de bruijn index, language of thought

ÖZ

DİL EDİNİMİ SIRASINDAKİ ANLAMBİLİMSEL BOYUT İNDİRGEME: KAVRAM TEMSİLİNE AÇILAN PENCERE

Özcan, Rojda

Yüksek Lisans, Bilişsel Bilimler Bölümü

Tez Yöneticisi: Prof. Dr. Cem Bozşahin

Ağustos 2022, 75 sayfa

Dil edinimi sırasında anlambilimsel boyut indirgemenin (Bİ) gerçekleşip gerçekleşmediğine, eğer gerçekleşiyorsa boyutların bu indirgenmesinin nasıl görünebileceğine dair içgörü kazanmak için CHILDES veritabanının Eve bölümünden elde edilmiş anlambilimsel temsillerdeki boyutluluğu araştırıyoruz. Lambda terimi (LT) biçiminde olan bu temsilleri araştırmaya onlar için üzerlerinde Bİ tekniklerinin kullanımına daha uygun olacak farklı temsiller bulmaya çalışarak başlıyoruz. Veri setini Bİ uygulamasına daha uygun bir hale getirmeleri sebebiyle orijinal veri setinin LTlerin de Bruijn terimlerine (DBTler) dönüştürüldüğü bir versiyonunu hazırlıyoruz. LTler ve DBTler için vektörel temsiller oluşturduktan sonra (DBTler için veri setinin 2 farklı versiyonunu hazırlıyoruz.) bu 3 veri setindeki boyutluluğu Temel Bileşen Analizi, Kosinüs Benzerliği karşılaştırmaları ve Affinity Propagation Kümelemesi kullanarak inceliyoruz ve sonuçları raporluyoruz.

Anahtar Kelimeler: dil edinimi, anlambilim, boyut indirgeme, de bruijn dizisi, düşünce dili

To Mind's Search for Meaning

ACKNOWLEDGMENTS

I would like to start by thanking my thesis supervisor Prof. Dr. Cem Bozşahin for his mentorship and support over the past few years. He helped direct the uncontrollable fire that is my passion for the study of meaning towards a sound and steady pursuit of well-defined research questions which are explored throughout this thesis work. I will be forever grateful to him for guiding me during my first significant piece of research.

The process of working on this thesis has been an immense learning experience, and its contributions to my knowledge and experience as a researcher far exceeded my expectations of what I would learn and how I would change when I first came to this department. To that end, I would like to extend my thanks to the Department of Cognitive Science at METU as a whole. Specifically, I want to convey my gratitude to some of my professors in the department. I thank Assist. Prof. Dr. Umut Özge, who had been an excellent teacher to me when I was first getting acquainted with theoretical and computational linguistics. I also want to give my thanks to Assoc. Prof. Dr. Barbaros Yet, whose course on machine learning had been influential for the work in this thesis. I also offer my sincere thanks to Assoc. Prof. Dr. Samet Bağçe and Assoc. Prof. Dr. Aziz Fevzi Zambak, for they have been influential in my thinking in the early stages of my career going into cognitive science.

I thank TÜBİTAK (The Scientific and Technological Research Council of Turkey) for their financial support of this thesis via the BİDEB 2210-A scholarship I have received during my studies in the department. I also thank the examining committee members of the thesis, Assoc. Prof. Dr. Duygu Özge and Assist. Prof. Dr. Özkan Aslan, for their comphrehensive feedback on the manuscript and for their support.

I also want to acknowledge my friends who have been a big part of my support system throughout most of my thesis work. I thank my dear friend Aysu Alkış for always being there for me, and for serving as a role model and a source of encouragement during the toughest of times in that I had the privilege to witness her thesis writing process well before I started working on mine. I thank Mert Küskü, who was often "Walkin" a similar path to mine in his thesis writing process and has frequently served as a companion. Meeting him was one of the best gifts coming to this department has offered me. Finally, I want to thank Atakan Garipler for his encouragement, his undying trust in my abilities, and his companionship during trying times in the past few years.

Finally, I want to acknowledge how thankful I am to my family. The courage and perseverance of my parents in pursuing their own goals have always been a source of inspiration. I am grateful to my mother for her trust in my ability to carve my own path and to follow my dreams, and for her endless support of me in every way possible. I also thank my father for his unlimited support and for the reality checks he has given to the idealist in me. Last but not least, I thank my sister and brother for being my cheerleaders all my life, and for their endless love, support, and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AP	Affinity Propagation
BoW	Bag of Words
BDBV	Bare De Bruijn Vector
DBI	De Bruijn Index
DBILT	De Bruijn Indexed Lambda Term
DBT	De Bruijn Term
DBV	De Bruijn Vector
DR	Dimensionality Reduction
LC	Lambda Calculus
LT	Lambda Term
LV	Lambda Vector
LOT	Language of Thought
LF	Logical Form
PoS	Part of Speech
PCA	Principal Component Analysis
t-SNE	t-distributed Stochastic Neighbor Embedding

CHAPTER 1

INTRODUCTION

The current work is an exploratory study concerned with semantic dimensionality reduction during language acquisition. The approach that is taken in this thesis study is to computationally explore the semantic representations of the words in a child-directed utterance data set, which is the Eve fragment (Brown, 1973) of the CHILDES database (MacWhinney, 2000). The semantic representations of these words are in the form of lambda terms (Church, 1941). The exploration of these lambda terms starts with the process of trying to come up with new ways of representing them in a way that would be most conducive to applying dimensionality reduction techniques to them. The focus is on the predicate-argument structure of the verbs in this data set of semantic representations, particularly on the way these verbs take their arguments as it relates to their semantic properties.

It becomes apparent that a method called De Bruijn index which allows one to remove the variable bindings in lambda terms by representing them numerically could be just what is needed for the purposes here (de Bruijn, 1972). This is because numerical data is, without a doubt, more conducive to the application of computational dimensionality reduction, and the De Bruijn index method helps transform at least part of the semantic representations into a numerical representation. Therefore, the data with the lambda terms was transformed into one with De Bruijn terms, and this data set was the second data set that was used in the thesis along with the original lambda terms with the goal of computationally exploring their dimensionality. Yet, the representations at hand still were not desirable as there still were non-numerical parts in the representations.

After an exploration of existing representational frameworks for non-linear and non-numerical data, a vectorial representation approach was taken for both of the data sets. In case of the De Bruijn indexed version, some features were attributed to the data to create feature vectors that capture the important information in the data. In the case of the original representations with the lambda terms, the vectorial representation was completely based on a Bag of Words approach.

In the upcoming section, a detailed account of the various research questions this thesis aims to gain insights into are presented, and the following sections make a deeper dive into some of the specific research questions that were laid out in the upcoming section.

1.1 Research Questions

The research questions this thesis sets out to answer are as follows:

- The question of whether or not a process of reduction in the dimensionality of the predicate-argument structure takes place during language acquisition is the main research question of this thesis, and the hypothesis is that such a dimensionality reduction does take place. (Addressed in Section 1.1.1.)
- Another research question behind the work in thesis is the question of how verbs' semantics may shape the way they take their arguments (i.e. argument realization (Levin & Rappaport-Hovav, 2005)). The exploration of this question starts, in this thesis, with the dimensionality reduction of semantic representations. Since a successful reduction would yield new semantic representations containing less semantic features than before, that would further the understanding of semantic features that may play a significant role in the syntactic behavior of verbs. (Addressed in Section 1.1.2.)
- In this thesis, the operationalization of the notion of semantic representations is achieved by denoting word meanings with expressions based on lambda calculus (Church, 1941) and De Bruijn indexing (de Bruijn, 1972). What the work in this thesis has shown time and time again is that exploratory work on the previous questions requires much more than sticking with the representations that come from any one formalism. It requires more exploratory work, this time about finding some flexibility in working with the representations by finding suitable encoding tools to transform them. For these reasons, the consequences of one's initial choice of representational formalism and the questions about how to encode these kinds of semantic representations are integral parts of this work. (Addressed in Section 1.1.3.)

1.1.1 Dimensionality Reduction

A well-known language acquisition phenomenon is that children are exposed to various surface structures that resemble each other in significant ways, yet ones that may have different semantic meanings and warrant different semantic representations. Also, they are exposed to surface structures that are vastly different yet that are similar in meaning. Yet, once the language is acquired, they will have solved this complex problem and have learned how syntax and semantics work in that they understand how similar syntactic structures may have differing semantics and how different syntactic structures may have similar semantics. Despite the level of complexity in what they encounter during language acquisition, they acquire the language and learn the words in their language. They build a lexicon in the sense that they have a vocabulary of words for which they understand the rules of use syntactically and also understand the fit of those words into contexts where they are conceptually appropriate. This phenomenon is the main phenomenon of interest to this thesis, and this subsection explains how what goes on during this process could be framed as a dimensionality reduction process.

Dimensionality reduction problems are ones where the amount of variables that seem to determine the outcome of a dependent variable are vast and this vastness is a hindrance to under-

standing the variable of interest. Word learning and representation during language acquisition could be approached as a matter of dimensionality reduction. This is because during the acquisition process, children can be thought to be in a constant process of assigning representations to the utterances they hear. The task in front them is one of understanding particularly the syntactic and semantic information that determine the meaning of a particular utterance they hear, so that they will then be able to create their own utterances with the lexical rules they have gathered before. In other words, one can think of them as trying to figure out just which variables they observe in an utterance (also including the information in the physical environment and linguistic context) determine the meaning of individual words and sentences as a whole.

Yet, it is quite clear that the information children receive is too complex and even messy. Them being exposed to so many surface structures can be thought of as them being exposed to many variables that they have to sift thorough to figure out whether or not they contribute to the meaning of the words they are so desperate to understand. What is even more perplexing is the fact that, in spite of the variation in the dimensionality of the data native speakers of the same language are exposed to as children, they can acquire the same language. These observations suggest that a dimensionality reduction mechanism may indeed be taking place whereby children take only the relevant information from the linguistic data they are exposed to.

Of great interest to this thesis is how this hypothesized process of dimensionality reduction may take place. Particularly the questions of how the initial semantic dimensionality the children are exposed to may look like, and of how the final dimensions may look like after these dimensions are reduced are the questions that are explored throughout the methodology and data analysis chapters of this thesis.

1.1.2 Argument Realization

Argument realization is a name that is used to refer to the particular kinds of relationships verbs have to their arguments (such as the maximum number of arguments a particular verb may take), which is a syntactic relationship that is thought to be related to the way a verb would be semantically categorized (Levin & Rappaport-Hovav, 2005). The nature and the direction of this relation in terms of whether it is the semantics of a verb that determines the syntactic phenomena of argument realization (semantic bootstrapping) or it is the syntax that precedes and facilitates the learning of the meanings of verbs (syntactic bootstrapping), is not very well understood and there is no consensus on it (see (Borer, 2004) for a survey of these arguments). The direction of this interaction between syntax and semantics in argument realization is of interest to the current thesis, particularly as it pertains to gaining insights about the role of semantics in this phenomena.

In other words, one question this thesis ponders is this: If it is the semantics of verbs that determine argument realization phenomena and if verbs could be semantically categorized such that the ones with similar semantic categories would behave similarly with their arguments, how would the semantic features that would need to be included in such an account of semantics and semantic categorization look like? This thesis aims to answer the previous question by means of reducing the dimensionality in the semantic representations to the min-

imum number of dimensions where the verbs' semantics are still intact to then see what these new dimensions could stand for as semantic features.

1.1.3 Selecting a Framework for Representing Semantics

The questions in the previous sections take at face value the representational framework that is used in this thesis, which is based on lambda calculus (Church, 1941) and De Bruijn indexing (de Bruijn, 1972) which gives a notational variant of the former. However, the representational formalism one uses to convey semantic phenomena becomes, naturally, the very subject of the inquiry when one sets out to computationally explore the representations based on that formalism to try and answer questions about semantic dimensionality reduction and argument realization.

In other words, an important focus area in this thesis was the semantic representations themselves. The research questions specific to the choice of representational frameworks stem from their evaluation with regards to their suitability for semantic dimensionality reduction. For this reason, the frameworks are evaluated based on the distinctness in dimensionality the framework allows so that these dimensions can then be reduced. A framework being conducive to the identification of distinct semantic dimensionality in its representations is really about whether or not there are independently extractable semantic features in the representations which can then be computationally explored. This process also requires coming up with suitable encoding methods for the non-numerical parts of the data. Identifying dimensionality and reducing dimensions is also about making sense of what is required from a semantic word representation by pondering about the fundamental components such a representation should have and trying to come up with ways to represent only the features that are necessary and sufficient. These questions are explored in this thesis, either during the process of identifying the dimensions in the existing representations or during the evaluation of the newly reduced dimensionality in the data.

In conclusion, another research question this thesis explores is whether or not semantic representations denoted with De Bruijn indexed lambda terms are better representations than ones with regular lambda terms when it comes to understanding and modeling the way dimensionality reduction can occur during the acquisition of the lexicon by the child. For reasons that will be made clear in Chapter 3, the hypothesis here is that the De Bruijn indexed representations are more robust than the regular lambda representations when it comes to modeling language acquisition.

1.2 Contributions of the Thesis

The contributions of the work done in this thesis are three-fold. The first one has to do with the main subject of interest, in that the thesis gives a computational account of how semantic dimensionality reduction could be taking place during language acquisition, how reduced semantic dimensions may look like, and what these reduced semantic representations tell about semantics' affect on argument realization phenomena.

The second contribution of the thesis is in the way De Bruijn Indexing is used to more successfully reduce semantic dimensionality and to cluster unreduced vector representations of words, suggesting that it is a good candidate as a representational framework that could be integral to the kind of computational mechanism that is involved in the acquisition of the lexicon.

The thesis work required coming up with a numerical encoding method for the semantic representations with lambda terms for the purposes of studying their dimensionality using pre-existing methods, and this has started a side-quest within the thesis for coming up with a way to represent the compositional structures in the data with vectors. Therefore, with the creation and exploration of three types of vectorial representations for the entries in data, a third contribution of this thesis was one of insight into how compositional phenomena could be accounted for in a distributional fashion and whether or not structures like De Bruijn indexed lambda terms may be a way to linearize compositional structures.

1.3 Organization of the Thesis

With the upcoming chapter, Chapter 2, the thesis continues with a brief review of the literature that is organized around the of theories about semantics and representation, as these terms are understood in cognitive science and linguistics. Chapter 2 aims to acquaint the reader with some of the literature and the theoretical challenges around these themes, and tries to make clear the theoretical commitments behind this research (and sometimes lack thereof) either in terms of the philosophical stances that are taken or the linguistic and cognitive scientific theories that served as foundational ideas.

Chapter 3 starts with an introduction of the data set that was used in this thesis, where the original child-directed utterance data came from, and the transformations it went through until the lambda terms that were used in this thesis came to being. The chapter then explains the process of how and the reason why a second data set with De Bruijn terms was created based on the first one. In this chapter, the methodology of the thesis is explained in a step-by-step fashion, making it clear along the way why the lambda terms and De Bruijn terms had to be numerically represented, and what a successful numerical representation of them would look like according to the aims of the thesis. This explanation also serves as a survey of some of the literature regarding numerical encoding methods that could be suitable for the data sets at hand. The chapter then closes with the description of the encoding methodologies the thesis landed on for the data sets, specification of the rationale behind choosing these representational methods, and how and why two versions of the De Bruijn indexed representations were created for use in the analyses.

In Chapter 4, the analyses that were conducted for the 3 data sets are described, along with an explanation of the reasoning behind using particular methods of analyses and reports of the results of these analyses. The analyses follow a three-pronged approach. The first one consists of the use of dimensionality reduction on the data sets (with the Principal Component Analysis (PCA) (Pearson, 1901; Hotelling, 1936) and with t-distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten & Hinton, 2008)) and comparisons of their reductions with different number of target dimensions. The second prong of the approach to understand-

ing and exploring the data sets was using a similarity metric (cosine similarity) in order to explore the similarities of semantic representations within each particular data set and also to aid in the comparisons between data sets (with varying dimensionalities) by way of cosine similarity-based heatmap visualizations. This analysis was meant to serve as a metric of comparison for the unique representational scheme of each of the data sets that were involved in the analysis in that the scheme for the semantic representations of the words within a data set would be considered more successful if an observation can be made that the pair-wise cosine similarity scores of the representations approximate the similarity scores that would be expected of the actual word meanings the representations stood for. Cosine similarity was also used as a measure of success of the linearly reduced representations for certain data sets.

The last prong of the approach to exploring the data sets that was reported in Chapter 4 is the use of Affinity Propagation (AP) clustering (Frey & Dueck, 2005) to visualize and see what kinds of clusters could be found within the De Bruijn Vectors (DBVs) data set, so as to gain insight into the success of this particular representational scheme. The chapter also explains how and why different subsets of the verbs in the DBVs data set were used in employing the analyses above. The thesis concludes, in chapter 5, with a brief discussion of the results, possible implications of the findings, and some pointers for future work that are given based on these findings.

CHAPTER 2

BACKGROUND

In this chapter, a survey of related work is presented. Note that in Chapter 3, which is about the selection of a representation method for the data set, there will be coverage of literature as well. In this chapter, the studies are grouped into sections that cover the theoretical background relating to the issues that are of interest to this thesis. Most of these sections are about notions on which the scientific community has not arrived at a consensus. Very broadly, these issues relate to the topic of the nature of semantic representations in the mind, particularly as it relates to linguistic meaning and thought. These sections are non-exhaustive surveys of some conflicting theoretical approaches that are important for the questions this thesis sets out to answer, hence they are conceptualized as a matter of theoretical commitment that change significantly one's approach to linguistic meaning.

2.1 Committing to a Theory of Semantic Representations and Representation of Content

There are many conflicting ideas in the literature about what meaning is in the context of cognition, how it is represented, what relationship linguistic meaning has to this more general notion of meaning, and many others. This section is a short survey of how these notions are understood nowadays, focusing only on the aspects of the discussion that are relevant to the current work. The section also tries to make clear the theoretical tendencies behind the work in this thesis towards certain camps of thought on these matters.

One important divide in the field is regarding the notion of meaning in cognition. The divide is in identifying the set of structures and the kinds of mechanisms that would be necessary in the facilitation of the mind's capacity for meaning-bearing cognitive processes, and in identifying those, defining the conditions for the approximation of such capabilities in a computer. Since this thesis takes on a computational modeling approach to tackling the questions about various psycholinguistic phenomena of interest, the underlying assumptions in the thesis about the structures and mechanisms required for a computer to model meaning-bearing processes are equally important as the assumptions about those required for a meaning-bearing process in the human mind. Hence, this disagreement in the literature will be mentioned here so as to clarify the assumptions of the thesis.

2.1.1 Searle's Chinese Room

A brief overview of Searle's Chinese Room argument (1980) is perhaps the best way of conveying the decades-long disagreement about the kinds of structures and mechanisms that must be at work in the mind when it comes to meaning, or ones that must be employed for a notion of meaning for a computer. Searle (1980) puts forth a thought experiment in which he, as a person who does not speak Chinese, is put in a room where the only interaction he has with the outside world is when Chinese expressions are sent to him, for which he then needs to find appropriate answers in Chinese to write down and send out. The question he poses is that, if the only thing aiding him in this task is that he has a rule book in which there are orthographic demonstrations and rule descriptions in English (a language which Searle knows) showing exactly and without fail which kinds of surface forms should be answered with which kinds of other Chinese surface forms, could we say that the Searle in this room understands Chinese (Searle, 1980)? Searle proposes that in this scenario he would be following a program that is based on pure symbol manipulation without any information about the semantic content underneath and that for this reason we could not say that he understands Chinese. The argument is made to make the same point about computers, namely, that they do not have an understanding of what they are doing, that all they are doing is engaging in formal operations over symbols.

There were many answers to the Chinese Room argument (Searle, 1980), and ever since its initial reception, it shaped the discussion around syntax, semantics, and their place in a computational theory of mind. Another line of argumentation that is related to the former (Searle, 1980) follows from another work of Searle's (1990) regarding the nature of digital computers and whether or not brains can be characterized as digital computers. In this work (1990), Searle makes the claim that computers are solely syntactic engines and do not have semantics that is intrinsic to them, while brains are capable of semantics by virtue of the causal powers that stem from their biological makeup. Here (Searle, 1990), the claim that is made is about the nature of computers and not just about the issue of computers not being capable of understanding the referents or the meaning of the programs they can run as it was in the case of Searle (1980).

These two arguments spanning the nature of computers (as in the claim about them being purely syntactic engines (Searle, 1990)) and the nature of running a program in a computer (as in the claim about programs being purely syntactic processes with no underlying semantic correlate for the computer (Searle, 1980)) are important for the purposes of making clear the theoretical commitments involved in this thesis as it relates to the possibility of computationally modeling natural language and its semantics. The thesis endorses the account of computers outlined in Bozşahin (2018) as being partially syntactic engines that are built up from the semantic primitives that are involved in the physical manifestations of said engines. The thesis also endorses the view that the programs that are run on the computers (as the kinds of engines described in the previous sentence) are syntactic symbol manipulation operations with semantic primitives involved in the physical manifestation of particular computations (Bozşahin, 2018).

Contrary to some accounts in the literature which, in an attempt to reconcile Searle's criticisms (Searle, 1980, 1990) resort to accounts that reduce semantics to being an emergent property of

other linguistic aspects of the natural language such as the syntax or the pragmatics, this thesis takes semantics as distinct from syntax in fundamental ways and denies that any sufficient account of natural language semantics could be deemed as syntactic just on the basis of it also involving computational manipulation over structures. The work here focuses on uncovering and highlighting the key features of semantic meaning representations that set them apart from the syntactic dimension of the same representations, mostly on the basis that they are compositional structures.

