DISTRIBUTIONAL GRAPH: CONNECTING LANGUAGE TOWARDS A REPRESENTATION OF KNOWLEDGE AND MEANING

BY

SHUFAN MAO

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Psychology
in the Graduate College of the
University of Illinois Urbana-Champaign, 2023

Urbana, Illinois

Doctoral Committee:

Assistant Professor Jon Willits, Chair
Professor Cynthia Fisher
Professor Kara Federmeier
Assistant Professor Jessica Montag
Professor Emeritus Gary Dell

Abstract

What is the nature of the relationship between Knowledge and Language? How does knowledge representation interact with language use? The critical role of world knowledge (lexical semantic) in language processing has long been noticed by psycholinguists, but relatively less attention has been paid to incorporating rich linguistic information in modeling knowledge and semantic representations. Current distributional models do form semantic representations from linguistic input. Despite their success in representing relatively simple semantic relations, these models are less effective in extracting rich linguistic structures from corpus, resulting in limited capability in representing complex lexical dependencies, achieving challenging semantic tasks, and accounting for relatively involved semantic behaviors. In this dissertation, I develop a novel type of distributional model – Distributional Graph – that transforms raw linguistic input into graphical forms and connects the graphlets to build a semantic network. The model encodes distributional patterns of linguistic units by a graphical topology, so that linguistically expressed concepts can be evaluated by network metrics. In particular, I adopt a spreading activation algorithm that gives rise to a graded measure of semantic relatedness on the network. I show, with two groups of studies, that distributional graphs equipped with spreading activation may better represent complex lexical relationships, compared to existing distributional models. In particular, I show that (i) The graphical encoding of co-occurrence leads to effective representation of indirect semantic relations which facilitates generalizing word-word lexical dependencies, and (ii) The explicit encoding of constituent structures in Distributional Graph leads to effective representation of multi-way lexical dependencies and success in compositional generalization tasks. The modeling works are conducted on artificially generated corpora with controlled distributional constraints, leading to a clear mechanistic account for the formal capabilities of the model. Meanwhile, the modeling results have profound implications on theory of language cognition and knowledge development in human. Considerable amount of behavioral studies are needed to validate Distributional Graph as a model for human semantic memory, and the experimental works require scaling up the Distributional Graph approach with naturalistic linguistic input. For the long-term vision, by integrating multi-modal inputs and finer grammatical information, the more full-fledged distributional graphs may give rise to generation of novel concepts that are both grammatical and meaningful. Importantly, as distributional graphs are based on explainable computational mechanisms, such advance may contribute to more interpretable AIs, and further the understanding of knowledge development and innovation in humanity.

To the Universe.

Acknowledgments

This project would not have been possible without the support of many people. To start with, I would like to thank my thesis committee members: Dr. Jon Willits, Dr. Cynthia Fisher, Dr. Kara Federmeier, Dr. Jessica Montag and Dr, Gary Dell, who provided guidance along completing the dissertation. In particular, I want to emphasize the contributions of Cynthia Fisher and Jon Willits. Cynthia Fisher is the first faculty member knowing the Distributional Graph model, and she helped me elucidating the essential research goal concerning the modeling endeavor through several long and thorough conversations filled with challenging questions. Jon Willits is my graduate advisor who supported the whole dissertation research. His guidance helped carving out the research ideas and parsing them into projects. His openness to different possibilities and perfectionist attitude towards scholarly works led to the high quality publication of the first project in the dissertation (Chapter 3).

Secondly, I am grateful to all researchers who directly or indirectly influenced me academically, and forged my research interests. Most directly, a group of outstanding psychologists, linguists and mathematician at UIUC and UC Berkeley equipped me with a solid foundation in cognitive science. They are Gary Dell, John Hummel, Cynthia Fisher, Dan Hyde, Jon Willits, Jessica Montag (Psychology, language processing and acquisition, cognitive science, math cognition, semantic memory, reading); Peter Jenks and Peter Lasersohn (Linguistics, formal syntax and semantics); Anush Tserunyan (Math, mathematical logic).