2.1.2 Language of Thought

Subscribing to a theory of semantic content that not only successfully bypasses the philosophical challenges posed by Searle (1980, 1990) and many others not mentioned here but also meets the criterion of being compatible with a model of natural language that fits psycholinguistic observations and related literature on theoretical linguistics is of utmost importance to the theoretical backbone of this thesis. The exploration in the thesis is on semantic representations, and ingrained in the process is an approach that posits that the semantic representations of the words in the data contained in their compositions individual concepts that contribute to the meaning of that particular word. In other words, by way of exploring compositional semantic representations of words, the thesis also explores the conditions for a possible conceptual structure in the mind which could then be considered as a precursor to a study of compositions of thoughts from concepts. The exploration just mentioned is motivated by the desire to explore the language of thought (LOT) hypothesis (Fodor, 1975, 2008), which holds that thought and language resemble each other in some key respects in that they both operate on compositional structures and have systematicity in their productions of meaningful tokens of thought and language. This thesis accepts and operates under these assumptions in that the semantic representations in the thesis are taken to be compositional and systematically producible, which are seen as semantic properties that should ideally be preserved in each truthful transformation of these representations either for dimensionality reduction or encoding.

In keeping with this goal of subscribing to a psycholinguistically plausible theory of content that also aligns with the linguistic properties that LOT proposes, let us revisit the challenges posed by Searle (1980, 1990) to a computational theory of mind. These challenges necessitate that one takes a different stance about the nature of semantic content and the kinds of structures that are involved in their creation so as to meet the strict satisfaction conditions for a process to be cognitive or computational by Searle's accounts (1980, 1990) and they therefore pose a threat to a computational model of natural language semantics. Therefore, the approaches in the literature to semantic content that try to address and reconcile those challenges land on significantly different claims about the content of semantic representations in the mind, and about how they can be computationally studied.

It is important for the conceptual groundwork of this thesis to briefly mention one particular approach to semantic content that is called Conceptual Role Semantics (CRS) as introduced in (Block, 1986), which seems to have gained traction among cognitive scientists and philosophers that tackle empirical and philosophical problems similar to the ones mentioned in this literature survey. This approach to semantics argues that words and expressions take their meaning-bearing content inside the context of the utterance in which they occur and thus

from their role in that particular context (Block, 1986). The conceptual role which is thought to be an emergent property of utterances is taken to be the meaning of individual expressions, however, the conceptual roles that are supposed to stand for word meanings are underdetermined by the theory itself unless clearly and more or less arbitrarily specified for a small set of expressions that are used to demonstrate the way the theory should work. Another way to describe this approach is that it is a use theory of meaning, inspired by Late Wittgenstein (1953), where the words do not bear meanings until they are used in syntactic constructions. Close neighbors of CRS (by virtue of them being use theories of meaning that also seem practically indistinguishable from CRS) are Inferential Role Semantics and Functional Role Semantics (see Block (1986) for an introduction). This approach to semantic content will later be addressed in Chapter 3 when discussing the possibility of making use of an implementation of CRS by S. T. Piantadosi (2021).

The kind of the theory of semantic content this thesis tries to give insights about is one where the systematicity of thought and language are utilized in a way that will allow big-scale computations on the meaning-bearing constituents in utterances. This thesis aims to give an account of natural language semantics that is based on compositional and systematic representations (see Fodor and Pylyshyn (1988) for arguments on why systematicity and compositionality are important properties semantic representations should have), while also keeping in mind the need to later bring together any emerging theory of semantic content with the greater scale of linguistic operations including but not limited to the syntax-semantics interface.

2.2 Committing to a Theory of Argument Realization and Language Acquisition

As it was clarified in the introduction chapter, the thesis takes into consideration a data set of child-directed speech where it sets out to gain insights into the factors that go into the child's internalizing argument realization phenomena during language acquisition. To better appreciate the complexity of the questions regarding language acquisition and argument realization phenomena, one has to get acquainted with the related literature in linguistics, specifically in lexical semantics, and this section aims to give a brief overview of the discussions in the literature.

There are two different explanations that have a foothold in the literature with regards to the language faculties that are theoretically involved when the child is to acquire the language, which are the Semantic Bootstrapping hypothesis and Syntactic Bootstrapping hypothesis (Borer, 2004). The first one of these approaches, the Semantic Bootstrapping hypothesis (see (Grimshaw, 1990; Abend, Kwiatkowski, Smith, Goldwater, & Steedman, 2017; S. Piantadosi, Tenenbaum, & Goodman, 2012)), purports that the lexical semantic information in the predicate-argument structure of verbs give way to their syntactic properties. The latter approach (see Gleitman, Liberman, McLemore, and Partee (2019)), on the other hand, purports that the syntax of the argument structure aids the child in acquiring the semantics of the predicate-argument structure. While these approaches differ in the emphasis they give to the syntax or to the semantics of verbs, they both depend on an ongoing interface between syntax and semantics.

Although these two approaches are prominent in the field, there are some other approaches that outright reject the idea that the child has to come to the world equipped with certain faculties (Chater & Christiansen, 2018) which predisposes the child to favor syntactic or semantic representations. These approaches try to overcome the problems that come with assumptions about the innateness of certain faculties in the child during language acquisition, and they emphasize learning in a way that makes it possible to study the acquisition phenomena with predominantly experimental methods (Chater & Christiansen, 2018) that do not rely on most of the assumptions of theoretical linguistics which may, for those in this camp, make it seem like language acquisition is a mysterious phenomena that is hard to make sense of (see Gleitman et al. (2019) for a discussion).

This thesis tries to make sense of language acquisition and argument realization phenomena in a way that favors the Semantic Bootstrapping hypothesis, because the approach here tries to come up with a model of dimensionality reduction for the semantic representations which could possibly reveal the syntactic properties of verbs.

2.2.1 Perspectives on Dimensionality Reduction During Language Acquisition

The motivations behind the goal of the thesis to give an account of semantic dimensionality reduction were laid out in Chapter 1. In this subsection, related literature is briefly reviewed to give a better picture of the approaches to semantic dimensionality that can be found in the literature.

The most notable findings on the topic suggest that the framework of dimensionality reduction may work to learn the compositional structures (Ellis, Dechter, & Tenenbaum, 2015; Katz, Goodman, Kersting, Kemp, & Tenenbaum, 2008). For instance, the approach in Ellis et al. (2015) makes use program induction to learn representations of compositional structures, doing, along the way what they call *symbolic dimensionality reduction*. The approach in Katz et al. (2008), in a similar vein, takes dimensionality reduction as a fundamental aspect of the acquisition of semantic knowledge by taking into account the sparsity that is inherent to learning from knowledge that is not rich enough for seamless learning, which, in a way, counteracts the curse of dimensionality that is encountered when acquiring concepts.

CHAPTER 3

THE DATA SET AND THE REPRESENTATION METHODOLOGY

As explained in the introduction chapter of the thesis, the main aim of the thesis is to gain insights into whether or not a process resembling dimensionality reduction as what is usually understood by the term goes on in children's minds during the acquisition of language, and if so, in what kind of abstractions then this reduction of semantic representations would result. The scope of the thesis is narrowed to include only the study of semantic dimensionality reduction during language acquisition because of time limitations.

The process started with the extraction of the semantic representations from a pre-existing child-directed utterance data set which was in the English language. The methodology is built around the aim of taking these logical forms (which are lambda terms) and reducing their dimensions using various methods to later compare and contrast a variety of reduced meaning representations that are created in the process. The reason why there was a need for a chapter specifically dedicated to the data and its representation has to do with the nature of the data, namely, with the semantic representations that made up the data being lambda terms which were not seen to be adequate in themselves for use in dimensionality reduction.

In the first section of this chapter, the data set which was the source of the lambda terms used in the thesis was introduced and described in detail with some examples where it was deemed necessary. Afterwards, brief introductions to lambda calculus and De Bruijn index method was given along with descriptions and exemplifications of the lambda terms in the data and the De Bruijn indexed lambda terms that were created using the data.

The second section is dedicated to explaining why the non-linearity in the data structure did not seem satisfactory for computational exploration and to surveying ideas and methods of creating a more desirable representation before attempting any dimensionality reduction. The section includes a report of the research that went into the exploration of the literature regarding possible numerical encoding methods for the lambda terms and De Bruijn terms, none of which was used for the encoding of these terms in this work. The section addresses the reasons why these methods were not seen as feasible methods for the data at hand, and what their application to the data would result in for some methods so as to make the rationale behind the ultimate representation methodology clear.

The third and the last section goes into the details of the representation method the thesis landed on to linearly represent three different versions of the data set (the original lambda

terms, De Bruijn indexed lambda terms, and the De Bruijn indexed lambda terms with their lambda symbols removed).

3.1 Data Set

The data set that was used in this work comprises lambda expressions that are extracted from a lexicon that consists of the words that appear in a child-directed utterance data set, which is the Eve fragment of the English Brown Corpus (Brown, 1973) in the CHILDES Corpora (MacWhinney, 2000).

The description of the data set creation process is presented in two parts. In the first part, the process of the construction of the Eve fragment by Brown and colleagues (Brown, 1973) is described. In the second part, the process of how, then, a lexicon was built using this data before the work that is done in this thesis began is laid out, together with the characteristics of the said lexicon.

3.1.1 Eve Fragment of the CHILDES Corpora

The Eve fragment is part of a longitudinal study on the acquisition of the English language by Roger Brown and his colleagues Ursula Bellugi and Colin Fraser which went on between the years 1962 and 1966, (Brown, 1973) during which they conducted interviews that lasted about 2 hours with 3 children once every two weeks. The data sets with the transcriptions of the child-directed speech for children given the monikers Adam, Eve and Sarah make up the CHILDES English Brown Corpus (Brown, 1973), and the data that is used in this study comes from the transcriptions involving the child called Eve. Eve was 18 months old when the study first started in 1962, and has left the study at 27 months old, which amounted to 20 sessions in total. She left the study because her family moved out of the city where the study was being conducted. The Eve fragment comprises the utterances directed to the child by her parents and also the researchers conducting the study and the utterances of the child herself, all of which are transcribed from the recordings of the conversations that take place between them.

Following is a fragment from the transcriptions for Eve. Note that 'CHI' stands for the child, 'COL' stands for Colin Fraser, one of the researchers conducting the study (Brown, 1973).

(1) **CHI:** where Mom ?
COL: I don't know .
COL: where's your Mom ?
CHI: where Mom ?
COL: I don't know ?
CHI: in Papa study ?
COL: in Papa's study ?
COL: oh that's where you Mom is .
CHI: where my Mom ?

COL: I don't know .
CHI: in Papa study .

Aside from the purpose of giving an actual example from the data, this particular example is picked because it was deemed to be a representative sample of the data in that it captures a repeating pattern. Notice how during these interactions between Fraser and Eve, Fraser makes some assumptions about what Eve is trying to convey, and repeats those back with a corrected version. For instance, when Eve says "in Papa study?" Fraser says "in Papa's study?" based on the inference he makes in the context of Eve's previous questions. This is a pattern in communication which is pervasive throughout the data whether Eve is interacting with a researcher, her mother, or her father. Brown and Bellugi in their work (Brown & Bellugi, 1964) call this phenomena *imitation with expansion*, a work where they list many other interesting phenomena they observed while conducting the aforementioned study.

The reason why this particular phenomena is mentioned here is to highlight how assumptions are made about the child's use of language during the acquisition process by the observers of this process. Of course, the assumptions do not stop there. Assumptions are present when transcriptions of the speech are made, they are present when the researchers annotate the data. As we shall see, they are going to be present when other researchers find the dependencies in the data, when yet others create a lexicon using these dependencies, and when, finally, features of the semantic representations of the words in the lexicon are represented for use in this work. Despite the effects of these assumptions on the data set that stem from the original observations which may skew the representations at each attempt at abstraction, it is a fair price to pay when it comes to studying phenomena as complex as language acquisition and semantic representation. The silver lining is having the opportunity to work on language acquisition data that a lot of researchers over the years took interest in and worked on, rendering it an invaluable data set that is filled with the assumptions that come from their intuitions about the English language and their expertise on linguistics.

3.1.2 Lexicon Construction

The lexicon of the some of the words in the Eve fragment that is used in this thesis was constructed by Şakiroğulları (2019) for use in a previous master's thesis. The lexicon creation process (Şakiroğulları, 2019) was carried out using the data set obtained from Abend et al. (2017) study where they selected 5123 of the utterances in the original Eve fragment based on their usefulness for their study on semantic bootstrapping during language acquisition. Abend et al. (2017) excluded utterances that were longer than 9 words and ones that were too short, such as one-word interjections. Utilizing a Bayesian learning algorithm, their goal was to use pairs of 5123 utterances and their corresponding meaning representations to train a Probabilistic version of a Combinatory Categorial Grammar (PCCG) (See (Steedman, 1996) for the CCG formalism) lexicon to discover the acquisition of various aspects of language. For the meaning representations, Abend et al. (2017) used the semi-automatically constructed Davidsonian logical forms Kwiatkowski (2012) created for the Eve fragment by making use of the dependency annotations in Sagae et al. (2010).

Şakiroğulları (2019) took the Abend et al. (2017) data set consisting of the utterances in the Eve fragment and he turned it into a data set of 5134 utterances by breaking down some of the utterances and by making some other slight modifications. Şakiroğulları (2019) needed to use utterance-meaning representation pairs to train a PCCG model to study the bias towards accusativity and ergativity. While he used a modified version of the utterances he took from Abend et al. (2017) as the utterances needed for training, he did not use the meaning representations of Kwiatkowski (2012) like Abend et al. (2017) did and constructed the logical forms corresponding to each utterance himself.

The CCG lexicon Şakiroğulları ultimately used, which contained entries for words that came up during the utterances, became the data set that was to be used in this thesis to study semantic dimensionality reduction during language acquisition. In this lexicon, each word entry consists of the surface form of the word, the part of speech (PoS) tag of the word, its syntactic representation that reflects the combinatory properties of specific syntactic classes that would interact with a given word, and the semantic representation of the word. Semantic representations in the lexicon are denoted with lambda calculus, and this part of the original lexicon was the part that was used in this thesis with the aim of studying the dimensionality of these representations.

To get a glimpse of some of the properties these word entries can have, see (2), an example word entry from the original lexicon (Şakiroğulları, 2019). In the data, which consisted of 4054 entries, there were 14 instances with the surface form "move" with differing syntactic and semantic categories, one unique representation out of those 14 is shown in the example (2).

(2) move tv1> := s [type=inf] \np / np : $\lambda x \lambda y.simp' \ prt' \ move' \ x \ y$

Per the CCG lexicon specifications followed here, the word forms are followed by their part of speech (PoS) tag ('tv1>' in this example stands for the word being a transitive verb ('tv') in the basic form (marked with '1') and one that is accusative (marked with the symbol '>'). Following the ":=" symbol comes the syntactic category of the word (Mind that this syntactic category is based on the CCG formalism. '[type=inf]' in S means that the final version of a syntactic construction with this particular representation will result in an informative sentence construction). Non-informative sentence constructions in the data set include questions, imperatives and non-sentential utterances (Şakiroğulları, 2019). The colon marks the end of the syntactic category and the beginning of the semantic category, which is a lambda term. What this particular lambda expression means will be explained in the next section to accompany a very brief overview of lambda calculus (Church, 1941) as a denotational formalism for linguistic meaning.

In this thesis, only the logical forms (LF) or the semantic representations of the words which were denoted by lambda terms (LT) were taken into account. Note that, since one of the early aims of the thesis was to gain insights into argument realization and the syntax-semantics interface, incorporating a study of the dimensionality of the syntactic representations in the data into the framework for semantics that was built in this thesis by studying their dimensionality reduction in relation to each other remains to be a future goal.

3.1.3 Introducing and Exploring the Data Set: Lambda Terms and De Bruijn Indices

In this section, the lambda terms arising from the previously described lexicon are introduced along with their transformation into the De Bruijn indexed (de Bruijn, 1972) lambda terms. Later on in the section, a survey of possible encoding methods that could be used on the data is presented together with an explanation of the reasons why there was a need for an encoding and the rationale behind not using any of those methods in this thesis.

It is a common practice in linguistics to denote word meanings with logical forms, and specifically with lambda calculus (Church, 1941). This practice is followed here, not only by virtue of it being used in the CCG lexicon data set that is being explored, but also because it is a robust formalism for representing compositional structures.

Lambda calculus is a mathematical formalism created by Alonzo Church (1941) whereby one can write abstract function applications by representing this abstraction by making use of the λ symbol to bind any variable that is in that lambda's range and that has the same name that the λ symbol accompanies (i.e. λx binds any variable in its range with the name 'x' and not 'y'). As a lambda expression takes arguments, the variables in the body of the lambda (i.e. the part of a lambda expression that comes after all the λ symbols in that expression) get substituted by the arguments, which is a process the order of which is determined by the ordering of the variable names that are attached to the λ symbols and requires one to keep good track of where a given λ's binding starts and where it ends.

There is no need to delve into the principles for the order of substitution and other properties of lambda calculus since they are not necessary to understand the lambda expressions in the data set. This is because even though most of the lambda terms in the data have a nested structure and are therefore complex at first sight, the boundaries of the lambda bindings are easy to understand, so one can easily make the correct substitutions. Also note that in the context of this thesis and under the representational scheme that the formalism itself brings, any term -with or without an actual lambda symbol in it- will be called a lambda term (Note that any expression without a lambda in it could be rewritten using a lambda.).

Of utmost importance for the purposes here is to convey what a lambda term denotes regarding the semantics of a given word in the data set, so that the next steps in the methodology regarding the ways to represent and reduce the dimensionality of lambda terms will be clear. For these purposes, see the lambda term (3) extracted from the entry in (2).

Before going into the explanation of the meaning the term conveys, note that the strings in the lambda body that are immediately followed by the *prime (′)* symbol are what, from now on, will be called the concept words. Whenever they are referred to in this text or in examples, they are either followed by the prime symbol or written in all capitals letters (e.g. 'simp′' versus 'SIMP'). The concept words can be thought of as qualifiers of the relationship the arguments will have with each other (when the variable names, 'x' and 'y' in this example, are substituted with the arguments that are received.). The way the potential arguments and the constituent concept words relate to each other is reflected in the structure and the compositionality of the lambda term itself, particularly in the information about the number and order of the arguments, and it is this structure that is the most central to the meaning of the word entry that is taken into account. The concept words themselves that are a part of the kinds

of structures that are used in this formalism, on the other hand, are not taken as semantic units that have internal constituent structures that could be subjected to scrutiny here. Rather, echoing the referentialist sentiment in Fodor (2008), they can be thought of as atomic units that serve as conceptual primitives that take their meaning from their arbitrary reference to the world.

The concept words in the data set as constructed by Şakiroğulları (2019), on the other hand, are the components of a lambda term that are informative about the tense (like 'PRT' in (3), standing for 'present tense') or aspect (like 'SIMP' in (3), standing for 'simple') of the constructions that are possible with that particular meaning representation. All of the utterances start with the concept words for aspect and tense, in that order. Other short-hands for aspects that appear in the data set are 'CONT' (meaning 'continous') and 'PRFT' (meaning 'perfect'). When it comes to the tenses, 'PST', 'FTR', 'GNG' stand for 'past', 'future' and 'going-to/gonna' respectively. Modal verbs in the data ('can', 'must', and 'would') also have their respective concept words and they are preceded by 'SIMP' at the beginning of the meaning representation.

One of the concept words in the lambda terms directly denote the concept behind a word entry (in the case of (3) the concept word 'MOVE'). Some word entries in the data, for instance, are various word forms that contain in their semantic representation the same concept word. However, since in those instances, the overall structures of the lambda terms containing that common concept are significantly different, the resulting meanings are distinct from each other.

(3) $\lambda x \lambda y.\text{simp}' \ \text{prt}' \ \text{move}' \ x \ y$

When it comes to understanding the structure of the lambda term and the overall meaning of the expression in (3), a parenthesized version of it as shown in (4) might help. One rule for parenthesizing the elements in a lambda term to keep in mind is that the components in the body associate to (so are parenthesized with) the component to their left while the lambdas themselves associate to the right. If one is to start with the innermost parenthesized expression in (4) which is '(simp′ prt′)', operating under the knowledge that the expression on the left is the function and the one to its right is its argument, we can see how the expression tells us that the aspect-tense pair modifies the concept 'move′'. If we take into account the next parenthesized expression with the modified verb concept as its first element, we see that this first component will apply to the variable 'x' when it takes an argument and thereby reduces the lambda binder λx. This means that this first argument will be an entity which is affected by the function in the concept words (one of being 'moved' under these tense-aspect restrictions), creating in the next innermost parenthesis a function that stands for this relationship. This function will then be applied to 'y' when it takes its argument, which will be the causer of 'x' being moved. Note that when 'y' takes an argument and λy is removed, the lambda term does not possess any further reduction possibilities so what is called 'the normal form' is reached.

(4) $(\lambda x \ (\ \lambda y \ (\ (\ (\ (\ \text{simp}' \ \text{prt}' \ ) \ \text{move}' \ ) \ x \ ) \ y \ ) \ ) \ )$

Having had access to lambda terms from the Eve fragment, the question was one of coming up with a way to reduce their dimensions by computational means. The thing that should be emphasized is that the nature of the data was one that made it hard to attribute it dimensions, let alone actually reducing any of this dimensionality in them since they contain nested function applications. One should keep in mind that lambda expressions denote states of affairs that are compositional and that these representations are therefore non-linear. Although these expressions are non-numerical, they are not exactly textual either. Where they denote concept names they have textual data (Prime($'$) symbols are removed during computation.), and they may otherwise have multiple lambda symbols and variable names (like x, y, z..). To understand how complex some lambda terms in the data can get, see example (5) from the data set that has 5 lambda binders.

(5) **Word:** doing
Lambda term: λo λx λy λz λa. cont$'$ z do$'$ (o a) x y

Since the hypothesis was that the current structure of the data, with them being made up of lambda terms, would not be very useful as they stand to try out techniques to reduce the semantic dimensionality of the overall data, it was clear in the beginning of the exploration that transforming them into a numerical structure would have the benefit of opening up new ways of computationally studying these representations. There exists a method that can be used on the lambda terms (LTs) to transform some parts of the representations into a numerical representation, which is called the De Bruijn index. In the upcoming subsection, the De Bruijn indices are described in detail along with examples from the data set to demonstrate how the data was transformed.

3.1.3.1 De Bruijn Index

In 1972, Nicolaas Govert de Bruijn came up with a method to numerically represent the variable bindings in lambda terms. The goal was to get rid of the variable names so as to solve the variable binding problem whereby the variable names in the lambda expressions made it harder to use them in computation because it was hard for the computer to keep track of which variable name is within the bounds of a particular lambda binder and to name each new variable with a different name which necessitated keeping track of all the others.

The application of De Bruijn indexing involves taking every variable name in a LT one by one and replacing it with an index that stands for its distance to the lambda that binds it (smaller numbers denoting them being closer) until there is no variable name left in the lambda body and there is no lambda binder left (de Bruijn, 1972).

For instance, the upcoming examples show the conversion of the lambda term (6) to a De Bruijn term . Since the lambda binder that binds the variable 'y' is not the first but the second lambda binder, it gets an index of 2. The closest lambda binder is the one that binds 'z' so it gets the index 1. Finally, the third lambda binder when we count starting from the closest one is the one which binds 'x', so it gets the index 3.

(6) λxλyλz. give′ y z x

λλλ give′ 2 1 3

Realize how it was clear that De Bruijn indexing would be very promising for the purposes here because it would help to reduce and abstract away parts of the entries in the data that were the hardest to deal with, namely the lambda binders and particular variables that were bound by those binders. The most attractive benefit of using De Bruijn indices was that they made a significant portion of the lambda terms in the data numerical. Also note that representing lambda bindings numerically meant that one of the most informative parts of LTs (informative in terms of compositionality) would have a numerical representation now, which would make it easier to study their dimensionality.

Given these perceived benefits, a new data set with the transformed versions of LTs into De Bruijn terms (DBTs) was created [1]. While DBTs were seen as a chance to use lambda terms in a way that is true to their nature (as a way of capturing their compositionality and non-linearity in a linear fashion), linear representations of the original lambda terms were created as well, so that they could be used as a metric of the success of De Bruijn indexing (Since these linear representations for the original LTs would be naive representations and could work as random controls -in terms of feature representation choice- and a baseline.).

Another promising aspect of the De Bruijn Index method comes from the original purpose of its creation by de Bruijn. In his work where he introduced the method (de Bruijn, 1972), he suggested that the De Bruijn indexed terms are more computer-readable, while it is thought that lambda terms are much easier for human beings to read and understand since the bindings are explicitly stated in a way that is more trackable. However, when we take into account the computational complexity using the lambda terms brings, an interesting angle to keep in mind while using and comparing these representations could be to investigate whether or not De Bruijn terms and dimensionality reduction done with them would be, from a modeling perspective, more akin to the process that goes on in the child's mind during the dimensionality reduction in language acquisition.

3.1.3.2 The Characteristics of the Data Sets

The DB indexed LTs and the LTs range from having no lambda bindings in them to having at most 5 lambdas. The number of unique concept words or markers in these logical forms is 1040. The total number of word entries in the lexicon was 4054, 2 of them were there to denote semantic rules regarding the use of the lambda terms in CCG during parsing, so they were dropped from the data set.

Since the main focus in this thesis is the argument realization phenomena in the data, of particular interest is the number of verbs in the data. Out of 4052 word entries, 2700 of them are for verbs. These verb tokens are of 255 distinct types of PoS. Note that, even though the

[1] For the implementation that was used in this thesis, see (Bozşahin, 2021)

main focus is on the verbs in the data, most of the techniques that were employed to explore the dimensionality in the data were also applied to the whole data set. More information about the statistics about the verbs were laid out in the next chapter where certain analyses were performed for certain subsets of verbs in the DBV data set.

3.2 Exploration of Methodology for the Problem of Representation

Now that a more promising representation of the data was found and the De Bruijn indexed representations were created, a search for computational means of reducing the dimensionality in these semantic representations began. In two of the upcoming sections, two distinct approaches to this task of dimensionality reduction for both the lambda terms and De Bruijn terms are laid out.

3.2.1 Theory-Driven Algorithmic Approach

One way to approach the problem would be to come up with a model of semantic representations with LTs and DBTs so as to build an algorithm that would find the commonalities across the data entries for each data set to minimize and make denser the representations and therefore to reduce their dimensionality.

The structural commonalities in the data with LTs could be found by looking for:

- commonalities in the binding patterns of across the data (Note that DBTs conveniently abstracted these bindings to a format where pattern recognition is possible.)
- repeating lambda terms that are nested in most of the lambda terms to reduce them to a higher level abstraction (e.g. We could find an abstraction for the tense-aspect pattern '(SIMP PRT)')
- LTs where only the concept words differ with the same structure in the term itself so as to then cluster them.

The structural commonalities in the DBTs indexed data could be found by checking:

- whether or not there are repeating patterns in the De Bruijn indices across the data (e.g. repeating '1 2's that are standalone or that occur in combination with a certain set of patterns like with '3 4' and with '4 3'), or patterns such as symmetrical De Bruijn indices in certain representations like '1 2 3' and '3 2 1'
- the patterns of co-occurrence involving the appearance of certain concept words with certain De Bruijn index patterns
- to find DBTs where only the concept words differ with the same structure in the term itself so as to then cluster them.

Writing an algorithm to find these kinds of commonalities could aid in the understanding of exactly which structures are redundant in the LTs or DBTs when it comes to predicting the meanings these terms denote, so that they could be explained away in the reductions. It could also aid in the understanding of the kinds of structures that could be the building blocks across the representations in each data set in that they are essential in distinguishing one meaning from another by finding denser representations for them.

The problem with finding and representing these kinds of patterns in these data sets is really a problem of not knowing what kind of patterns are actually relevant without a theoretical commitment to argument realization. When making a decision about whether a pattern is actually redundant or not, it would not be enough to use one's intuitions and knowledge as a semanticist. Knowing the role of the semantic representations in argument realization would be fundamental for making a decision about which features could be considered negligible and could therefore be reduced completely, and about which features were the ones to keep because they are essential for argument realization.

However, committing to a theory of argument realization was not possible (for the reasons why, see the relevant literature survey in Chapter 2) and also not really attractive for the purpose here, which was to give an exploratory account and an empirical study of the dimensionality in the data without making any more assumptions than what was already made during data set construction and later, as the thesis progresses, in the phase of feature vector creation for the two data sets. Creating numerical representations for the data sets and letting dimensionality reduction and clustering algorithms do the job of finding and reducing the patterns in the data seemed to be a straightforward and unbiased way of working on the data.

3.2.2 Formal Approaches

The algorithmic approach such as the one outlined in the previous section did not seem feasible, so a numerical representational approach for the two data sets was sought out. It is also important to note that the computational exploration of the dimensions in the data by way of creating numerical representations of them was seen as something that would aid in the advancement of the understanding of argument realization phenomena in that the reduced dimensions could point to the semantic features that are most relevant to argument realization. Numerical representations that are obtained in the process could be reduced by utilizing one of the frequently used linear dimensionality reduction techniques in the field of machine learning such as Principal Component Analysis (Pearson, 1901; Hotelling, 1936) so as to start somewhere with the exploration of the data, continuing with the broadening and the refining of the types of techniques that are used as the data sets and their characteristics in terms of dimensions are better understood.

A numerical representation could be a vectorial representation based on feature vector creation or could be obtained by any other kinds of encoding methods. While some numerical encoding techniques were surveyed in this chapter, it became clear during the exploration that trying to come up with vectorial representations would be more congruent with the use of the kinds of dimensionality reduction techniques that were being considered as exploration tools. The desirability of creating feature vectors for the representations comes from this method of numerical encoding being a method that requires one to try to come up with sen-

sible numerical representations for the features in the data that are seen as important. These representations maybe simplistic, such as in a binary encoding scenario where the a bag of words technique is used, just representing whether or not a particular feature that is present in the overall data is also present in a single data point. Yet, for the purposes here, feature vectors were considered to be more useful than any other encoding method where the assignments of numbers are arbitrary thus, in most cases, hard to make sense of. One could, on the other hand, look at a hand-crafted feature vector and understand precisely the term it stands for.

Conceptually, the idea of using vectorial representations help frame the problem of dimensionality reduction in language acquisition more concretely. One can think of the number of dimensions in a certain surface structure that the child has to deal with as the number of features in a vector that represent the unique meaning of that surface structure. Thus, one can conceptualize the task of reducing this dimensionality as a task of finding features or independent variables -based on the original feature vector- that more prominently, consistently, and accurately predict the meaning of a surface form which can be seen as a dependent variable.

An important reason why a vectorial representation of the data would be beneficial is that it would make it possible to work on the data in the vector space and get access to their distributional semantic properties. It would be possible to reason about the similarities of the representations even before applying any dimensionality reduction technique to them, which could then can be compared to the similarities of the ones with reduced dimensionalities. This comparison, in turn, could be informative with regards to making a choice about how many dimensions should the original representations be optimally reduced to so as to get similarity scores that are more correct, thereby gaining insights about dimensionality reduction during language acquisition. Also, if the vectorial representations that are created for the DBTs or LTs are successful to some degree, these vectors could be investigated further by way of examining their combinatorial properties and the similarity outcomes for the combinations of these vectors under some linear algebraic operation, which could serve as a test of the success of the vectorial representations of these compositional structures in the first place and as a method of gauging the kinds of improvements that could be made in their vectorial representations.

Since the DBTs had some numerical representations already, for DBTs at least, the task turning those into a form where they had a numerical encoding was off to a good start. In case of the LTs, the numerical representations one could come up with would be non-ideal in that trying to represent them sequentially even though they have multiple nested structures with function applications could hardly do these representations justice. The problem of representing LTs with an encoding method or with feature vectors is that it is hard to tell what features they actually do have given that they are non-linear.

In the remaining subsections, before going into the representational framework that was used in this thesis, a survey of the encoding strategies that were explored and a literature survey of some of the attempts to linearly represent compositionality and lambda calculus are laid out. The justifications for why none of these surveyed methods were used in this thesis are given as well.

3.2.2.1 Gödel Encoding

Since the purpose was to turn the data sets into ones with numerical representations that would enable the usage of one of the widespread dimensionality reduction techniques, encoding the data with Gödel numbering (Gödel, 1931) was one of the techniques that was explored. Gödel numbering is an encoding method that can be used for the expressions of a formal language. Each symbol in the language is assigned its own natural number. So, when encoding a sequence of symbols, each symbol in the sequence is replaced with the natural number that was previously assigned to it. The resulting encoding is a sequence of numbers that is unique to a particular sequence of symbols (Gödel, 1931). Also, Gödel used prime factorization for integers, which is relevant here because the DBTs contain DB indices.

The problem with using Gödel encoding for this data was that it was not a technique that could give the uniform representations that were needed (i.e. representations that are number sequences of same length). Even the slightest change from one lambda term to another resulted in a whole different Gödel number. LTs and DBTs in the data sets have different combinations of concept words, varying numbers of lambda symbols and variable names/DB indices. So, when Gödel numbering was applied to samples from the data sets, it would yield number sequences with lengths that significantly varied. These representations were also hard to make sense of, in that, just by looking at the resulting Gödel number, one could not tell much about the underlying LT or DBT.

While Gödel numbering makes it possible to assign unique numbers to each semantically significant unit that could possibly appear in these data sets (such as a lambda symbol or a specific concept word), the significant length variation in the resulting Gödel numbers was a discouraging factor. Another discouraging factor was that, even though there was the aforementioned opportunity to give unique abstractions to each type in the data, the arbitrary nature of these assignments would make it hard to trust the dimensionality reduction manipulations that were based on the resulting Gödel numbers. The previous point is especially valid when it comes to the number assignments to the concept words, which could enjoy less arbitrary and less misleading number assignments.

Take the example of these two word entries in the data sets which have an underlying common concept: 'vitamin′' and 'vitamin-time′'. If we were to assign them unique natural numbers, their representations would significantly differ. As long as these kinds of cases are tracked, the situation could be remedied by coming up with a consistent coding for the string they have in common, which is 'vitamin'. The real problem arises when one realises that there are subtler versions of the problem above such as the one-time appearance of the concept 'blankie′' and the concept 'blanket′' in the utterances in the data. Considering the subtle similarities of some concept word in the data, it is hard to decide how far one should go trying to control for these kinds of cases while using Gödel numbering given that if they are not controlled for, the arbitrary nature of the natural number assignments could make the number given to each as different as night and day.

Another issue that would arise with the use of Gödel numbering had to do with the De Bruijn indices. As stated earlier, the numerical representations in the De Bruijn indices were meaningful for the purposes here in that they provide simple numerical representations to denote where the variable bindings lie in the word entries. If the De Bruijn indices were to be used in

conjunction with the use of Gödel numbers for the rest of the DBTs that were not numerical, the resulting meaning representations would be inconsistent, barring the problem of Gödel numbers causing huge variations in the length of the entries. They would be inconsistent if the De Bruijn indices were kept as they are, because of systematic and simplistic nature of the De Bruijn indices as compared to the encodings Gödel numbering would bring for the concept words, where these indices represent exactly the features they aim to represent and where they have a small range of numbers they can take which are '1, 2, 3, 4, 5, 6' in the case of this specific data set of LTs, does not match the ever-increasing natural numbers that the Gödel numbering would make use of. Changing the values of these De Bruijn sequences was not desirable, but if they were changed, it would mean that prime factorization would be used on these digits, and this, again, would result in inconsistencies throughout the representations because of the discrepancy that would occur with the Gödel numbers that are generated for the concept words, versus for the De Bruijn sequence digits.

3.2.2.2 Compression Approach

Another approach that was considered during the exploration was making use of a compression method. What compression methods do is that they produce a way of representing the information that the data represents with its current form with a new representation that is of a smaller length than the original one, thereby reducing the redundancies in the data (Lelewer & Hirschberg, 1987). To see if a lossless compression (i.e. one without any information loss) for the LTs and DBTs was possible, one would have to compress and decompress them to see if the decompressed versions are the same as the original ones.

For the result of a lossless compression to be a good re-representation of the data entries, though, the result of the compression should be readable in the sense that they could be studied to make judgements about the kinds of semantic features the compressed versions consist of. Realize that, if a lossless compression could be done on the data and it results in a readable data structure; the aim of reducing dimensions could, in a sense, be achieved. This is because, as stated earlier, compression methods find a denser representation for the same information that is contained in the data by removing redundancies from it. Yet, since a lossless compression does not result in any information loss, it may not be the best method of reducing the dimensionality in the data by virtue of it not actually getting rid of some of the features in the data. In any case, a compression of some sort is still a good candidate for the explanation of the underlying mechanism that maybe involved in the main phenomena of interest where the children have to make sense of a lot of variables that are tied to the meaning of the words they encounter to successfully acquire a lexicon, regardless of whether or not this mechanism is to be defined as dimensionality reduction.

With these goals and standards in mind, the data sets were compressed using the Pickle library of Python (van Rossum, 2020). As expected, the results of this compression was not meant to be understood by human beings in that they did not turn each data point into a solely numerical encoding like the one that was sought out here or into a solely textual encoding containing remains from the original strings in the data which could at least be read and interpreted as to what they could mean.

3.2.2.3 Category Theoretical Approach

Clark (2013) describes that he had an aim that is similar to the one here, one of finding and working on the distributional properties of the semantic representations he used in the computational linguistics framework he had been using, which was Combinatory Categorial Grammar (Steedman, 1996). In Clark (2013), he describes the process by which one could find linear representations for semantic representations and in what ways they could be subjected to algebraic operations, building on the work in Coecke, Sadrzadeh, and Clark (2010) which he was a part of. This work (Clark, 2013) describes the process by which they constructed vector space rewrites for the semantic representations of words according to the words' corresponding syntactic representations, representations which were based on pregroup grammar (Lambek, 2008).

Lambek (2008) proposed a grammar formalism called pregroup grammar to create a system that overcomes the limitations of categorial grammars, which he found to be unsuitable for algebraic operations. This grammar formalism is built around Category Theory (see Eilenberg and Mac-Lane (1945)) in such a way that the semantic representations the system affords could be seen as vectorial representations by virtue of them being suitable for the application of linear algebraic operations such as summation or taking the tensor product of vectors (Clark, 2013).

Since the framework that was built in Clark (2013) and Coecke et al. (2010) depended on a very specialized grammar (Lambek, 2008) in that it was based on Category Theory, it did not fit the representational framework in this thesis which is based on lambda calculus and De Bruijn indices. A future goal might be to try and reconcile LT and DBT representations with a category theoretical approach to representing meanings.

3.2.2.4 Smolensky and the Tensor Product Representations

It has been Paul Smolenksy's goal to come up with a way of using connectionist methods to represent compositional structures and constituency relations (1990). He claimed in many of his works that vectorially representing symbolic structures would become an achievable goal with the use of what he called *Tensor Product Representations* (TPRs) (Smolensky, 1990). A type of symbolic structure that contains constituency relationships Smolensky claims could be vectorially represented with the TPR method is the tree structures. One way to go about applying the method involves looking at the nodes of the tree to assign each of them a representation of their location information with respect to the other nodes in the tree, for instance. Other kinds of information about the relations in the tree could be encoded as well.

While LTs and DBTs could be conceptualized and written as having a tree structure which would reflect the nested relations in the function applications, taking these structures as trees and encoding them based on various relationships between the nodes (such as their respective locations) would increase the dimensionality in the data sets rather than reduce it because of the richness of the kind of information Smolensky (1990) suggests one should encode, and rightfully so. However, one caveat of having such a rich representation for these data sets is that the representations would get exponentially bigger with every added lambda binder,

which would be a problem not only because this would mean that the resulting data sets could not have uniform vectors of same length, but also because they would be computationally taxing.

Another caveat of using TPRs in this thesis has to do with the aim the thesis set out to achieve by using dimensionality reduction, and is related to the point made above. If the representations of the LTs and DBTs are constructed such that they are too rich in information, finding the similarities in the representations could become harder and not easier because with richer and richer representations comes the risk of increasing redundancy and these redundancies could create significant noise in the data that would be misleading during computational analyses. This would be true especially after the application of a dimensionality reduction technique in that it would make it harder to address the kinds of semantic features that have the most significant contribution to the meaning of a given word entry because the features are now hard to recognize because the representations they come from do not bear any resemblance to the structure of the original LTs or DBTs. Since the goal in this thesis is to make inferences about a possible semantic dimensionality reduction mechanism that aids in the creation of a lexicon by the child during language acquisition based on the investigation of the reduced data sets, aforementioned caveats are not ones that could be tolerated.

Last but not least, selection of the relevant tree structure information to be represented and the actual process of encoding them would prove to be a challenging task in the LT and DBT data sets because they stand for more than ordinary compositions of words or syntactic constituency relations, they represent abstractions for function applications. If, in the future, these caveats seem more manageable, pursuing a TPR representation for LTs and DBTs could be worthwhile endeavor.

3.2.2.5 Piantadosi's Combinatory Logic Framework for Concept Representation

S. T. Piantadosi (2021) builds a novel framework for representing concept meanings that is based on the approach to meaning Conceptual Role Semantics (CRS) posits, and computationally implements it in a system called Churiso. The name reflects the process of the *ChurIso* model whereby it comes up with *Church Encoding* (Church, 1936, 1941) aided combinatory logic representations (Schönfinkel, 1967) for an input (in the form of a set of concepts) that are *isomorphic* to the symbolic structure in the input set. The type of inputs that are given ought to be a set of related concepts and the symbolic specifications of the base facts about the relationships between these concepts. The model takes into account the relationships these concepts have with each other, and comes up with isomorphic structures for each one of the concepts in a set that reflect their relation to one another, which are based entirely on combinators (Schönfinkel, 1967).

In the exploration phase of this thesis, the Python implementation of the framework which is made available to public [2], Pychuriso, was implemented using the input sets and base facts that are also made available, to try and see whether this computational framework could be used as a basis from which the lambda calculus based representations in the data set (Lambda terms and De Bruijn terms) could be transformed into isomorphic combinator structures.

[2] See (S. T. Piantadosi, 2016) to access the latest implementation.

Remember that the lambda term representations in the thesis data came from a lexicon that was created for use in parsing according to the Combinatory Categorial Grammar (Steedman, 1996) formalism where using these semantic representations in computation together with accompanying syntactic representations relied heavily on function applications and combinators. Also note that it is a long term goal of the research program behind this thesis to later combine the semantic dimensionality reduction framework that is created here with a coherent framework for syntactic dimensionality reduction, and to study language acquisition at the level of the syntax-semantics interface back within a CCG parsing context. Therefore, transforming the LTs and DBTs in the data into combinatory logic formulas using Churiso to replace the word meanings these terms denote was seen as a viable methodology for the exploration of the combinatory properties these lambda terms have that could be uncovered by a reduction to combinators. A reduction to combinatory logical forms could be seen as a dimensionality reduction in the sense that the resulting representations would potentially denote only the most important properties of the semantic representations.

The implementation efforts started with running the model for every one of the concept sets that were given in the data set (S. T. Piantadosi, 2016, 2021); trying out, in the process, alternate representation strategies for each set by using Pychuriso's feature where one can select the type of combinator or a set of combinators on which the model should base the combinatory formulas it outputs. In the process of getting to know the data provided by the implementation S. T. Piantadosi (2016) better, noticing varying levels of base fact complexity for certain data sets and how it factors into computations, and observing the speed at which the outputs for certain data sets could be returned, it became clear that the data sets this thesis was trying to explore could not, at this stage, be explored within the kind of representational framework that Churiso requires as inputs and it could not benefit from the representational framework that it returns in its outputs.