More broadly, a number of linguists, psychologists, mathematicians and philosophers largely sculptured my research goals: Gottlob Frege, Noam Chomsky, Richard Montague, and Jerry Fodor convinced me that there is structure in Language and Thought; Jeffery Elman, James mcclelland, and Paul Smolensky showed me how structures in language could be captured in Connectionist representations; John Firth, Susan Duimas, Thomas Landauer, Curt Burgess and Micheal Jones directed my attention to distributional models; and Ross Quillian, Allan Collins, Elizabeth Loftus and John Anderson paved my way to modeling knowledge and meaning with semantic networks.

On the highest level, I would acknowledge two giants in science who had inspired me since when I was seventeen, and brought me to the fascinating field of psychology, the study of human mind. The first figure is Sir Isaac Newton, who demonstrated through his life's endeavor that formal theory can be used to formulate, explain and predict the phenomena in the physical world. His efforts showed the whole humanity that we may know the world better through systematic formal reasoning. The second figure is Dr. Wilhelm Maximilian Wundt, the founder of modern experimental psychology, and the pioneer of extending the objective scientific investigation in the physical world to the fascinating mental universe. Their existence encouraged me to pursue formalizing behaviors and cognition, in my doctoral career and my future academic life.

Lastly, this 8-year scholarly journey could have not been completed without the support from people nearby. I appreciate the accompany of all my lab mates in Learning and Language Lab at UIUC. Especially, I need to mention Phillip Huebner, who is the major collaborator of all projects in the thesis research. There have been thoughtful discussions and debates between us, which led to the high quality of the works. I would thank Jacki Erens, Suyeon Hwang and Belgin Unal, my cohort in the cognitive division. We entered the department together (as first-year psychology graduate student) and walked through the 'unusual' graduate career spanning the pandemic. At the very end, I must highlight my families who backed me up through the whole process. The academic expedition could not have been started without their economic and mental supports. Their encouragement through the eight years have largely contributed to this dissertation, and will probably motivate potential outcomes in my future research.

Table of contents

List of Abbreviations

DG	Distributional Graph
LON	Linear Order Network
CTN	Constituent Tree Network
HAL	Hyperspace Analogue to Language
LSA	Latent Semantic Analysis
RNN	Recurrent Neural Network
LLM	Large Language Model
GPT	Generative Pre-trained Transformer
SR	Semantic Relatedness

List of Symbols

ω	Weight of edge in networks.
P	Path in networks.

Chapter 1

Introduction

1.1 Overview

How people represent knowledge and understand language have been the central topics in cognitive science. On one hand, theoretical discussions and empirical works suggest that representation of (world) knowledge can be crucial to language comprehension (Langacker, 2008; McRae, Hare, Elman, & Ferretti, 2005; McClelland, John, & Taraban, 1989; Willits, Amato, & MacDonald, 2015). As an example, to understand the English sentence *Mary cut the cake with the knife*, one needs to first grasp its semantic structure, i.e. answering to the 'Who did what to whom?' type of questions, e.g. *Who cut the cake?*, *What did Mary cut?*, and *What did Mary do to the cake?* But more importantly, to fully understand the sentence, one needs to know the **semantic contents** of the lexical terms on top of the semantic structure. We need to know about the manner of the verb *cut*, the features of the objects *cake* and *knife*, and the fact that *knife* is a canonical tool to cut food. In this way, we can tell that the sentence describes a plausible event, in contrast to less plausible sentences like *Mary was cutting orange juice.* or *Mary cut the tree with the carpet/comedy show.* In this case, the (world) knowledge about the lexical items, e.g. *cut, cake, knife*, is essential for comprehending natural language.

While it is broadly admitted that (world) knowledge matters a lot to language, it should be noted that language plays a big role in knowledge representation. First of all, we need language to 'describe' knowledge. Outside in the world, intuitively, a great portion of knowledge is declarative and expressed with language, from Brownie recipes to philosophy theses. Stepping into the theoretical domains, in almost all theories and models of knowledge/concept/semantic memory representations, the basic units to represent are mostly linguistic expressions such as words, phrases and even clauses. Moreover, modern AI practice have also shown that models exclusively trained on language input like GPT-3 (Brown et al., 2020) incorporate enormous amount of world knowledge. These points can be easily accepted, as the semantic contents in language are generically world knowledge. However, what has been less attended to is the other component of language. As mentioned earlier, language has not only semantic contents, but also structures (semantic and syntactic structures). In addition to the contents, how do the language structures contribute to the representation of knowledge? As the dissertation is developed, I will show that the semantic content and language structure interacts with each other and collectively contributes to better knowledge representations.