One reason for this contention, and perhaps the simplest one to resolve, is that even if the data sets this thesis explores could be given to Pychuriso in an appropriate format where the word representations are expressed in a way that are conceptually related to each other in the form of base facts the model will use to come up with corresponding combinatory formulas (which is already unlikely for reasons that will be given later), it was clear that, in the implementation's current version, the computational resources would not be sufficient for this task as the two data sets were too big and the relationships in them were too complex yet disparate compared to the data sets Pychuriso could work with.

An important and impressive feature of Churiso which would certainly have advantages in tasks other than the one at hand but would work more like a bug when it comes to reducing the dimensionality in the representations was that it returned as outputs, for most of the cases and for most of the combinatory restrictions, around a thousand different ways in which a combinatorial isomorphism could be found for a particular concept set. This, barring the fact that the data sets in this thesis could not even be inputted to Churiso, would not be a desirable way of going about investigating the dimensionality in the data sets. If anything, this feature of Churiso could be used as an enumeration strategy for laying out unique sets of features the semantic representations in the data sets (LTs and DBTs) could be conceived of as having. Yet, note that it could also be the case that Churiso could not enumerate as many possibilities for this specific data set because the data set is large and it would be hard to find one, let alone more than one isomorphic combinatory structures for the relationships in the data set.

Therefore, the presentation of this feature of the framework may be unfounded if the data set could be given to Churiso in the way that is needed by the implementation.

The task of having to decide on an exact combinatorial representation option (or maybe on a handful of options) that would reflect the true nature of the semantic representations with their combinatory properties, would in fact become a dimensionality reduction problem that is even more severe than the one that was present at the start of this thesis work. The dimensionality problem would be exacerbated and not be reduced in this paradigm. The structures in the lambda-based formalisms (representations with lambda terms and De Bruijn terms) provide more specified information that would help individuate the combinatory properties the word that is being denoted would have in sentential contexts, these structures are therefore lower in dimensionality, or they can be said to be closer to the true dimensionality of a specific word meaning phenomena the child's mind relatively effortlessly captures during the acquisition of language. Another aspect of the lambda-based structural formalism in the data sets that makes them lower in dimensionality by these standards is that these representations have in them as atoms the constituent concepts which further help individuate the meaning of particular structures.

If all of these meaning representations are put in a *like Combinatory Logic* (S. T. Piantadosi, 2021) like in Churiso all at once, all referring to and constrained by the same set of base facts (and in a theoretically equivalent way if all of the contents of one's lexicon are represented with this kind of Combinatory Logic all at once, being tied to each other in the ways specified in the base facts about their conceptual roles), it would not matter if they became unidentifiable from a symbolic perspective in the sense that these are structures S. T. Piantadosi (2021) proposes that could be similar to the subsymbolic/neural structures of representation in the brain. Them becoming unidentifiable would not matter from a subsymbolic level of explanation, however, only with the constraint that they are set up in ways all of the lexicon entries are tied to each other, because that would be the only way in this paradigm the combinatory logic would be computationally restricted and forced to come up with a unique formulaic representation for each type of unique semantic representation.

If however, the specifications of base facts and therefore of the words' conceptual roles are constrained to partitions in one's lexicon and are instantiated only with reference to a network of related conceptual roles, the kind of *like Combinatory Logic* (S. T. Piantadosi, 2021) in S. T. Piantadosi (2021) would not be guaranteed to have unique representations for any particular semantic representation, and the combinatory logic would lose its individuating properties in the subsymbolic level as well. There, however, should be unique representations that separate concepts, regardless of the level of analysis being symbolic or subsymbolic. This uniqueness challenge is already acknowledged by S. T. Piantadosi (2021) and also somewhat addressed by them taking 3 different conceptual sets and computing the combinatory representations for them by posing overall uniqueness constraints on all. The reason why these challenges are outlined here, however, at least at first glance, has to do with the nature of the problem being one of dimensionality reduction in this thesis and with the size and the complexity of the data sets.

The dimensionality and uniqueness concerns above inevitably lead to the real issue with implementing Churiso to study, in this particular case, the dimensionality and features in lambda calculus based meaning representations, and it can be summarized with the question of "How

does one go about tying these meanings to each other if they all have to be tied together to get unique representations for each and if we do not know of any unifying base facts that would tie them to one another?" Notice how the broadness of the knowledge of concepts that is required to answer this question is probably not so different than that of the knowledge that is demanded of a satisfactory theory of semantic dimensionality. However, there is a marked difference between the kinds of answers that each of them demand, and it is in their conceptions of dimensionality or features.

The dimensionality/features that would satisfactorily create the base cases for the former question are being defined over atomic concepts baring some inferential relationship to each other in the conceptual ontology, while the notion of dimensionality that is required in the latter case (and for this thesis) is one that can be thought of as reducing structural dimensionality to a handful of rich structural features such that they lead to a unique semantic representation for a word which still allows for similarity judgements of words, albeit only based on their compositional properties until or unless a mapping to the real world is later added. This second sense of dimensionality does not require base facts about the conceptual relationships one can observe in the world, it requires a linguistic theory that describes the rules over which any relation in the world could be described in the first place, whether in thought or in language. Creating, on the other hand, for each of the LT and DBT representations, one large representation of base facts is a task that could be achieved by creating an ontology of conceptual relations, and it is not in the scope of this thesis.

3.3 Final Decision on the Representational Framework

This section gives a detailed account of the encoding method that was used to create three different vectorial representations for the data set, and specific examples of how the data set for the DBTs looked like when encoded in this way. Given that the methods and approaches surveyed beforehand did not fit various criteria regarding the aims of this thesis, a novel representational scheme was used to represent the DBTs, which involves coming up with feature vectors for them based on elementary assumptions about the features that are present in the entries in the data sets. In the first subsection, the representational scheme that is followed to create feature vectors of the same length across the entries of the DBTs data set is described in detailed.

Since there was no sophisticated way of representing the bindings in the LTs and since the decision was to come up with simple vectors for them was motivated by the idea to compare them to the feature vectors of the DBTs, they were represented by using the bag of words (BoW) technique involving each type of token that could occur in a given lambda term. The second subsection briefly describes this process of vectorization for the LTs. A third version of the data, which was created by way of modifying the version for the DBTs, is introduced in the last subsection.

Whenever a tokenizer was used in the encoding that are described below, the implementation in this thesis used NLTK's White Space Tokenizer (Bird, Klein, & Loper, 2009) and Scikit-learn Feature Extraction's CountVectorizer tool (Pedregosa et al., 2011).

3.3.1 Vectorization of De Bruijn Indexed Lambda Terms

In the following subsections, one can see the assumptions that were made in terms of the features that are present in the DBTs, and the way in which they were represented. The first feature is the number of lambdas in a DBT, the second one is the concept words that are present in a DBT, and the third one is the De Bruijn index sequence in a DBT.

3.3.1.1 The Number of Lambdas as Features

Each data point in the DBT set is a lambda term even though they may not have any lambdas in them, and they may have 5 lambdas or 5 bindings at most. So, the number of lambdas could be taken as a feature that would have some specific value for each of the word entries. This feature representation task could be approached by adopting a thermometer encoding method where with each increase in the number of lambdas, one more 0 turns into 1 in the encoding, starting with [1,0,0,0,0,0] for 0 lambdas and ending with [1,1,1,1,1,1] for 5 lambdas.

See (7) for the thermometer encoding convention that is used for the number of lambdas which will be the first feature in the DBT vectors.

(7) .

$$\lambda = 0 \rightarrow 100000$$
$$\lambda = 1 \rightarrow 110000$$
$$\lambda = 2 \rightarrow 111000$$
$$\lambda = 3 \rightarrow 111100$$
$$\lambda = 4 \rightarrow 111110$$
$$\lambda = 5 \rightarrow 111111$$

The reasoning for the choice of thermometer style encoding is that while it is known that there is a hierarchical relationship between say, a lambda term with 0 lambdas and one with 3 lambdas, the degree of this relationship is unknown and unquantifiable. Therefore, the thermometer representation is the best encoding technique for reflecting this hierarchy between the terms with differing numbers of lambdas.

3.3.1.2 Concept Words as Features

It was harder to choose a feature representation method when it came to the concept words in the data, and the final decision was to go with a Bag of Words approach. Since there are 1040 concept words in total, these vectors were bound to be sparse, which was not very desirable.

It was clear that such an approach could in no way be an ideal representation for DBTs because with this technique the order of the concept words could not be preserved and the concept words were separated from the De Bruijn indices that in some word entries surround the concept words. However, this technique was the only one that was feasible. A word embedding approach could not be pursued because a significant number of the concept words in the data were not words that one could find an embedding for since they were semantic annotations for tense, aspect (such as 'pre' and 'simp' in the examples above) and many others.

Also note that the nature of the BoW technique which lead in the isolation of the De Bruijn sequence digits from the concept words whenever they occur in between two distinct concept words does not usually present a problem for this data set because the concept words the digits could appear between are usually for the tense or the aspect of the overall word representation.

3.3.1.3 De Bruijn Indices as Features

De Bruijn indices were included in the feature vectors without making any transformations on them since they were as informative as they could get and they were already numerical. However, these indices were padded with zeroes until their array reached a certain length (of 6 digits) to mirror the length of the array for the number of lambdas. A slight problem with this feature representation was posed by the cases where these indices did not continuously follow each other but were intertwined with concept words, in which case they were isolated from them.

To understand how all these three representations were brought together, see a template of the resulting feature vectors for the DBTs in (7).

(7)

[<#-of-lambdas thermometer> <bag-of-words> <de-bruijn indices padded with zeroes>]

See (8) and (9) as examples of the encoding method with two cases from the opposite ends of the spectrum, first one a DBT with 0 lambdas and the second a DBT with 5 lambdas. Note that, for the sake of brevity, the sparse vectors that would result from the BoW encoding are shown inside the symbols '< .. >' and are shortened with the use of '...' which is used to suggest the possible continuation of the digit patterns with zeroes. Also note that the lambda symbol is denoted by the 'LAM' token in these examples, and also in the actual data set which was used in the analyses that will be laid out in Chapter 4.

(8) **Word:** Eve
De Bruijn Term: EVE
Vectorial Representation:

[1 0 0 0 0 0 < 000...[1 instead of 0 wherever the location in the BoW dictionary is of the concept word EVE]...000 > 0 0 0 0 0 0]

(9) **Word:** Think
De Bruijn Term: (LAM (LAM (LAM (LAM (LAM (((((SIMP 2) THINK) 4) (5 1)) 3))))))
Vectorial Representation:
[1 1 1 1 1 1 < 000...[1 instead of 0 wherever the location in the BoW dictionary is of the concept word THINK]...000...[1 instead of 0 wherever the location in the BoW dictionary is of the concept word SIMP]...000 > 2 4 5 1 3 0]

The De Bruijn Vectors (DBVs) were of the length 1052, with 6 of them representing the lambda array, 1040 of them representing the BoW for the concept words, and with 6 of them representing each of the De Bruijn sequences.

3.3.2 Vectorization of the Original Lambda Terms

The original lambda terms were turned into sparse feature vectors with the use of the BoW method. The dictionary that was used to build these vectors took into account the array of length 6 for 6 lambda binders (e.g. "lambda x"), 1040 concept words (e.g. "THINK") and variable names such as "x" (5 of them because one of those variable names also stood for a concept word which is included in the array of 1040 just described). The length of the lambda vectors was therefore 1051. Note that the vectors that were created for the original lambda terms are from now on called lambda vectors (LVs).

3.4 Summary of the Data Sets To Be Analyzed

The vectors for the DBTs (which will be called De Bruijn Vectors or DBVs) were used to create a data set of with different kinds of DBVs where the only difference is that these new representations will not have the feature array for the lambda terms. In other words, this new version of DB vectors contain only the concept word array concatenated with the array for DB indices. This new data set is called Bare DB Vectors or BDBVs for short, which is of length 1046. Note that the arrays for the DB indices parts of the DBVs were created by taking the DB indices and adding zeroes until the array length reached to 6 so as to match the length of the arrays for the lambda binders. Even though the BDBVs do not have lambda binder arrays, the added zeroes in the DB arrays were not removed when creating the BDBVs because this convention had the benefit of keeping the vectors in the BDBVs data set the same length.

The next chapter lays out the analyses that were conducted for each of the 3 data sets, and for various subsets in the DBVs data set such as verb-only subsets or subsets with verbs with exactly 2 or exactly 3 lambdas.

3.4.0.1 A Step by Step Demonstration of the Type of Vectors the 3 Distinct Data Sets Would Have

See how the utterance below is represented as a logical form, and the linguistic properties of the word "making", along with 3 distinct vectorial representation possibilities for the word that is based on lambda calculus and De Bruijn indices.

Utterance: Mommy is making coffee.
LF: ((((”CONT” ”PRT”) ”MAKE”) ”COFFEE”) ”MOMMY”))
Word: ’making’
PoS Tag: tving>
Syntactic Category: S[type=inf]\NP\AUX[type=be]\VNP
Lambda Term: λx.λy.λz.cont′ y make′ x z
De Bruijn Term: (LAM (LAM (LAM ((((CONT 2) MAKE) 3) 1))))
DB Vector: of Length 1052
[1 1 1 1 0 0 <000...[1 for CONT]0000[1 for MAKE]...000> 2 3 1 0 0 0]
Bare DB Vector: of Length 1046
[<000...[1 for CONT]000[1 for MAKE]...000> 2 3 1 0 0 0]
Lambda Vector: of Length 1051
[<000...[1 for CONT]000..[1 for MAKE]..1[for λx]...000>]

CHAPTER 4

DATA ANALYSIS AND RESULTS

The data exploration process, the results of the data analyses and dimensionality comparisons for 3 different versions of the data that were specified in Chapter 3 are reported in this chapter.

The methods that were used in this exploration are unsupervised machine learning techniques that are frequently used to better understand the trends in data sets. One such method is the use of dimensionality reduction for feature extraction, which is employed in the analyses here both in terms of a systematic look at the reduction possibilities for the dimensionality in the data sets and by visualizing the said data in 2-D and 3-D by making even further reductions to the dimensionalities when necessary. These visualizations are used to make sense of the data sets and the dimensionality they have. Another type of unsupervised machine learning that was employed in the analyses is the use of a clustering method to find patterns and groups in the data. With the goal of gaining specific insights regarding specific lexical semantic representations, the word entries within each data set were also evaluated and compared based on the cosine similarities of the vectors that represent them.

The exploration of the dimensionality in the 3 data sets started with the evaluation of the features that could be extracted from the data sets using Principal Component Analysis (PCA) (Pearson, 1901; Hotelling, 1936), a linear dimensionality reduction technique. After reporting the insights from the PCA of the 3 data sets, the analyses continue with the application of t-SNE, a non-linear dimensionality reduction technique that is frequently used for data visualization (van der Maaten & Hinton, 2008). After reporting the results of the t-SNE visualizations, the report of the analyses continue with the discussion of some of the insights that were gained from the data sets and their representations with regards to the within-data representation similarity scores based on cosine similarity (Huang, 2008). The last section of the analysis reports the results of the use of the Affinity Propagation clustering algorithm (Frey & Dueck, 2005) for certain data sets and their subgroups based on the number of arguments the verbs take.

4.1 Dimensionality Reduction

The curse of dimensionality (a term coined by Bellman (1966)) refers to the situations where the amount of the variables that seem to mark the observations (the features/vari-

ables) about a phenomena, in other words, the dimensionality in the phenomena, is an amount which is hard to handle when analyzing the data, especially in situations where the amount of observations about the phenomena (the number of individual data points) is not a desirably sufficient amount for data analysis to begin with. The notion of the curse of dimensionality captures the nature of the problem children successfully overcome in acquiring their language, since they are exposed to a lot of dimensionality (i.e. to a lot of features that could possibly be important to acquire the meanings of words) and yet they are certainly not exposed to a lot of data that would serve as good examples of various semantic phenomena when compared to the amount that would seem sufficient from a machine learning perspective. The children overcoming this so-called curse of dimensionality is the problem this thesis tries to model and make sense of with the analyses that are reported in this chapter. The data set obtained from the child-directed utterance data of Eve (Brown, 1973) reflects the dimensionality problem well in that while there are only 4052 data points, each of those points are marked by more than 1000 dimensions in each 3 versions of the data set. The working assumption, as it is the case when setting out to solve dimensionality reduction problems, is that the underlying dimensionality in these data sets should be marked by a handful of dimensions only and that these dimensions could, in principle, be discovered via dimensionality reduction methods.

The application of dimensionality reduction (DR) techniques to the data was done with two purposes in mind.

- To compare the dimensionalities of the 3 data sets to gain intuition about whether or not the distinct structure in the representational scheme of the data set is more conducive to having efficiently reduced dimensionality and therefore to providing a better model for the semantic dimensionality and its reduction during language acquisition (dimensionality that is reduced with PCA)

- To investigate the features of the 3 distinct representations after having reduced their dimensionality to a reasonably low dimensionality with minimal loss in feature information (the first reduction with PCA) by way of visualizing the features that were extracted from the data sets with another dimensionality reduction technique (dimensionality that is further reduced and visualized with t-distributed stochastic neighbor embedding (van der Maaten & Hinton, 2008))

4.1.1 Dimensionality Reduction with Principal Component Analysis

A dimensionality reduction technique that is called Principal Component Analysis (PCA) was used on the 3 data sets, which is a technique that was independently discovered and introduced by Pearson (1901) and Hotelling (1936), and was named by Hotelling (1936). PCA is a linear technique for reducing dimensionality in data to a set number of components that would explain most of the variation in the data (Jolliffe, 2002). It is a linear DR technique because it involves the application of a linear function to the original features to combine them to make up the principal components, which involves linear projections of the original features (Jolliffe, 2002) to lower dimensions. The DR

that is done with PCA in this thesis is an unsupervised process (compared, for example, to Linear Discriminant Analysis which is a supervised linear dimensionality technique) because there was no employment of class information that would come from doing a labeling for the word entries that could be done based on the corresponding surface forms of the words or their PoS tags, which could guide the process (see (Chao, Luo, & Ding, 2019) for an overview of supervised versus unsupervised DR).

The reason why a supervised approach to DR was not taken is because imposing supervision with the kinds of labels that could possibly be provided (for parameters like a BoW encoding of a surface word form or a corresponding PoS tag) was not seen as ideal considering this work's aim of making sense of language acquisition in the light a hypothesis that lines up with a semantic bootstrapping approach where the child is either yet to learn certain parameters that lead to the classification of words in a certain way (which is a milestone that in principal would be aided by the learning and reduction of semantic dimensionality) or where the surface word forms are too noisy to be considered as informative labels for semantic representations because the same form or similar forms could stand for representations that are vastly different.

While there are non-linear dimensionality reduction techniques in the literature that could be potentially be used that operate with the underlying assumption that the data is marked by non-linear relationships (also called Manifold Learning techniques (Chao et al., 2019)), at least initially, a linear DR approach was intentionally sought for the analyses in this thesis, especially when comparing the 3 versions of the representations. The reason for this is that, as could be seen in the exploration process depicted in Chapter 3, the methodological approach to the representation of semantics and its dimensionality in this thesis is one that tries to linearize compositional structures that robustly represent meaning (i.e. the representations with lambda calculus) and one that explores whether or not certain versions of these structural representations could already be capturing the compositionality in a linear way (i.e. the representations with De Bruijn indices). Therefore, the use of PCA (an otherwise powerful DR technique the use of which is seen as problematic when it comes to the cases with non-linearity in data) was seen as a good approach to bring forth and evaluate the effects of linearity/non-linearity assumptions for the data sets.

Rather than reducing the dimensionality in the data sets to any set number of components, PCA was first used to explore the dimensionality in 3 distinct data sets to see the maximum of the extent to which their dimensions could be reduced by deciding on a trade-off of the least amount of principal components that would explain as much of the variance within these data sets as possible. The word representation entries in all three of the data sets are represented with more than a thousand features or dimensions. The goal, with using PCA, was to find *components* in the original features/dimensions such that they reflect most of the features in the data set that reliably and consistently affect the outcome of the representations (i.e. the semantics of the words) independently from any other feature's individual affect. In other words, the goal here was to extract from the data set highly representative features in order to gain insights about the patterns in the semantic representations by accentuating the most informative ones and getting rid of noise, which could then be used to find patterns that predict certain outcomes in terms of leading a semantic representation to have certain combinatorial properties (such as

revealing feature patterns that allow verbs to be grouped based on the number of arguments they can take). Note that this technique does not select a set of features from the existing ones, it rather makes up or extracts new features from the existing feature set.

The process of the exploration of the dimensionalities, the selection of the number of components to reduce these dimensionalities to, and the evaluation of the extracted features using the PCA technique are reported in the upcoming subsections for each respective data set. Note that in these analysis of the components, the data in the data sets was not scaled, so no standardization was done on them. The decision to not to scale the data was based on the observations that:

- The scaled versions of each of the 3 data sets (when standardized using Sklearn Preprocessing's StandardScaler which scales the features by subtracting the mean from each and dividing them with the standard deviation (Pedregosa et al., 2011).) required significantly higher numbers of components to keep a decent level of explanation in variance intact compared to the unscaled versions, leading to the contention that scaling the data sets results in the loss of valuable information.

- Intricate knowledge of the encodings in the data, about the encodings being mostly binary with sparse BoW arrays with the only difference emerging in the ones with De Bruijn indices that vary within a very small scale of numbers (1-6), resulted in the idea that the unscaled versions of the data sets are exactly as informative as it is needed from them and are therefore would be unit-wise proportionate with regards to the variance in their encoding.

- Normalizing the features to have values that lie somewhere between 0 and 1 was also not desirable because of the sparsity in the data and because it was better in the case of the De Bruijn Vectors and Bare De Bruijn Vectors if the features with values that are greater than 1 indeed dominated the components to be found with PCA.

The implementation of the PCA that was used in the following analyses were done in Python 3.8.12 (van Rossum, 2020) with the help of the PCA tool in Scikit-learn Decomposition (Pedregosa et al., 2011).

4.1.1.1 De Bruijn Vectors and PCA

De Bruijn Vectors (DBVs) is a data set of 4052 entries, each of them consisting of 1052 features that were selected and encoded according to the process that was described in detail in Chapter 3, which involved thermometer encoding for the number of lambdas, BoW encoding for the concept words, and the use of additional arrays for the De Bruijn indices that were padded with zero until the array length reached 6 digits for each data point. The first step in the exploration of this data set was to get an overview of relationship between the number of principal components to select for and the amount of variance in the data that would be explained as the number of components are increased; which yielded the graph in Figure 1 for DBVs.

The plots of the cumulative explained variance (as shown in Figure 1) for a certain number of principal components (also called eigenvalues) are shown throughout the PCA exploration of the 3 data sets. These eigenvalues, or principal components, are based on the eigenvectors of the data covariance matrix, and in the implementation that was employed here, they were computed using the singular value decomposition of the data matrix (Pedregosa et al., 2011).

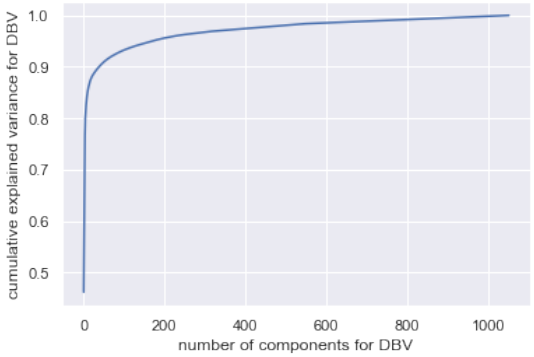

Figure 1. The plot showing the number of components that would be needed to explain more and more of the variation in the De Bruijn Vectors Data Set

As it can be seen from the plot above, the dimensionality needs to be reduced to a number that is close to the original number of 1038 dimensions to account for almost all of the variation in the data (99 %). In Table 1, one can see some of these statistics in Figure 1 more clearly.

Table 1

The Ratio of Explained Variance per the Number of Principal Components for the DBVs

Number of Components	% of Variance Explained
739	99%
170	95%
40	90%
11	85%
5	80%
4	75%

Looking at Table 1, one can get the sense that the trade-off between the reduction in the dimensions and the reduction in the ability of the reduced dimensions to account for the amount of variance in the data would be the most acceptable if one was to select for a reduction to 40 components that would account for 90% of the variance in this data set.

In Principal Component Analysis, the very first principal component the linear function yields from the original features in the data set is always the component that captures most of the variance in the data and the numerical value of this component is as high as

the percentage of variance in the data it captures. The amount of variance the second component reflects is the second highest in value and this trend goes on until the last principal component, with each new component reflecting a smaller amount of variance. Most importantly, the values of the components reflect the amount of variance each of them capture independently from other components.

As it can be seen in Table 2 under the independent effects of the components, the first principal component captures 3.49% of the variance in the DBVs, followed by the second, third and the forth principal components capturing respectively 0.84, 0.79 and 0.66 percent of the variance in the data, completely independently from the variance they would predict in their interactions with the other principal components.

Table 2

First 10 Principal Components and The Variance They Independently and Collectively Capture in the DBVs

Independent Effects of the Components	% of the Cumulative Effect
3.4903	46.16%
0.8450	11.17%
0.7928	10.48%
0.6622	8.76%
0.2621	3.47%
0.1044	1.38%
0.0860	1.14%
0.0667	0.88%
0.0568	0.75%
0.0448	0.59%

The cumulative effect of the number of components that takes into account the interactive effects of the components with each added component can be seen in Table 2 as well. Note how the sum of the percentages for the collective effects of the first 5 components make up 80%, exactly the amount of variance that would be explained with 5 components according to the statistics in Table 1.

Before going forward with this dimensionality reduction and visualization of the components, see how these statistics play out for the other two data sets.

4.1.1.2 Bare De Bruijn Vectors and PCA

Bare De Bruijn Vectors (BDBVs) data set is a version of DBV with the features denoting the number of lambda terms dropped (so, with 6 features removed). Just like DBVs, it has 4052 individual data points, in this case with each data point consisting of 1046 features.

See Figure 2 for a plot of the cumulation of the ratio of explained variance in the BDBVs as the number of components used in dimensionality reduction increase. This plot is

pretty similar to the graph of DBVs, which is not surprising given the similarities of their feature vectors. Their differences are worth looking into while trying to uncover more and more information about their principal components given that one of the purposes of the analyses is to compare these two representations that are based on De Bruijn indexing to see whether or not there really is a need to encode the lambda symbols into the semantic feature vectors even though they are normally a component of De Bruijn indexed lambda terms.

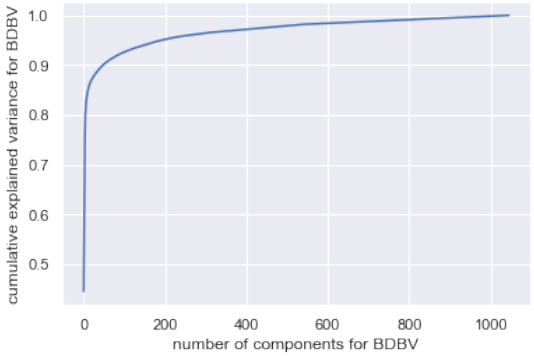

Figure 2. The plot showing the number of components that would be needed to explain more and more of the variation in the Bare De Bruijn Vectors Data Set

In the following table, one can see the number of components that the analysis yielded for a select number of ratios of the amount of explained variance for the BDBVs data.

Table 3

The Ratio of Explained Variance per the Number of Principal Components for the BDBVs

Number of Components	% of Variance Explained
765	99%
192	95%
49	90%
12	85%
5	80%
4	75%

These components should be explained further to make any judgements about the significance in the differences that the comparison of BDBVs and DBVs yield in terms of the components needed at different explanatory levels. It seems that the best combination of the number of components per the amount of explained variance will, in the case of the BDBVs data, be a reduction to 49 components with 90% of the variance in data explained.

See Table 4 for the first 10 principal components, which are also the most influential in their caption of the variance of the BDBVs data, with their influence decreasing in every

next component. The first principal component independently explains approximately 3.03% of the variation in this data, followed by the second, third, and the forth principal components also independently capturing 0.78, 0.76, and 0.60% of the variance in the data. The cumulation of the effects of each added component on the ratio of explained variance can also be seen in Table 4.

Table 4

First 10 Principal Components and The Variance They Independently and Collectively Capture in the BDBVs

Independent Effects of the Components	% of the Cumulative Effect
3.0301	44.51%
0.7841	11.52%
0.7614	11.19%
0.6087	8.94%
0.2581	3.79%
0.1008	1.48%
0.0772	1.113%
0.0517	0.76%
0.0377	0.55%
0.0334	0.49%

Further analyses regarding these comparisons and the actual DR in this data will be reported after the following section on the exploration of the principal components of the Lambda Vectors that are based on the original lambda terms.

4.1.1.3 Lambda Vectors and PCA

The Lambda Vectors (LVs) data set consists of the encodings of the original 4052 lambda terms, which were vectorized based on the unique tokens in them such as unique lambda binders (e.g. λx) and concept words (e.g. PL′). The vectorization yielded unique BoW feature vectors with a set length of 1051 (1051 features) for these 4052 lambda terms.

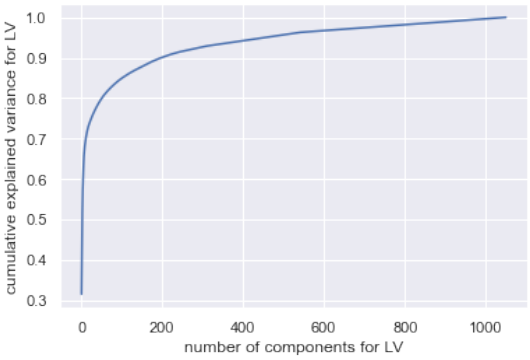

Figure 3. The plot showing the number of components that would be needed to explain more and more of the variation in the Lambda Vectors Data Set

The plot above is a graph of the number of components that would be needed if more of the variance in this data of LVs was to be explained after DR with PCA. As it can be understood from the trend in Figure 3, optimally informative dimensionality reductions of this data set seem to require significantly more number of components as compared to both DBVs and BDBVs data sets, especially when it comes to lower and lower ratios of explained variance. The following table makes explicit some of the 'ratio explained' to 'number of components needed' correspondences in this data set.

Table 5

The Ratio of Explained Variance per the Number of Principal Components for the LVs

Number of Components	% of Variance Explained
909	99%
453	95%
198	90%
102	85%
52	80%
25	75%

Note how, even with the 75% of the variance explained, the number of components could not be reduced to a single digit number for the LVs data set, as opposed to the DBVs and BDBVs data sets where 5 components were enough to account for 80% of the variance in the respective data sets. In keeping with the selection of the number of components to reduce data to for the DBVs and BDBVs, which was to keep the explanation of variance at 90%, the number of components the LVs will be reduced to was determined to be 198.

Table 6 above shows the variance that is independently accounted for by the first 10 principal components for the LVs. The first principal component accounts for 1.05% of the variance, with the second and third components capturing 0.38 and 0.29 percent of the variance respectively.

Further analyses to compare the reduced versions of the 3 data sets are reported in the upcoming section. It seems clear from the principal component analyses so far that the representations with the De Bruijn indices are more robust than the ones with regular lambda terms in terms of their linear representations being more conducive to be represented with lower dimensionality while retaining low levels of information loss. At 90 % explained variance, the DBVs and the BDBVs respectively need 40 and 49 components, which goes to show that a decent amount of reduction is possible for these two data sets. It is also interesting how the number of components needed significantly increases if one wants more of the variance in these data sets to be explained. Although it seems at first glance that the data set with the De Bruijn indexed vectors that contains an array representing the number of lambdas (DBVs data set) has a representation that

Table 6

First 10 Principal Components and The Variance They Independently and Collectively Capture in the LVs

Independent Effects of the Components	% of the Cumulative Effect
1.0565	31.56%
0.3897	11.64%
0.2919	8.72%
0.1870	5.59%
0.1087	3.25%
0.0899	2.68%
0.0780	2.33%
0.0460	1.38%
0.0332	0.99%
0.0268	0.80%

slightly increases the data set's linear reducibility to smaller numbers of components (40 components as compared to 49 components for the BDBVs) compared to the representation without the array for the number of lambda symbols, more exploration is needed to support this contention.

Also note that, the individual values that were to be found for a select number of principal components that all 3 of these data sets could be reduced to are not useful in and of themselves to tell anything about the nature of the reduced versions of the original features that are made up of a select number of new features the PCA finds for each data point, because the linear function that is applied to the original features makes up new features that do not bear any resemblance to the original features. However, the results of the PCA are still useful because they are informative about the dimensionality in the overall data sets and the reduction possibilities for this dimensionality.

4.1.2 Visualizing the Principal Components with t-SNE

In the previous section, the exploration of the dimensionality in the 3 data sets that was done using PCA was reported along with the decision on the optimal number of components the dimensionalities in the data sets should be reduced to (which is 40 components for the DBVs, 49 components for BDBVs and 198 components for LVs), and the rationale behind this decision (so that 90% of the variation within each data set will be accounted for). As expected, after making the trade-off and reducing the dimensions based on this decision, the dimensionality in each of the data sets was still too high to allow for any meaningful exploration and it was especially too high for further exploration that could be aided by visualization (because there were still much more than 2 or 3 dimensions in the data sets). While it would be possible to plot the 3 data sets by reducing each of them to have as features 2 or 3 principal components such that there would be 2-D and 3-D visualizations for each of them, doing this reduction was not

meaningful since for each of the data sets, taking 2 or 3 components would mean that a significant portion of the variance in these data sets would not be accounted for.

With the goal of gaining more insights into the reduced versions of the data sets (the ones reduced to the PCA components reported in the previous chapter) by way of meaningful visualizations and also continuing with the original aim of reducing the dimensions that were north of 1000 for each of the 3 data sets, the t-distributed Stochastic Neighbor Embedding (t-SNE) method was employed. It was introduced by van der Maaten and Hinton (2008) as a non-linear dimensionality reduction algorithm that is also useful for the visualization of the reduced versions of the data. It is a manifold learning algorithm in that it reduces the data set's dimensionality from an assumed high dimensional manifold the original data lies to a manifold that is lower in dimensionality (Chao et al., 2019), and the assumption about the data instances making up a manifold is why this technique operates under an assumption of non-linearity.

PCA and t-SNE worked to compliment each other in this thesis because the use of t-SNE by itself would not work efficiently with the high number of dimensionality in the data sets at the start (each having data points that consist of 1000 and more dimensions), so it would need preprocessing with another dimensionality technique to reduce the dimensionality to around a maximum of 50 dimensions. This condition was already met here with the use of PCA previously in the analyses.

The use of PCA, a linear dimensionality reduction technique, was previously justified on the basis that it was seen as a good way to test the linearity assumptions about the data sets with the De Bruijn indices, and its use has proven to be somewhat affirmative with regards to these assumptions in that the principal components for the data set that did not utilize the De Bruijn indices in its vectorial representations (the LVs) was less robustly representative of the features in the data and that this data set's DR with PCA required significantly more components for the same ratios in the amount of explained variance compared to the other two data sets. However, with the additional use of t-SNE, which is a non-linear technique, the exploration takes a turn in that the assumption about the dimensionality becomes such that non-linearity is assumed to mark the true dimensionality in these PCA-reduced forms of the data sets.

This analysis was done in Python 3.8.12 (van Rossum, 2020) with the help of the t-SNE tool in Scikit-learn Manifold (Pedregosa et al., 2011) and the results were visualized with the help of Matplotlib (Hunter, 2007). The implementation that is used here takes the pair-wise similarities of the data points which are calculated using Euclidean distance, and converts these similarities to joint probabilities by assuming a Gaussian distribution in the overall variance of the data (van der Maaten & Hinton, 2008; Pedregosa et al., 2011). The algorithm finds a lower dimensional manifold for the data set by trying to minimize the Kullback-Leibler divergence between the joint probabilities that were calculated for the original high dimensional data and the joint probabilities possible lower dimensional manifolds would have, and selects the lower dimensional manifold that minimizes this divergence (Pedregosa et al., 2011).

4.1.2.1 t-SNE Visualization for DBVs

4052 data entries of DBVs with 40 components as their new features were visualized using t-SNE with dimensionality reduction to 3 components. The visualizations were plotted to 2-D (see Figure 4) and 3-D (see Figure 5).

The 2-D and 3-D visualizations for the versions of the DBVs with 40 extracted features point to some clusters that these features can capture, which account for 90% of the variance in the data. The closer any two data points are in the 2-D and 3-D visualizations, the smaller variance they represent between those two data points. An important point to be noted is that all of the figures for the t-SNE visualizations that are shown in this work for the 3 data sets do not have a color scheme because the use of t-SNE in each of these cases was unsupervised, meaning that no information about categories each data point may fall under was not given to the algorithm. Therefore, the algorithm takes every data point as an independent from the others, showing each of them with the same color.

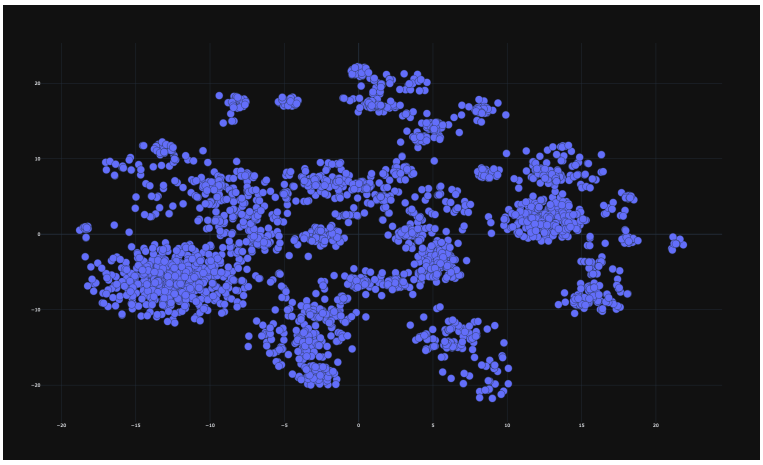

Figure 4. A 2-D plot showing the t-SNE visualization for PCA-reduced DBVs

Note that, in the 3-D plot shown in Figure 5, the 3 axes that the data points lie between do not have any significance and cannot be interpreted. All that one can learn from these 2-D and 3-D visualizations is that there appears to be some distinct clusters in these versions of the data considering that data points that are close to each other are lower in variance compared to the ones that are further apart, and one can see many distinct groups of close-nit data points in these visualizations.

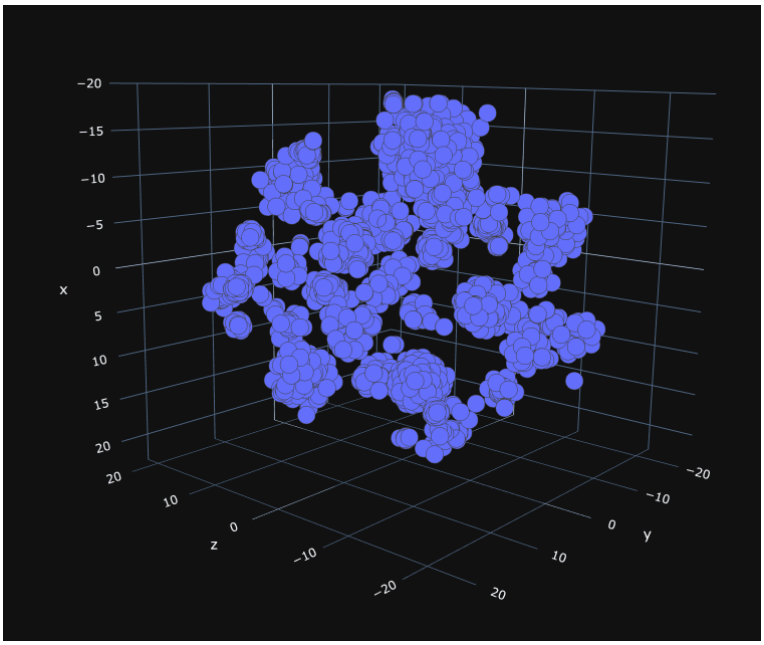

Figure 5. An image capture of a 3-D plot showing the t-SNE visualization for PCA-reduced DBVs

The process above was repeated on the original DBVs with 1052 dimensions to evaluate the effects of the initial PCA on the success of t-SNE. These visualizations were also plotted to 2-D and 3-D (see Figures 6 and 7), and these figures show that t-SNE algorithm was not successful in discovering groups in the original version of the data under the non-linearity assumption, and it would not succeed until after the linearity was captured with the reduction to the components the initial PCA provided as demonstrated in the previous case.

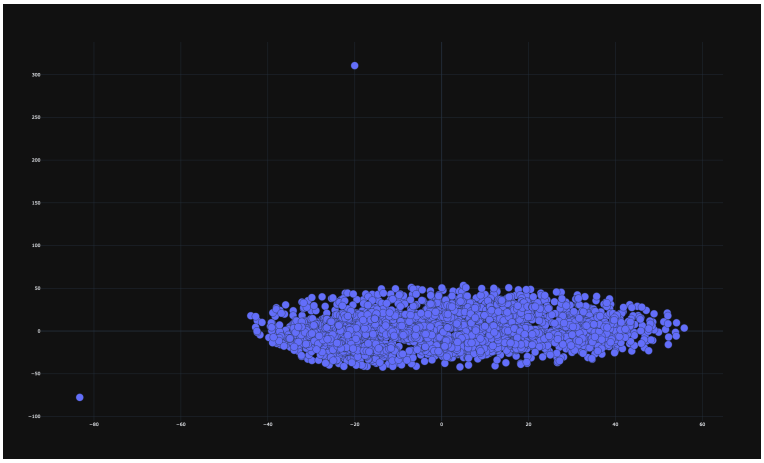

Figure 6. A 2-D plot showing the t-SNE visualization for the unreduced DBVs

One can see some peculiar outliers in the visualizations along with a huge cluster of undifferentiated data points when all 1052 of the original features are used without first linearly reducing the dimensions with PCA.

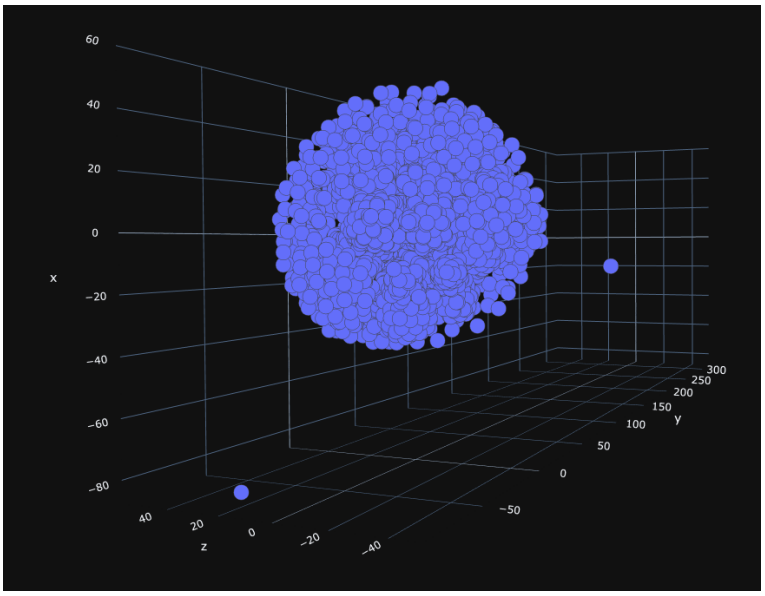

Figure 7. An image capture of a 3-D plot showing the t-SNE visualization for the unreduced DBVs

The same trend can be observed from Figure 7 that is a 3-D visualization of the original DBVs data set with t-SNE.