The big-picture goal of this dissertation is to bridge the endeavors of knowledge representation and language comprehension. To be more specific, the goal here is to develop a computational model of knowledge/semantic memory representation, such that (1) the model may better support the mental process of language comprehension, and (2) the formation of the representational structure in the model is psychologically grounded. As I will review in detail, the classic semantic theories and models (Anderson, 1983; Collins & Quillian, 1969; Collins & Loftus, 1975; Osgood, 1952; Smith, Shoben, & Rips, 1974) and the more recent distributional approaches (defined later) (Landauer & Dumais, 1997; Lund & Burgess, 1996; McClelland et al., 1989; Mikolov, Chen, Corrado, & Dean, 2013) have respective issues towards the general goal. They either have trouble effectively representing the essential semantic knowledge necessary for language comprehension or fail to explain how the structure could have been formed from available input, or both. I argue that the deficiency of the existing models may arise from the less ideal representational form and encoded information.

The deficiencies in existing semantic models lead to my proposal of a novel modeling approach towards knowledge/semantic representation. As a preview, it encodes distributional linguistic data, in graphical (i.e. network) structures for knowledge representation, referred to as 'Distributional Graph'. In this type of models, certain kinds of linguistic co-occurrence (encoded information) are taken as building blocks to construct graphical structures (representational form). More abstract semantic relations can be abstracted through network techniques, e.g. spreading-activation, to generate knowledge/information crucial for language comprehension. To briefly summarize, the proposed approach constructs a knowledge structure from linguistic data in a graphical form, such that the 'language' (semantic) information encoded in the graphical representational structure can in turn support the process language understanding.

In the following chapters, I present studies and formal discussions to show how the graphical representational form and the distributional linguistic information collectively lead to effective knowledge representations towards comprehension. The rest of chapter 1 motivates the Distributional Graph proposal. In section 1.2, I claim that lexical relationship is one type of knowledge necessary to language comprehension. I review in section 1.3 a selective set of classic semantic models, recognizing their achievements while stressing what they lack for the purpose of language understanding. In Section 1.4 and 1.5, I review more recent distributional semantic models and spreading-activation based graphical models. I argue that the two approaches have complementary pros and cons, and integrating them may give rise to a model better supporting language comprehension. This leads to the proposal of Distributional Graph as an integration of the two lines of works in Chapter 2, where I develop a formal theory. Section 2.1 defines the spreading-activation based measure of semantic relatedness in distributional graphs. Section 2.2 and 2.3 introduce two types of distributional graph: Co-occurrence Network and Constituent Tree Network respectively. Chapter 3 and 4 examine the capability of Co-occurrence Network in representing graded indirect semantic relatedness. Chapter 3 disentangles the contribution of representational form (graph vs. vector space) and encoded information. Chapter 4 provides a theoretical account of the advantages of graphical form in representing indirect semantic relations, by formally establish the structural and processing equivalence between graph and vector space structures. Chapter 5 and 6 focus on the role of language (constituent/syntactic) structure in representing multi-way lexical relationships (defined later). By testing in Chapter 5 both CTN and the co-occurrence network on formally designed compositional generalization tasks, I show the important role of constituent structure in representing multi-way relations. In Chapter 6, I further explore the necessary computational mechanisms for representing multi-way relations by conducting the similar set of compositional generalization tasks on various semantic model, and by comparing them with CTN. I discuss the theoretical and practical implications of the Distributional Graph and future directions in Chapter 7 before concluding in Chapter 8.