4.1.2.2 t-SNE Visualization for BDBVs

4052 data entries of BDBVs with 49 components as their new features were visualized using t-SNE with dimensionality reduction to 3 components. The visualizations were plotted to 2-D (Figure-8) and 3-D (Figure-9).

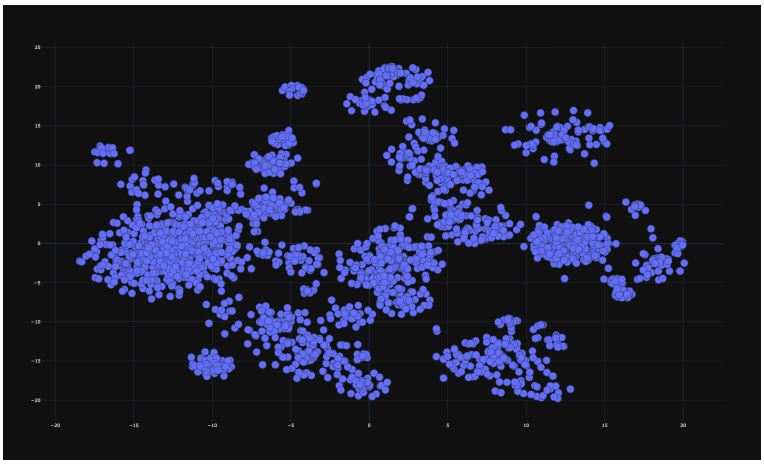

Figure 8. A 2-D plot showing the t-SNE visualization for PCA-reduced BDBVs

Note, again, how it seems that the algorithm manages to capture distinct groups of data points that are lower in variance.

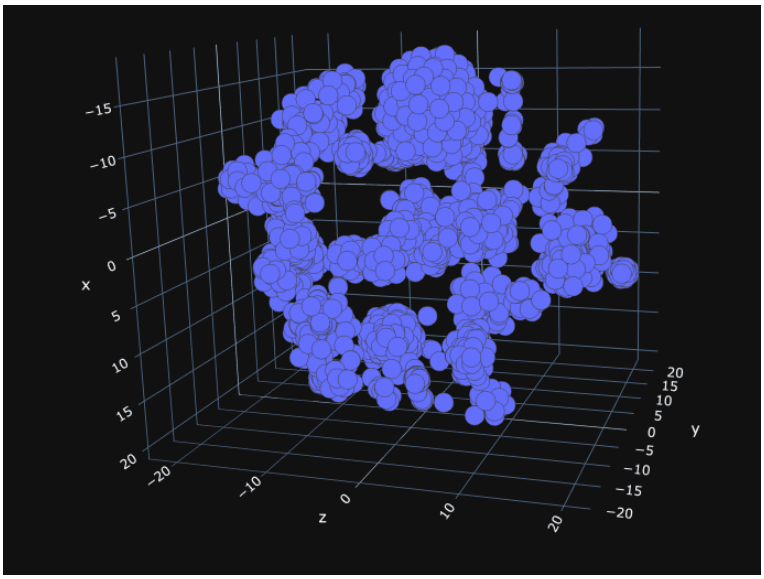

Figure 9. An image capture of a 3-D plot showing the t-SNE visualization for PCA-reduced BDBVs

The success of the t-SNE algorithm for the version of the BDBVs with linearly reduced dimensionality is quite similar to the one achieved for the reduced DBVs. One can see the formation of distinct groups within the data when 49 features that were linearly extracted with PCA were further reduced to a 3 dimensional feature representation with the use of t-SNE.

The same process of using only t-SNE to reduce the dimensionality to 3 dimensions was repeated on the original BDBVs with all 1046 of the dimensions to evaluate the effects

of the initial PCA on the success of t-SNE. These visualizations were also shown in 2-D and 3-D plots (Figures 10 and 11).

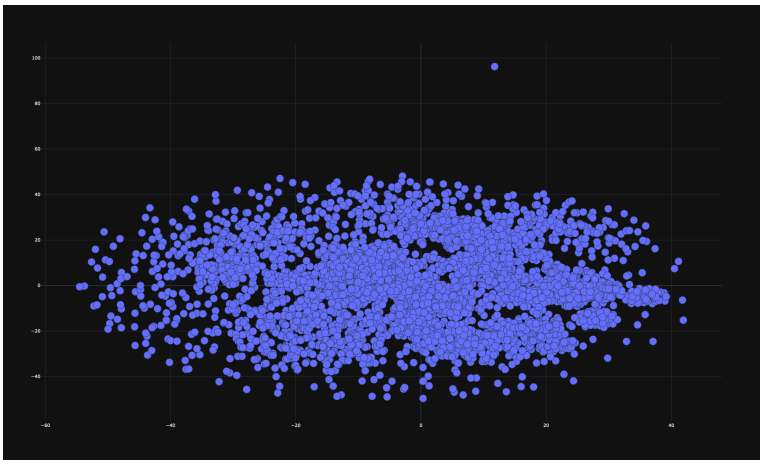

Figure 10. A 2-D plot showing the t-SNE visualization for the unreduced BDBVs

Again, it can be seen that t-SNE is not successful under the non-linearity assumption for the BDBV data if the dimensions were not first linearly reduced with PCA.

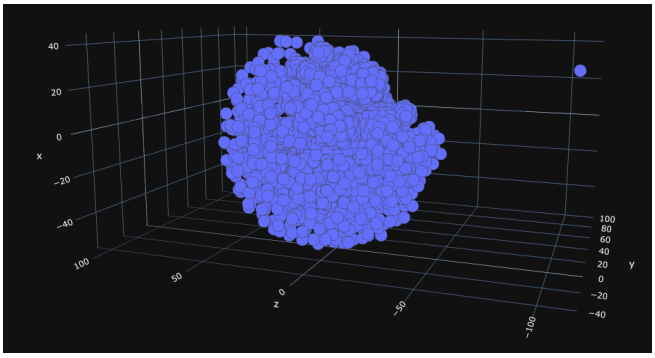

Figure 11. An image capture of a 3-D plot showing the t-SNE visualization for the unreduced BDBVs

4.1.2.3 t-SNE Visualization for LVs

4052 data entries of LVs with 198 components as their new features were visualized using t-SNE with dimensionality reduction to 3 components. The visualizations were plotted to 2-D and 3-D (see Figures 12 and 13).

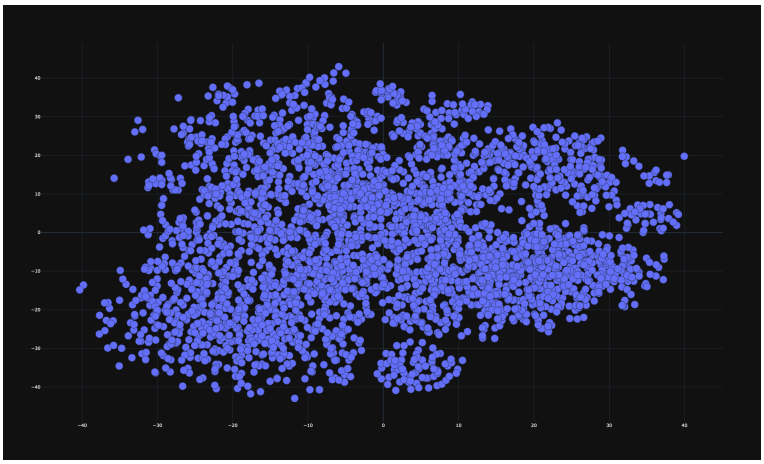

Figure 12. A 2-D plot showing the t-SNE visualization for PCA-reduced LVs

The algorithm manages to somewhat differentiate the data points for the PCA-reduced LVs even though the PCA reduction was to 198 components, which means that the dimensionality was still high for t-SNE to handle.

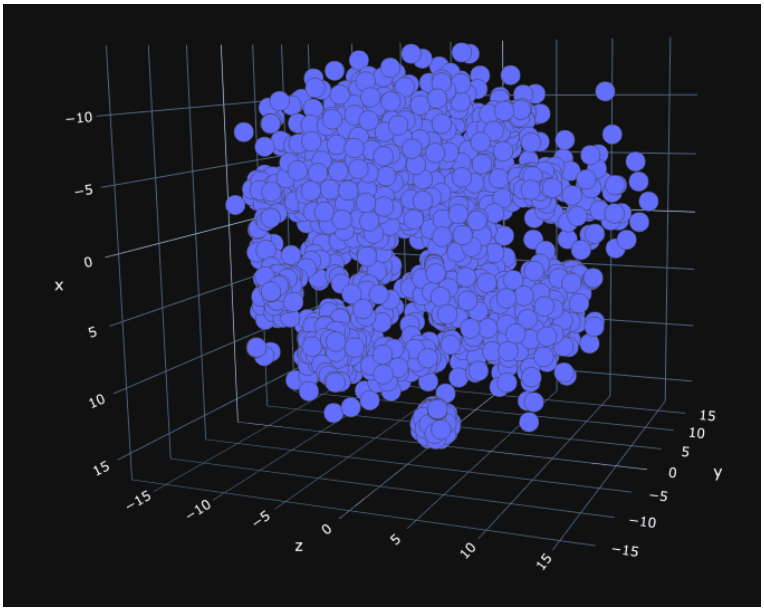

Figure 13. An image capture of a 3-D plot showing the t-SNE visualization for PCA-reduced LVs

To test the non-linearity assumption for the LVs data set, the dimensionality reduction to 3 dimensions with t-SNE was done for the original LVs with 1051 dimensions to evaluate the effects of the initial PCA to 198 components on the success of t-SNE. These visualizations are shown in 2-D and 3-D plots (Figures 14 and 15).

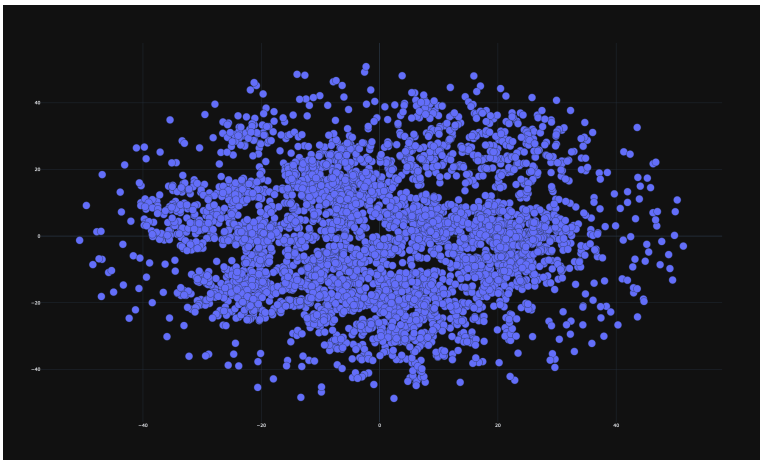

Figure 14. A 2-D plot showing the t-SNE visualization for the unreduced LVs

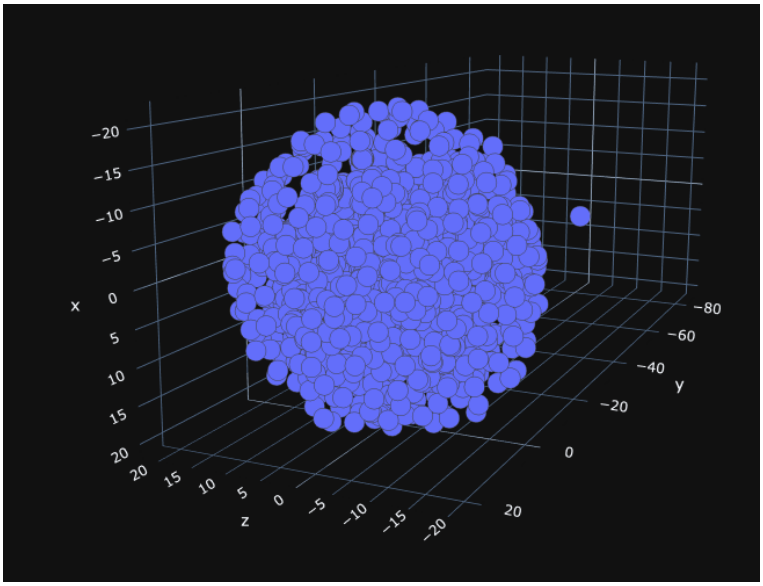

Figure 15. An image capture of a 3-D plot showing the t-SNE visualization for the unreduced LVs

Again, it seems that the choice to first linearly reduce the dimensionality with PCA and to later use a non-linear DR technique such as t-SNE seems to help in capturing more refined groups in the data.

4.2 Analyses involving Cosine Similarity Measures

In order to test the success of the original representations that were based on lambda calculus and De Bruijn indexing and also the success of the reduced representations,

a similarity metric called cosine similarity was used (Huang, 2008). Cosine similarity was preferred rather than using, for instance, a similarity metric that utilizes Euclidean distance because cosine similarity is a measure that takes into account the similarity of the directions of any two vectors (deeming vectors with similar angles from the origin to be similar to each other), which leads to comparisons that are independent of the magnitude each of the compared vectors may have (because it uses the normalized version of the dot product) (Huang, 2008). It is therefore a more desirable method when the data is high in dimensionality, and when the vectors are sparse. Note that cosine similarity would yield the same results with the Euclidean measure if the data sets were normalized prior to the comparisons. However, as it was reported in the first section of this chapter, the data sets were not normalized, and hence the two metrics would yield different results.

Cosine similarity was used as a metric of success of any and all vectorial representations arising from the original data in various forms they can take (reduced or unreduced) to compare these representations and determine the success of the formalism or the reduction of representations under certain formalisms. A similarity measure can be used as such a metric because by using common sense knowledge about the actual semantic similarities of words, one can tell apart successful representations schemes from less successful ones. These similarity metrics could be further used and tested as part of a clustering algorithm to categorize words into groups, which can, again, be tested against the knowledge of the semantics of words to tell apart the success of the clustering algorithms and the success of the representational scheme or the dimensionality level of the data set that was used.

In the upcoming subsection, cosine similarity metric is used to get a bird's-eye view of the similarity relationships in the 3 versions of each 3 of the data sets: the version with the original number of dimensionality, the one with the PCA reduction in dimensionality, and the version with the added t-SNE reduction. In the subsection that follows, representation similarity is used to gain insights into individual word representations within and across the three versions of the data sets.

4.2.1 Cosine Similarity Heatmaps for the 3 Data Sets

For each data set, 3 distinct similarity matrices composed of the similarity scores of each and every data point with each other were created using the cosine similarity tool in Scikit-learn in Python 3.8.12 (Pedregosa et al., 2011; van Rossum, 2020). These similarity matrices were then used to generate heatmaps (with the aid of the Seaborn library (Waskom, 2021) in Python (van Rossum, 2020)) that visualize with color the point-wise similarity between 4052 representations within each 3 of the data sets and for 3 different levels of their dimensionalities. This technique was employed to aid in general comparisons of the representational schemes of the 3 data sets, and the evaluations of their dimensionality and the reduction of their dimensionality. The insights gained from these visualizations are briefly described.

For all of the heatmaps that are shown in this section, in accordance with the color legends on the right side of each heatmap, as the number in the legend gets closer to 1, the

corresponding colors in the legend indicate higher similarity between 2 word representations, which is reflected in the use of the said color inside the map where two word representations' similarity with each other is indicated. Each heatmap reflects the similarity comparison of each word representation with all other 4052 word representations (including itself).

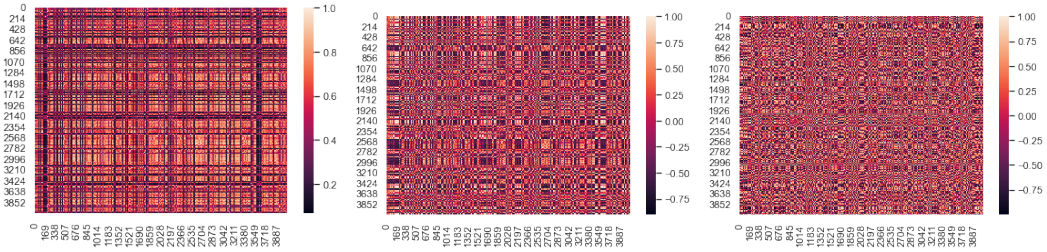

Figure 16. Figure Consisting of the Cosine Similarity Heatmaps for the DBVs with 1052 dimensions (original), 40 dimensions (reduced w/ PCA), and 3 dimensions (reduced w/ t-SNE) respectively

Figure 16 shows the heatmaps for the 3 versions of the DBVs data set side by side. Even though the maps are hard to read because of the vast amount of data points that are involved (4052), these 3 figures seem to capture a trend where, as the dimensionality is reduced, the heatmap representations become more fine-grained or nuanced in the overall similarity in the data, which is to be expected from the feature extraction that comes with DR. Looking at the first heatmap that is constructed with the data set that had all 1052 of the dimensions, one can identify consistent instances of individual data points that are highly dissimilar to almost all other data points that they are compared to. Indeed, the first heatmap seems to be marked by extremes other than the previously mentioned outlier cases, where, there is an overall dominance in orange to yellow colors, indicating the dominance of levels of similarity that are unrealistically high across the data set. This observation seems to be consistent with the previous visualization and dimensionality reduction of the 1052 dimensional DBVs with t-SNE where there were a couple of outliers that were situated far away from one large and undistinguished cluster consisting of all the remaining data points.

With the second heat map in Figure 16, finer groups seem to emerge, which, based on their purple to black coloring, indicates quite low levels of similarity. It is important to note that these are in the form of groups of certain representations rather than in the form of a couple of outlier data points that result in types of coloring that are vertically or horizontally consistent. With the final heatmap, one can see even smaller and even finer chunks of coloring that are diverse across the heatmap, which is closer to the variance in similarity that would be expected from this lexicon data.

Figure 17 consists of the heatmaps for 3 versions of the BDBVs. One can see that the observations above regarding the change in heatmaps as the dimensionality decreases also apply to the trend in Figure 17 for the BDBVs. Perhaps the only notable difference is in the salience of these same trends that are observed in the case of the BDBVs.

One can see that the outlier cases show even lower similarity scores in the outlier cases of the BDBVs and the outliers even seem to be slightly higher in number compared to the first heatmap in Figure 16 with the unreduced dimensionalities. As the BDBV dimensionalities get reduced, the trends in the remaining two heatmaps turn out to be quite similar to the ones in the last two heatmaps for DBVs, which is to be expected as the data set gets closer to the true dimensionality of the data that is bound to be similar to the one found in the reductions for the DBVs.

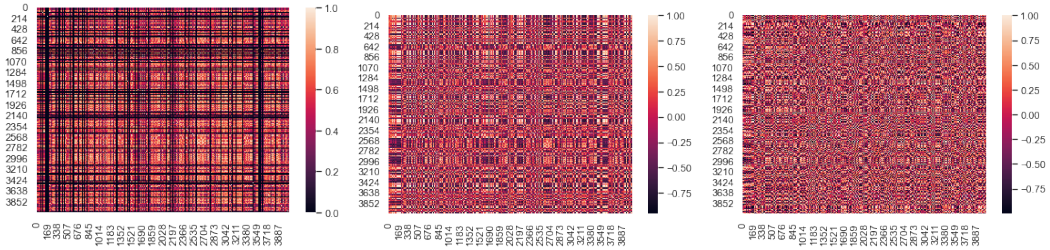

Figure 17. Figure Consisting of the Cosine Similarity Heatmaps for the BDBVs with 1046 dimensions (original), 49 dimensions (reduced w/ PCA), and 3 dimensions (reduced w/ t-SNE) respectively

Finally, one can see the heatmaps for the 3 versions of the LVs data set in Figure 18, each containing representations with fewer number of dimensions.

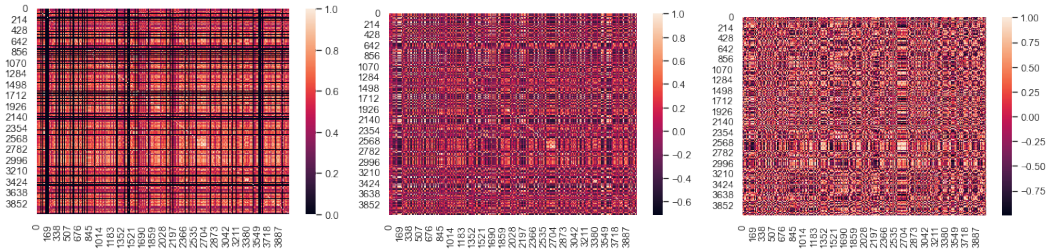

Figure 18. Figure Consisting of the Cosine Similarity Heatmaps for the LVs with 1051 dimensions (original), 198 dimensions (reduced w/ PCA), and 3 dimensions (reduced w/ t-SNE) respectively

The comments that were made for the heatmaps of each three of the versions of DBVs and BDBVs, which were comments that resulted from the effects of the dimensionality reduction in these data sets, can also be made for the heatmaps for LVs. However, one interesting observation that can be made from Figure 18 is about the second heatmap that was achieved through PCA. One can see that the color that is dominant in this heatmap (which has a more pink/purple tone) is different than the one that is dominant in the second heatmap of the DBVs and BDBVs (which has a more red/orange tone) where this color in LVs is more indicative of overall dissimilarity of the data points within the LVs data set with 198 dimensions, even though the first heatmap of the LVs were more

dominated by orange (showing more similarity). So, even though the dimensionality reduction with PCA reduced outliers and emphasized the groups in the LVs data, it seems that it also extracted features that accentuate the dissimilarities in the data which may be due to the dimensionality reduction step here being a linear one and the dimensionality still being too high to make sense of the important features in this data set. With the t-SNE reduction, one can see the finer grained groups emerging and a return to the domination of colors that signal similarity across the data points, which maybe due to t-SNE's success in discovering non-linearity in data.