1.2 Lexical Relationship: Semantic Knowledge Necessary for Language Comprehension

Understanding sentences may require multiple types of knowledge. For example, to understand *Mary cut the cake with the knife*, one needs to know **linguistic knowledge** such as how to parse the sentence, and the argument structure of the main verb *cut* (e.g. how many arguments it takes); as well as **world knowledge** such that CAKE is something that can be cut, KNIFE is a canonical tool for cutting foods, and the physical properties of CAKE and KNIFE. In this section, I specify a portion of these knowledge and group them under the overarching label 'lexical relationship' (or lexical relation). I argue that these are the knowledge necessary and essential for language comprehension, and will be the focus of this dissertation. In the literal sense, 'lexical relationship' should refer to relation between lexical items/terms. I will specify what do I mean by 'lexical', and what type of lexical relationship is of concern here. While traditionally, 'lexical level' knowledge is considered as 'world knowledge' (Chomsky, 1965), I will show that the overarching 'lexical relationship' specified in this context involves both world knowledge and linguistic knowledge. Moreover, I claim that integrating world and linguistic knowledge may give rise to semantic representation that facilitates language comprehension, a point that will be thoroughly examined and discussed in Chapter 5, 6 and 7.

1.2.1 Defining 'Lexical'

Lexical effects in language comprehension have long been noticed and studied: The lexical choice of one or more thematic roles (as components of a sentence) affects the lexical choice of other components of the sentence, e.g., ***mechanic*** *check* ***engine*** vs. ***journalist*** *check* ***spelling*** (Bicknell, Elman, Hare, McRae, & Kutas, 2010). In turn, the dependency between lexical choices within a sentence may influence structural parsing (Garnsey, Pearlmutter, Myers, & Lotocky, 1997; Trueswell, Tanenhaus, & Kello, 1993; Trueswell, Tanenhaus, & Garnsey, 1994; MacDonald, Pearlmutter, & Seidenberg, 1994) or semantic integration of the whole sentence (Bicknell et al., 2010; Ferretti, McRae, & Hatherell, 2001; Kamide, Altmann, & Haywood, 2003; Matsuki et al., 2011; Rayner, Warren, Juhasz, & Liversedge, 2004). Notice that in the studies above, the use of 'lexical' emphasizes a sense of 'lexical choice' over more abstract categories: the 'lexical relationship' concerned is more about the fit between specific 'lexical entries'. To illustrate, consider the 'lexical relationship' between *cut* and *cake*, the interest here is on the relation between the two particular lexical concepts CUT and CAKE, rather than that between a verb and a noun, or between a general type of action and a type of food. In this sense, I treat *cut-cake* and *cut-tree* as two separated relations, or two separated set of relations, and care about the difference between the relations sensitive to the varying lexical terms (*cake* vs. *tree*). On the other hand, the 'lexical relationship' specified here does not restrict to relation between word pairs. It can be multi-way relations between three or more words, between a phrase and a word, or between phrases. What matters is that, these relations have to be down to the lexical level, e.g. it has to be between the lexicalized phrases and words, such as *cut cake* and *knife*, rather than abstract categories such as 'verb phrase' and 'noun' or 'a *cutting* verb phrase' and an 'instrument'.

1.2.2 The Lexical Relationship of Primary Concern

After clarifying about the entities in relation, I specify what type of relationship is of concern. There are at least two ways to partition lexical relationships. One partition concerns how the lexical terms co-occur with each other, i.e. syntagmatic vs. paradigmatic. The other partition concerns how we can evaluate the relationships, i.e. quantitative vs. qualitative. In this dissertation, I primarily focus on the quantitatively evaluated syntagmatic relations. I first introduce the two partitions and then explain why quantitative evaluation on syntagmatic relation is of interests. That is, how this type of lexical relationship matter to language comprehension.