4.2.2 Cosine Similarity Comparisons for the Evaluation of Selected Word Representations Across Data Sets

Cosine similarity metric was implemented in Python 3.8.12 (van Rossum, 2020) using a pair-wise cosine similarity function that uses the normalized dot products of the vectors, in order to take a closer look at and compare the similarities of word representations within a given data set, and the inferred accuracy in the outcomes of these comparisons were used to evaluate the success of the representational scheme unique to the data sets with the original dimensionalities.

Table 7, 8, and 9 list the cosine similarities between the same select group of word pairs for the vectorial representations in DBVs, BDBVs and LVs data sets. Table 10 summarizes two important characteristics of the original lexical entries for the words that were compared in the aforementioned pairs: their PoS Tag and their semantic representation with a unique lambda term.

Table 7

Pairwise Cosine Similarity Percentages for the Original DBVs Data Set

Pair	Cosine Similarity Percentage for the Pair
(gave, take)	75%
(gave, get)	75%
(gave, did)	85%
(gave, put)	87%
(gave, Eve)	15%
(take, get)	97%
(take, did)	54%
(take, put)	80%
(get, did)	54%
(get, put)	80%
(did, put)	74%
(get, Eve)	11%
(furniture, Eve)	49.99%

The table above that lists the cosine similarities between the word pairs reveal certain aspects of the semantics of the representations that were done with the DBVs. One can

see that the verb representations and noun representations have low similarity ratios such as 15% and 11% similarity, while verb-verb pairs or a noun-noun pairs start with a base ratio of around 49% similarity. As the verbs get more similar in their combination properties (such as taking closer number of arguments), their similarity ratio increases. For instance the gave-get pair which shares the property of being a verb taking respectively 3 and 4 arguments has a similarity score of 75% while the take-get pair which shares the property of being a verb and the property of taking 4 arguments has a similarity score of 97%. Another verb-verb pair, gave-did which take 3 and 2 arguments respectively has a similarity score of 85%. These kinds of trends in these similarity scores seems to be consistent with what would be expected from the semantics of these verbs.

Table 8

Pairwise Cosine Similarity Percentages for the Original BDBVs Data Set

Pair	Cosine Similarity Percentage for the Pair
(gave, take)	72%
(gave, get)	72%
(gave, did)	85%
(gave, put)	84%
(gave, Eve)	0%
(take, get)	96%
(take, did)	49.99%
(take, put)	79%
(get, did)	49.99%
(get, put)	79%
(did, put)	70%
(get, Eve)	0%
(furniture, Eve)	0%

In Table 8 one can see the pair-wise similarity scores of the same verbs denoted with the BDBVs. One crucial observation is that, in this representation, the similarity scores of words that belong to same broad categories (such as noun or verb) do not necessarily show a baseline correlation. Instead, as can be seen in the result for the furniture-Eve pair, the similarity could turn out to be 0. In a fashion that is consistent with the previous observation, one can see that the BDBVs similarity pairs show similar trends to DBVs in their pair-wise similarity, however, most of their values are 3-10% less than the ones that are observed in the DBVs. This trend suggests that DBVs provide a baseline level of similarity for the representations in the data set. It also seems that this baseline similarity can work as a differentiating factor in the DBVs. Where the get-Eve and furniture-Eve pairs have a similarity score of 0 in the BDBVs, the DBVs differentiate them as verb-noun versus verb-verb pairs by virtue of their respective similarity ratios of 11% and 49.00%.

Lastly, Table 8 lists these pair-wise similarities for the LVs data set. One can see that the similarity scores with the LV representation is even less differentiating then that of BDBVs, with lower baseline similarity levels and score that could be as low as 0%.

Table 9

Pairwise Cosine Similarity Percentages for the Original LVs Data Set

Pair	Cosine Similarity Percentage for the Pair
(gave, take)	70%
(gave, get)	70%
(gave, did)	80%
(gave, put)	80%
(gave, Eve)	0%
(take, get)	88%
(take, did)	62%
(take, put)	75%
(get, did)	62%
(get, put)	75%
(did, put)	71%
(get, Eve)	0%
(furniture, Eve)	0%

Table 10

Semantic Representations and the PoS Tags of the Words in the Example Pairs

Word Form with Its PoS Tag	Word's Semantic Representation with a Lambda Term
gave dv2>	λx.λy.λz.SIMP PST GIVE x y z
take dv1-outof>	λx.λy.λz.λa.SIMP z TAKE y x a
get dv1-for	λx.λy.λz.λa.SIMP z GET y x a
furniture n	FURNITURE
Eve pn	EVE
did tvd>	λx.λy.SIMP PST DO x y
put tv1>	λx.λyλz.SIMP y PUT x z

In light of these observations from the data and considering the real properties of these word entries, it seems that the DBVs are the more successful representations of word-meanings compared to the BDBVs and LVs. It is certain that both the DBVs and BDBVs are more successful than the LVs based on the cosine similarity scores they yield for the selected word pairs. However, the success of DBVs in comparison to that of BDBVs was worthy of further exploration, especially worth investigating was the possible reasons for the claim that DBVs seem to provide a baseline similarity score for all of the pairs compared in contrast to the lower baseline scores for the pairs in the BDBVs.

Remember that the only difference between the DBVs and the BDBVs was that the DBVs have the additional array of length 6 that is based on the thermometer encoding of the number of lambdas contained in each data point. This difference in their encoding may be the reason why the pairs for the DBVs tend to have higher baseline similarities, since they have additional arrays that in most cases add one or more 1's to the encodings along with some number of zeroes, which turns out to be a significant difference given that the vectors in each of the data sets are sparse and any additional non-zero value that could get added with the thermometer encoding has the power to change the representations of the vectors significantly.

The sparsity in the data was intentionally left as is after creating 3 types of vectors from the lambda terms and De Bruijn terms. In other words, no dense vector representation was sought out for the sparse vectors in the 3 data sets. This is not only because the PCA algorithm is capable of handling sparsity in the data, but also because the sparsity in the vectors is generally found to be beneficial if PCA is to be implemented. Another rationale for using the sparse vectors without making them denser was regarding the goal of keeping the numbers in the De Bruijn sequences (which are included as arrays both in the DBVs and BDBVs) intact and also so that the numbers in the De Bruijn sequences which usually consist of values that between 1 and 6 have a more pronounced affect in each of the otherwise sparse feature vectors in the data sets.

Given the considerations above, it seems reasonable and also desirable that the DBV representation provides a more accurate measure of the cosine similarity between two words, however, there may be reasons why this effect is not reflective of the true dimensionality of the phenomena regarding the semantic representations of words. It may be that the original, unreduced, versions of the DBVs and BDBVs were not able to account for the independent effect of using the De Bruijn sequences in both, which is a feature set that carries the most important representations (the linear representation of the function application phenomena found in the original lambda terms) in terms of it reflecting the most important structural parts of the semantic representations and it carrying the most weight in individuating a unique semantic representation together with the actual content of the concept words (which were encoded with BoW).

If it is indeed the case that the original representations in the DBVs and BDBVs were not reflective of the true dimensionality in the source data set, it could be that the additional lambda array in the DBVs may be aiding them in the right direction by adding extra information about the structure of the De Bruijn sequences (by giving the number of lambdas), therefore yielding better similarity scores with higher baselines. One way of testing the aforementioned claims about De Bruijn sequences themselves being impor-

tant dimensions in the original data that could potentially be useful in and of themselves but them not being reflected enough in the original versions of the DBVs and BDBVs is to compare the pair-wise cosine similarities of these data sets using the versions where their dimensionality is reduced with PCA. If one sees that the similarity scores for the PCA-reduced DBV pairs remain as good as they were in the original versions, and if they are still better than that of the BDBV pairs, it would be reasonable to claim that the array indicating the number of lambdas should be considered to be an essentially important part of the vectorial representations of the lambda terms and features that are not redundant when it comes to the true dimensionality of the data.

To that end, PCA was applied to the DBVs and BDBVs to compare the pair-wise cosine similarities of the reduced versions. One important thing to note is that, as it was seen in the first section of the chapter, the use of PCA to reduce both of these data sets to significantly lower number of components results in a significant loss in the amount of variance explained for these data sets. New version of the DBVs and BDBVs were prepared where they were reduced to their principal components that account for 95% of variance in each data set, reducing their number of features to 170 and to 192 respectively, so as to test a version of these 2 data sets that still explained an important amount of the variance (95%) in these data sets. New versions of the DBVs and BDBVs with principal components that account for 90% of variance in the data were also created, which meant 40 features for the DBVs and 49 features for the BDBVs.

Note that if the additional lambda array in the DBVs is given important weight when PCA uses the sparsity in the data during DR (which is a likely outcome), it would be expected that the effect of the lambda array will have a more pronounced effect in the reduced versions of the DBVs along with the array for the De Bruijn sequences. In a similar vein, it may be that the reduced versions of the BDBVs will better and more prominently reflect the effect of the De Bruijn sequences, and this may mean that their similarity scores for the PCA-reduced pairs will be more accurate than that of or now will be as accurate as the DBVs, if it is indeed the case that the De Bruijn sequences are the most important dimensions of the original data and that the array for the number of lambdas is actually redundant. Also, if this is the case, the reduced versions of the DBVs having more pronounced effects from the lambda arrays and the De Bruijn sequence arrays may have an adverse effect in these representations, and could yield less accurate cosine similarities for the pairs in question, as compared to their unreduced versions.

Table 11 shows side by side the cosine similarity scores that were found for the pairs for the DBVs and BDBVs reduced to components accounting for 95% of the variance in each respective data sets.

Table 12 shows the cosine similarity scores that were found for the pairs for the DBVs and BDBVs reduced to components accounting for 90% of the variance in each of the data sets.

Both of these tables show a collective trend towards PCA-reduced BDBVs having scores closer to that of the reduced DBVs, and one can even see BDBVs becoming slightly more accurate than the DBVs' similarity scores in some instances. One can also observe that the PCA-reduced DBVs, while still yielding pair-wise similarity scores closer to the

Table 11

Cosine Similarities for the DBVs and BDBVs Reduced to Their Components That Explain 95% of the Variation

Pair	Similarity for DBVs-95%-PCA	Similarity for BDBVs-95%-PCA
(gave, take)	25.9%	24%
(gave, get)	26%	24%
(gave, did)	43.6%	48.5%
(gave, put)	55%	50%
(gave, Eve)	-61%	-59%
(take, get)	93%	92%
(take, did)	-42%	-43%
(take, put)	39%	38%
(get, did)	-41.9%	-42%
(get, put)	39%	38%
(did, put)	3.4%	5%
(get, Eve)	-75.5%	-79%
(furniture, Eve)	99.9%	1%

Table 12

Cosine Similarities for the DBVs and BDBVs Reduced to Their Components That Explain 90% of the Variation

Pair	Similarity for DBVs-90%-PCA	Similarity for BDBVs-90%-PCA
(gave, take)	27.9%	24.5%
(gave, get)	28%	24.5%
(gave, did)	46.8%	49%
(gave, put)	59%	51%
(gave, Eve)	-65.6%	-60%
(take, get)	93%	92%
(take, did)	-42%	-43%
(take, put)	39%	38.9%
(get, did)	-42%	-42.9%
(get, put)	39%	39%
(did, put)	3.4%	5%
(get, Eve)	-75.5%	-79%
(furniture, Eve)	1%	1%

ones they yielded for the unreduced DBVs, started yielding some significantly different similarity scores that were also less accurate compared to the original scores.

An important result to note is that, even though the exact same cosine similarity function was used in the computations of the pair-wise cosine similarities, the cosine similarity scores that were found for each 4 new case with reduced DBVs and BDBVs started yielding similarity scores that can get negative values. Note that cosine similarity is ought to be -1 and 1, and the function that is implemented here could yield results with values that are smaller than 0 when it was used for the unreduced DBVs and BDBVs, but it did not. It only started yielding results with values smaller than 0 and even ones that get significantly close to -1 when the dimensionality was reduced in the two different cases reported above. It means that the vectors of the reduced versions started to indicate for the representations cases where the similarity of a word pair could indicate them being opposite in a way (opposite in their directions in the coordinate system).

This effect of starting to get negative similarity scores with the reduced versions may be due to the fact that the principal components that each 4 versions were reduced to consisted of negative values in their features as well as positive ones. When these scores that varied from -1 to 1 were scaled to be in between 0 and 1 using cosine distances, it was again observed that the cosine similarities for the BDBVs started to get closer to those of DBVs and the overall similarity scores seemed to be getting more selective, yielding lower baseline similarity as compared to the scores from the unreduced versions of the two data sets.

For instance, while the similarity score for the Furniture-Eve pair in the BDBVs was previously 0, for the unreduced versions, the score for the pair became 1 in most of the cases, which is an example of the cosine similarity scores becoming more accurate in the right direction even if it was with a slight overshoot for the similarity scores for the reduced versions of BDBVs. For the same pair, the unreduced version of the DBVs had 49% similarity while the reduced version started yielding 1. The convergence of most of the similarity scores into a certain value for most of the reduced versions in both DBVs and BDBVs seem to suggest that the PCA that was done in the reduction extracted features that are predominantly reflective of the De Bruijn sequence values in the DBVs and the BDBVs. It can be said that the reduced versions of the DBVs and BDBVs accentuate the importance of the De Bruijn sequence values and they seem not to differentiate the vectors based on the values in their concept arrays (since they are in most cases 1 or more features with the value of 1, and since the concept arrays are usually same with or quite similar to each other barring the ordering of the 1's and 0's). For instance, what differentiated the original vector representation for the word "furniture" from the one for "Eve" was that in their concept arrays, they had the value of 1 for either the concept furniture$'$ or the concept Eve$'$ in different locations. It seems that the reduced versions of the DBVs and BDBVs made this difference in their vectors redundant because the cosine similarity for the pair started yielding 1 in most cases, making them equivalent to each other, when really they should not be. However, using the PCA-reduced versions of these vectors seem to be a good strategy to use these vectors according to their combinatory properties that are reflected in the specific De Bruijn sequences they have.

4.3 Cluster Analysis using the Affinity Propagation Algorithm

A clustering method called Affinity Propagation (AP) (Frey & Dueck, 2005) was employed to further explore the implications of the original dimensionalities and the reduced dimensionalities for the DBVs data set, which was considered to be the data set with the most useful vectorial representation for the semantics of words based on the previous analyses. Clustering was employed especially with the purpose of seeing whether verb classes could be found based on the DBV representations. In other words, clustering was used to see how successful the original DBVs were in them making it possible to cluster verbs according to the order and number of arguments any verb may take.

The reasoning for choosing this clustering algorithm was based on its flexible parameter selection possibilities, which did not require a specification with regards to the number of clusters to be made, a restriction which this thesis would not benefit from imposing. Affinity Propagation Algorithm, which was introduced by Frey and Dueck (2005) is an algorithm that combines the strengths of many clustering algorithms such as the ones that are hierarchical and the ones that are based on mixture models such as the Expectation Maximization algorithm that is based on the Gaussian Mixture Model (Crouse, Willett, Pattipati, & Svensson, 2011), all of which were reasons why AP was considered to be the ideal algorithm to use. AP takes the similarity of data points according to some metric of similarity and by taking each and every data point as potential exemplars that share information with other data points, it tries to come up with a way to cluster them according to some converging preference for weights given to the effects of each exemplar on each other data point (Frey & Dueck, 2005).

The AP that is implemented in this thesis was based on Scikit-learn Cluster's Affinity Propagation tool (Pedregosa et al., 2011).

4.3.1 Affinity Propagation For the DBVs

Out of the 4052 word entries in the data, 3508 entries had at most 3 lambdas in them, 1343 of which had exactly 3 lambdas, 818 of which had exactly 2 lambdas and 521 of which had exactly 1 lambda, and 826 of which had zero lambdas. To discover fine-grained clusters of verbs based on their argument realization properties, the analyses in this section focused on finding clusters in verbs with at most 3 lambdas in them because these were the ones that have sufficient (in terms of aiding in the discovery of clusters) amount of instances in the data.

AP was used to find clusters in this collection of 3508 DBV entries with at most 3 lambdas in them to get a sense of the overall data, and then it was used to find more specific clusters within the verbs in these 3508 entries that are in an equivalence class in terms of the number of arguments they take. 2700 of the 4052 entries in the data set were for verbs, and these verb representations were tokens belonging to 255 unique types of part of speech tags. 2185 of these 2700 verbs had at most 3 lambdas in their semantic representation. A further refinement was done to select a subset of the verbs where, out of the 2185 verbs, only the verbs that belong to a subclass of PoS tags with transitive

(TV), intransitive (IV), and ditransitive verbs (DV). 1960 of 2185 verbs with at most 3 lambdas met this condition, while the amount of verbs that belong to those PoS classes is 2471 out of 2700 in the verb-only subset of the overall data. Out of those, 1247 verbs had exactly 3 lambdas in them, 638 verbs had exactly 2 lambdas in them, and 75 of them had exactly 1 lambda in them. Since 75 instances was not enough to use in comparisons of clusters, only the TV, IV, and DV verbs with exactly 2 and exactly 3 lambdas in them were considered in specific verb subclass comparisons.

The reason why the verb subset of the overall data set was refined to only include PoS tags that are TV, IV, and DV (thus, for instance, excluding verbs that have auxiliary verb or modal verb as their PoS tag), was so that the analyses focused on the most informative phenomena (because of the large number of their tokens in the data set) and the most interesting verbs in terms of gaining some insights into argument realization phenomena. The PoS tags with TV, IV, and DV make yup 245 unique PoS tags specifically, and 149 of these are for TVs, 33 are for IVs, and 63 are for DVs.

For a visual comparison of the clusters AP came up with for the 3508 entries with at most 3 lambdas and the aforementioned 1960 verb-only entries with at most 3 lambdas, see Figure 19. Within the second cluster in Figure 19 are the verbs that take exactly 1, exactly 2, and exactly 3 arguments. Note that the AP algorithm that is used to generate the clusters in both of the visuals in Figure 19 was specified such that it took into account 3 features while deciding on the clusters, which was a number that was intentionally selected to use this clustering algorithm to consider a lower amount of dimensionality while finding the clusters.

One can see how the algorithm manages to find visible clusters from both of the visualizations even when it is restricted to consider an amount of features that is as small as taking into account only 3, and one can also see that the refined verb subgroup shows even more distinct classes.

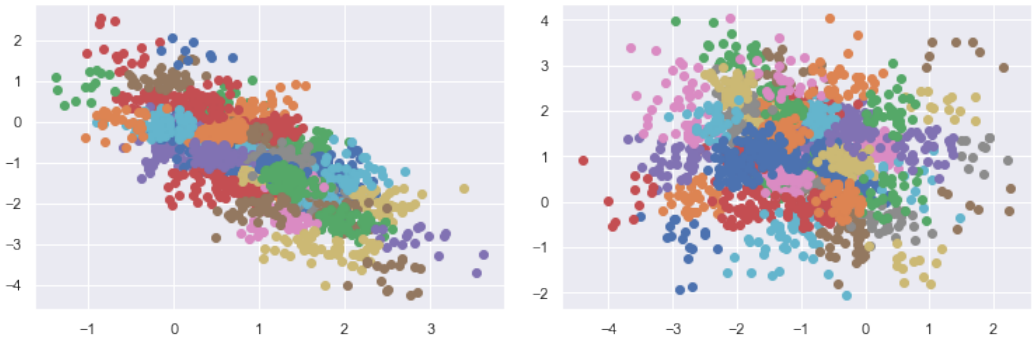

Figure 19. Side by Side Visualization of the AP Clusters for Words with at Most 3 Lambdas and the AP Cluster for a PoS-specific Subclass of Verbs with at Most 3 Lambdas in the DBVs Data Set

In the selected subclass, the number of verbs with exactly 3 lambdas was 1267, and 638 entries had exactly 2 lambdas. These two groups were the largest ones, and were

isolated to take a closer look at the clusters AP yielded for them. The visualizations of these clusters are given in Figure 20 below.

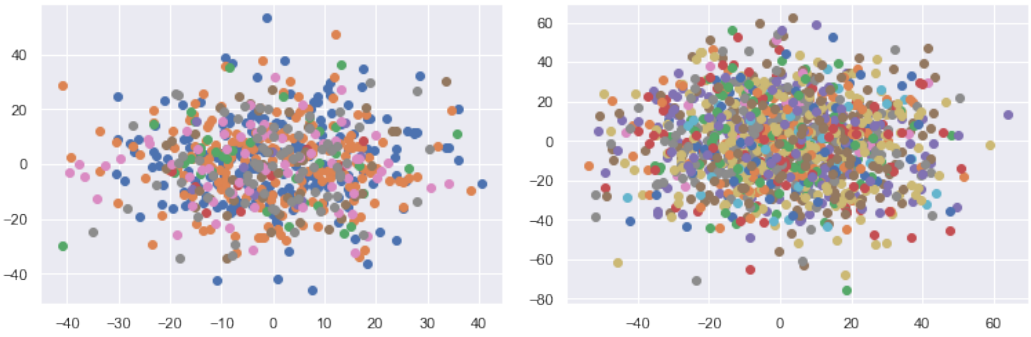

Figure 20. Side by Side Visualization of the AP Clusters for Verbs with Exactly 2 and Exactly 3 Lambdas in the DBVs Data Set

Note that these AP clusters were generated based on using all of the data points as features both in case of the verb subset with exactly 3 lambdas and the one with verbs that contain 2 lambdas. For instance, the AP clusters shown in Figure 20 were found by making sure that the algorithm looks for 638 features (for it to consider every data point as independent features). The aforementioned way of setting up the features turned out to be a method which yielded successful clusters. Setting up the number of features to be any lower than the number of data points for each of the subsets yielded unsuccessful clusters.

Another important note about the implementation of AP which resulted in the clusters that can be seen in Figure 20 is that the implementation made use of a cosine similarity matrix of each of the data sets that were computed with Scikit-learn's Cosine Similarity tool (Pedregosa et al., 2011). The use of this metric was intentional in that when the AP was implemented using Euclidean distances as a measure, the resulting clusters for each of the data subsets were not deemed to be satisfactory because in these clusters, there was not any differentiation between, say, a DBV which has a De Bruijn sequence that is '1 2', from a DBV which has a De Bruijn sequence that is '2 1'. It seems that using a similarity metric that does not take into account point-wise distances but is rather based on the angle of the vectors from the origin (i.e. cosine similarity metric) is what enables the AP implementation here to capture clusters based on structural similarities of DBVs.

For an example of the success of the implementation of AP that is done in the fashion described above, see the two consecutive tables below (Table 13 and Table 14), which show the contents of the 1st cluster and 4th cluster out of the 26 clusters that the AP found for the data subset containing only the verbs with exactly 3 lambdas.

These two examples are representative of the overall success of the AP implementation for both the verb subset with 2 lambdas and the subset with 3 lambdas. Realize how, barring a few outlier cases, the algorithm manages to group together the vectorial representations that have the same De Bruijn sequence in the same cluster.

Table 13

Cluster 1 of the 26 Clusters Found in the Verb Subset with Exactly 3 Lambdas

The Word and Its PoS Tag	DB Terms for The 3-Lambda Verb Entries in Cluster 1 of 26
been tvb>	(LAM (LAM (LAM ((PRFT 2) (3 1)))))
been tvb>	(LAM (LAM (LAM ((((PRFT 2) EQ) 3) 1))))
done tv3>	(LAM (LAM (LAM ((((PRFT 2) DO) 3) 1))))
eaten tv3>	(LAM (LAM (LAM ((((PRFT 2) EAT) 3) 1))))
finished tv3>	(LAM (LAM (LAM ((((PRFT 2) FINISH) 3) 1))))
got tv3>	(LAM (LAM (LAM ((((PRFT 2) GET) 3) 1))))
had tvh3>	(LAM (LAM (LAM ((((PRFT 2) HAVE) 3) 1))))
heard tv3>	(LAM (LAM (LAM ((((PRFT 2) HEAR) 3) 1))))
lost tv3>	(LAM (LAM (LAM ((((PRFT 2) LOSE) 3) 1))))
put tv3>	(LAM (LAM (LAM ((((PRFT 2) PUT) 3) 1))))
used tv3>	(LAM (LAM (LAM ((((PRFT 2) USE) 3) 1))))

Table 14

Cluster 4 of the 26 Clusters Found in the Verb Subset with Exactly 3 Lambdas

The Word and Its PoS Tag	DB Terms for The 3-Lambda Verb Entries in Cluster 4 of 26
been tvb<	(LAM (LAM (LAM ((((PRFT 1) EQ) 2) 3))))
been tvb<	(LAM (LAM (LAM ((2 (PRFT 1)) 3))))
been tvb	(LAM (LAM (LAM ((((PRFT 2) EQ) 1) 3))))
called dv3<	(LAM (LAM (LAM ((((PRFT 2) CALL) 1) 3))))
done tv3<	(LAM (LAM (LAM ((((PRFT 2) DO) 1) 3))))
done tv3	(LAM (LAM (LAM ((((PRFT 2) DO) 1) 3))))
eaten tv3<	(LAM (LAM (LAM ((((PRFT 2) EAT) 1) 3))))
eaten tv3	(LAM (LAM (LAM ((((PRFT 2) EAT) 1) 3))))
finished tv3<	(LAM (LAM (LAM ((((PRFT 2) FINISH) 1) 3))))
finished tv3	(LAM (LAM (LAM ((((PRFT 2) FINISH) 1) 3))))
got tv3<	(LAM (LAM (LAM ((((PRFT 2) GET) 1) 3))))
got tv3	(LAM (LAM (LAM ((((PRFT 2) GET) 1) 3))))
had tvh3<	(LAM (LAM (LAM ((((PRFT 2) HAVE) 1) 3))))
had tvh3	(LAM (LAM (LAM ((((PRFT 2) HAVE) 1) 3))))
heard tv3<	(LAM (LAM (LAM ((((PRFT 2) HEAR) 1) 3))))
heard tv3	(LAM (LAM (LAM ((((PRFT 2) HEAR) 1) 3))))
lost tv3<	(LAM (LAM (LAM ((((PRFT 2) LOSE) 1) 3))))
lost tv3	(LAM (LAM (LAM ((((PRFT 2) LOSE) 1) 3))))
put dv3<	(LAM (LAM (LAM ((((PRFT 2) PUT) 1) 3))))
put tv3<	(LAM (LAM (LAM ((((PRFT 2) PUT) 1) 3))))
put tv3	(LAM (LAM (LAM ((((PRFT 2) PUT) 1) 3))))
used tv3<	(LAM (LAM (LAM ((((PRFT 2) USE) 1) 3))))
used tv3	(LAM (LAM (LAM ((((PRFT 2) USE) 1) 3))))

It is not certain why in a minority of the clusters the AP clustering implementation yields outliers in the verb subset with 3 lambdas where the De Bruijn sequence of the outlier does not match the dominant sequence in the cluster. However, it is worth noting that the clusters for the verb subset with 2 lambdas always returns clusters with matching De Bruijn sequences. For examples of 2 of the 8 distinct clusters found for the verb subset with 2 lambdas, see Table 15 and Table 16.

Table 15

Cluster 4 of the 8 Clusters Found in the Verb Subset with Exactly 2 Lambdas

The Word and Its PoS Tag	DB Terms for The 2-Lambda Verb Entries in Cluster 4 of 8
done tv3	(LAM (LAM ((((SIMP 2) DO) 1) -)))
eaten tv3	(LAM (LAM ((((SIMP 2) EAT) 1) -)))
finished tv3	(LAM (LAM ((((SIMP 2) FINISH) 1) -)))
finish iv1	(LAM (LAM (((SIMP 2) FINISH) 1)))
got tv3	(LAM (LAM ((((SIMP 2) GET) 1) -)))
heard tv3	(LAM (LAM ((((SIMP 2) HEAR) 1) -)))
lost tv3	(LAM (LAM ((((SIMP 2) LOSE) 1) -)))
put tv3	(LAM (LAM ((((SIMP 2) PUT) 1) -)))
used tv3	(LAM (LAM ((((SIMP 2) USE) 1) -)))

Table 16

Cluster 5 of the 8 Clusters Found in the Verb Subset with Exactly 2 Lambdas

The Word and Its PoS Tag	DB Terms for The 2-Lambda Verb Entries in Cluster 5 of 8
done tv3	(LAM (LAM ((((SIMP 1) DO) 2) -)))
eaten tv3	(LAM (LAM ((((SIMP 1) EAT) 2) -)))
finished tv3	(LAM (LAM ((((SIMP 1) FINISH) 2) -)))
finish iv1	(LAM (LAM (((SIMP 1) FINISH) 2)))
got tv3	(LAM (LAM ((((SIMP 1) GET) 2) -)))
heard tv3	(LAM (LAM ((((SIMP 1) HEAR) 2) -)))
lost tv3	(LAM (LAM ((((SIMP 1) LOSE) 2) -)))
put tv3	(LAM (LAM ((((SIMP 1) PUT) 2) -)))
used tv3	(LAM (LAM ((((SIMP 1) USE) 2) -)))

The most important insight that was gained from the success of AP that is implemented with cosine similarity to cluster a subset of DBVs is that it seems that there may be a way to successfully use lambda terms in the vector space if the lambda terms are represented with De Bruijn indexing, which leads to the contention that De Bruijn sequences may be a way of linearizing the compositional structure in the state-of-the-art lambda terms.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this final chapter of the thesis, the results of the analyses that were laid out in the previous chapter are discussed and some future directions are given based on the findings of the work that was done in this thesis.

The work in this thesis focused on semantic representation phenomena in a language acquisition setting to come up with insights into whether or not a dimensionality reduction can be said to occur in the child's mind when acquiring the concepts in the language. Another question that was pondered was about how and how much of a reduction in the dimensionality may take place, if it indeed does. The thesis also aimed to utilize its exploration of semantic dimensionality such that insights would emerge regarding the role of semantics in the argument realization phenomena. Finally, the thesis aimed to produce insights into what kind of representational formalisms could be preferred if one is to represent the semantics of words, by way of comparing LTs to DBILTs.

By its investigation of the research questions summarized above, this thesis tried to make use of its formalization of the questions about the curse of dimensionality in language acquisition, in a manner that would serve as a window into how concepts may be represented in the mind once they are acquired either with a reduction in dimensionality (in case of the similarity comparisons of the reduced versus unreduced versions) or without a reduction (in the case of employing the AP clustering on the unreduced DBVs) but under distinct formalisms such as LTs and DBILTs.

5.1 Discussion of the Results

The results, both from the cosine similarity comparisons and from the AP clusters, seem to confirm the hypothesis that the De Bruijn indexed representations, particularly the ones with the arrays representing the lambda binders, are more robust than regular lambda terms in terms of their suitability for vectorial encodings that would allow for the dimensionality in the data set to be computationally reduced. The representations that utilized De Bruijn Indexing, namely the DBVs and the BDBVs, were found to be more robust representations for the purposes in this thesis in that they required fewer principal components compared to the LV representations while still sufficiently explaining the variance in the data. It could be argued that the dimensionality reduction during language acquisition utilizes semantic representations that are more like the De Bruijn

indexed ones than the ones with the regular lambda terms, on the basis that the ones with the De Bruijn indices are more effectively reducible to lower dimensions under linearity assumptions, but also under the non-linearity assumption brought by an additional reduction aided by t-SNE visualizations.

An important indicator of the robustness of the DBVs was the accuracy of the cosine similarity ratios they yielded in the comparisons of various word entries compared to the same ratios in both the BDBVs and the LVs. It was also worth noting how the reduced versions of the DBVs and the BDBVs started yielding comparable cosine similarity scores, and how it may be indicative of the De Bruijn sequences' prominence in reflecting the true dimensionality of the semantics of words.

When the AP algorithm was utilized to further investigate possible clusters in the overall DBVs data set and in two verb subgroups, the success of the clusters were seen to be dependent on the similarity measure that was used, and the success of the cosine similarity implementation for the AP seems to be in support of the contention that cosine similarity which was also used as a success metric of the 3 data sets via the pair-wise similarity scores was a good choice as a metric of the success of distinct types of vectorial representations for the semantics of words.

Coming to the question of the semantics' effect on the argument realization phenomena, the successful clusters found with the implementation of AP seemed to suggest that the structure of the semantics of verbs, even though they are compositional, if this compositionality is accounted for via a more linear representation (such as DBVs), may very well be tantamount in determining the syntactic phenomena of verbs where they go on to take a distinct number of arguments. Still, more work is needed, especially on the dimensionality reduction of syntactic representations to gain more insights about the argument realization phenomena, and into whether or not semantic versus syntactic bootstrapping is more likely during language acquisition.

One limitation of the work here is that since the thesis explored theoretical constructs for concept representation that are arrived at after a lot of abstractions under a certain formalism (lambda calculus), any evaluation of the results to the effect that the results may be reflective of how concept acquisition works and how dimensionality may be reduced is ultimately skewed by the assumptions in the encodings and in the layers of abstraction that the original data had been subjected to. Note that the selection of a formalism to represent semantic meaning and its later encoding ultimately comes with assumptions about the dimensionality in the data, which may not be reflective of the true dimensionality in the data. However, exploring certain reduction possibilities puts into perspective the features that may indeed be prominent in the true dimensionality of the data, one of which was claimed to be the De Bruijn sequences in this thesis.

The success of the clustering based on the vectors that were created with the De Bruijn sequences seem to suggest not only that the De Bruijn representations are more conducive to be used in computational exploration, but also that the systematic and compositional nature of the semantics of words could possibly be accounted for under the right formalism since the successful clusters bring together the words that are similar in their meaning in terms of the verbs' combinatory properties.

5.2 Future Directions

As stated earlier in the thesis, a future goal is to use the approach taken here to explore syntactic dimensionality reduction during language acquisition using the lexicon data for parts of the Eve fragment that was obtained from Şakiroğulları (2019), and combine insights from both syntactic and semantic dimensionality to make sense of argument realization phenomena. The results of this thesis indicate that the De Bruijn Index based representations seem to be efficient tools for bridging the gap between the study of semantics and its effect on the syntactic phenomena of argument realization, especially considering the results that were obtained from the cosine similarity measures and the AP clusters.

This thesis utilized a novel representational scheme to numerically represent lambda terms and their De Bruijn term equivalents. As the literature in Chapter 3 makes clear, there are frameworks in the literature that could possibly utilized in the future even though they were not aligned with the goals of this thesis regarding the direct use and exploration of lambda terms and De Bruijn terms. Trying to write the semantic representations in the data set in light of Category Theory to try out the method in Clark (2013) and Coecke et al. (2010) could be another future goal, as well as trying to build a vectorization framework where Smolensky (1990)'s Tensor Product Representations could be effectively utilized for the tree structures of the semantic representations. While the thesis used lambda calculus based meaning representations and their De Bruijn indexed rewrites, it is worth noting that various other techniques and formalisms (such as Lambek's (2008) pre-group grammar) that were explored in Chapter 3 which were not used in the methodology of the thesis could, in the future, also be explored in terms of their dimensionality and compared to the ones explored here after the groundwork that was done in this thesis.

An important limitation of the work in this thesis was that the analyses that were conducted were insufficient in that despite them being great feature extraction tools for reducing dimensionality and exploring the characteristics of the data sets, they were not tools for selecting features from the semantic representations. Since the work was one of feature extraction rather than feature selection, the reduced representations for the De Bruijn Terms or the Lambda Terms could not be made sense of with regards to their prior compositional properties because they could not be reverted back to a notation where they could be used in a parsing context for computations of semantics.

However, feature extraction can be seen as an achievement in that the data now could be used in vector space operations because it now has distributional semantic properties. Building from the previous point about, an important future direction in this research is to explore the distributional properties of the DBV representations by seeing if they have latent combinatory properties such that these vectorial versions of the original lambda terms can be composed with the application of a linear algebraic operation such as taking their tensor products, an experimentation the success of which could, again, be determined by the use of similarity metrics.

Bibliography

Abend, O., Kwiatkowski, T., Smith, N., Goldwater, S., & Steedman, M. (2017). Bootstrapping language acquisition. *Cognition*, *164*, 116–143.

Bellman, R. (1966). Dynamic programming. *Science*, *153*(3731), 34–37.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Block, N. (1986). Advertisement for a semantics for psychology. *Midwest Studies in Philosophy*, *10*, 615–678.

Borer, H. (2004). The grammar machine. In E. A. A. Alexiadou & M. Everaert (Eds.), *The unaccusativity puzzle: Explorations of the syntax-lexicon interface.* Oxford University Press.

Bozşahin, C. (2018). Computers aren't syntax all the way down or content all the way up. *Minds and Machines*, *28*, 543–567.

Bozşahin, C. (2021). *De bruijn index.* `https://github.com/bozsahin/debruijn-index`. GitHub.

Brown, R. (1973). *A first language: The early stages*. Cambridge, Massachusetts: Harvard University Press.

Brown, R., & Bellugi, U. (1964). Three processes in the child's acquisition of syntax. *Harvard Educational Review*, *34*, 133–151.

Chao, G., Luo, Y., & Ding, W. (2019). Recent advances in supervised dimension reduction: A survey. *Machine Learning and Knowledge Extraction*, *1*(1), 341–358. doi: 10.3390/make1010020

Chater, N., & Christiansen, M. H. (2018). Language acquisition as skill learning. *Current Opinion in Behavioral Sciences*, *21*, 205-208. (The Evolution of Language) doi: https://doi.org/10.1016/j.cobeha.2018.04.001

Church, A. (1936). An unsolvable problem of elementary number theory. *Journal of Symbolic Logic*, *1*(2), 73–74. doi: 10.2307/2268571

Church, A. (1941). *The calculi of lambda-conversion*. Princeton University Press.

Clark, S. (2013). Type-driven syntax and semantics for composing meaning vectors. In M. S. C. Heunen & E. Grefenstette (Eds.), *Quantum physics and linguistics: A compositional, diagrammatic discourse.* Oxford University Press.

Coecke, B., Sadrzadeh, M., & Clark, S. (2010). Mathematical foundations for a compositional distributional model of meaning.

Crouse, D., Willett, P., Pattipati, K., & Svensson, L. (2011, 08). A look at gaussian mixture reduction algorithms. In (p. 1 - 8).

de Bruijn, N. G. (1972). Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the church-rosser theorem. In (pp. 381–392). Amsterdam: Koninklijke Nederlandse Akademie van Wetenschappen.

Eilenberg, S., & Mac-Lane, S. (1945). General theory of natural equivalences. *Transactions of the American Mathematical Society*, *58*, 231—294.

Ellis, K., Dechter, E., & Tenenbaum, J. B. (2015). Dimensionality reduction via program induction. In *2015 AAAI spring symposia, stanford university, palo alto, california, usa, march 22-25, 2015*. AAAI Press. Retrieved from `http://www.aaai.org/ocs/index.php/SSS/SSS15/paper/view/10284`

Fodor, J. A. (1975). *The language of thought*. New York: Thomas Y. Crowell.

Fodor, J. A. (2008). *Lot 2: The language of thought revisited*. Oxford: Oxford University Press. doi: 10.1093/acprof:oso/9780199548774.001.0001

Fodor, J. A., & Pylyshyn, Z. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, *28*, 3–71.

Frey, B. J., & Dueck, D. (2005). Mixture modeling by affinity propagation. In Y. Weiss, B. Schölkopf, & J. Platt (Eds.), *Advances in neural information processing systems* (Vol. 18). MIT Press.

Gleitman, L. R., Liberman, M. Y., McLemore, C. A., & Partee, B. H. (2019). The impossibility of language acquisition (and how they do it). *Annual Review of Linguistics*, *5*(1), 1-24. doi: 10.1146/annurev-linguistics-011718-011640

Grimshaw, J. (1990). *Argument structure*. MIT Press.

Gödel, K. (1931). Über formal unentscheidbare sätze der principia mathematica und verwandter systeme i. *Monatshefte für Mathematik und Physik*, *38*, 173–198.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, *28*(3/4), 321–377.

Huang, A.-L. (2008). Similarity measures for text document clustering..

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. doi: 10.1109/MCSE.2007.55

Jolliffe, I. T. (2002). *Principal component analysis*. Springer Series in Statistics.

Katz, Y., Goodman, N., Kersting, K., Kemp, C., & Tenenbaum, J. (2008, 01). Modeling semantic cognition as logical dimensionality reduction.

Kwiatkowski, T. (2012). *Probabilistic grammar induction from sentences and structured meanings* (Unpublished doctoral dissertation). University of Edinburgh.

Lambek, J. (2008). *From word to sentence: A computational algebraic approach to grammar*. Italy: Polimetrica.

Lelewer, D. A., & Hirschberg, D. S. (1987). Data compression. *ACM Computing Surveys*, *19*, 261–296.

Levin, B., & Rappaport-Hovav, M. (2005). *Argument realization*. Cambridge University Press.

MacWhinney, B. (2000). *The childes project: Tools for analyzing talk, third edition*. Mahwah, NJ: Lawrence Erlbaum Associates.

Pearson, K. (1901, November). *LIII. On lines and planes of closest fit to systems of points in space.* Zenodo. doi: 10.1080/14786440109462720

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Piantadosi, S., Tenenbaum, J., & Goodman, N. (2012, 01). Bootstrapping in a language of thought: A formal model of numerical concept learning. *Cognition*, *123*, 199-217. doi: 10.1016/j.cognition.2011.11.005

Piantadosi, S. T. (2016). *Pychuriso: a python implementation of churiso.* `https://github.com/piantado/pychuriso`.

Piantadosi, S. T. (2021). The computational origin of representation. *Minds and Machines*, *31*, 1–58.

Sagae, K., Davis, E., Lavie, A., & Macwhinney, B. (2010). Morphosyntactic annotation of childes transcripts.

Schönfinkel, M. (1967). On the building blocks of mathematical logic. *From Frege to Gödel*, 355–366.

Searle, J. R. (1980). Minds, brains, and programs. *The Behavioral and Brain Sciences*, *3*, 417–457.

Searle, J. R. (1990). Is the brain a digital computer? *American Philosophical Association Proceedings*, *64*, 21–37.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, *46*, 159–216.

Steedman, M. (1996). *Surface structure and interpretation*. MIT Press.

Şakiroğulları, Ç. (2019). *Measuring Empirical Bias Toward Ergativity and Accusativity* (Unpublished master's thesis). Middle East Technical University.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*(86), 2579–2605.

van Rossum, G. (2020). *The python library reference, release 3.8.2*. Python Software Foundation.

Waskom, M. L. (2021). Seaborn: statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. doi: 10.21105/joss.03021

Wittgenstein, L. (1953). *Philosophical investigations*. Oxford: Blackwell. (Translated by G.E.M. Anscombe)