Quantitative vs. Qualitative

The quantitative aspect focuses on how closely two concepts are related, or similar to each other, referred to as 'semantic relatedness/similarity'. Throughout the dissertation, I may use **lexical dependency** as an approximate term for quantitative lexical relationship. On the other hand, qualitative aspect is concerned with the type of relation, e.g. synonym, antonym etc. Bejar, Chaffin, and Embretson (1990) have summarized a wide range of types of qualitatively evaluated relations and created a taxonomy for them. Some examples are: *Contrary* (e.g. *hot:cold*), *Taxonomic* (e.g. *flower:tulip*), *Item:Attribute* (e.g. *beggar:poor*), *Action:Object* (e.g. *plow:earth*) and *Cause:Effect* (e.g. *joke:laughter*). Notice that the partition merely reflects different perspectives on lexical relations and they may simultaneously apply to the same concept pair. For example, *asleep* and *awake* are antonyms from the qualitative perspective, however, the pair can as well be evaluated quantitatively on their relatedness/similarity.

Syntagmatic vs. Paradigmatic

Historically, there are different ways to refer to the two broad classes of relation: associative or syntagmatic vs. semantic or paradigmatic, etc. While different terminologies vary by the exact classification standard, they share the sense that one type of relations refer to concept pairs that may occur together in the same sentence or context, taking different thematic roles (functions) and collectively contribute to a holistic meaningful scene (associative, syntagmatic). For example, the relation between *cut*, *cake* and *knife* in the sentence *Mary cut the cake with the knife.* is associative or syntagmatic, as they co-occur and collective make up a meaningful event. The other type of relations (semantic, paradigmatic) apply to pairs that tend to have the identical thematic role (function) in events, so that one can be substituted with the other across contexts. As in the *cut cake* example, the noun *cake* can be replaced by *pie*, and the two words usually have the identical thematic role across scenes, so that the pair has a semantic/paradigmatic relation. From now on, I will use 'syntagmatic' and 'paradigmatic' to refer to the two classes of relations.

It is useful to take into consideration the 'syntagmatic/paradigmatic' partition when viewing lexical relationship qualitatively or quantitatively. Viewed qualitatively, almost all kinds of semantic relations are either syntagmatic or paradigmatic. For example, in the relation taxonomy (Bejar et al., 1990), *Contrary* (e.g. *hot:cold*) and *Taxonomic* (e.g. *flower:tulip*) are paradigmatic, while *Item:Attribute* (e.g. *beggar:poor*), *Action:Object* (e.g. *plow:earth*) and *Cause:Effect* (e.g. *joke:laughter*) are syntagmatic. Viewed quantitatively, the notion of semantic relatedness has been conflated with similarity (Tversky, 1977). To me, the relatedness vs similarity difference reflect the syntagmatic vs. paradigmatic partition. The intuition follows that relatedness refers to the general extent of being related, so that it is the quantification of lexical relationships in general. On the other hand, similarity is a more specific type of relatedness that fits better for 'replaceable' concepts, and therefore the quantification of paradigmatic relationships in particular. Illustrated with an example: the action *cut* is related to *cake*, but the action is by no mean similar to the object. On the other hand, *pie* is both related and similar to *cake*. In this dissertation, 'relatedness' will be differentiated from 'similarity', and more relevant investigations can be found in Chapter 3, 4.

Why Syntagmatic Relationship Evaluated Quantitatively

The interest in syntagmatic relationship is straightforward: The primary focus in this dissertation is the lexical relationships within a sentence. By definition, these are syntagmatic, but paradigmatic relations. However, paradigmatic relationship can be relevant when we need to construct indirect syntagmatic relationships for generalization, and I will get back to this later in the section. Concerning the quantitative vs. qualitative partition, both perspectives are significant to comprehension. While the quantitative perspective provides a generic evaluation of holistic plausibility, the role of qualitative relationships such as 'antonym', 'synonym' can be irreplaceable in certain circumstances. For example, to make sense of the sentence *Mary thought the cake was too small and bought a bigger one*, it is essential to know that *big-small* is a pair of antonyms. In particular, no quantitative relatedness can replace the role of antonym relation in this scenario. Given their theoretical significance, qualitatively evaluated lexical relations should worth independent modeling endeavors, and a thorough examination on the topic is beyond the scope of this dissertation. The studies presented here will primarily focus on quantitatively evaluated lexical relationships.

1.2.3 Multi-way Relations and Linguistic Structure Knowledge

As emphasized earlier, I investigate lexical relationships between both word pairs (two-way), in Chapter 3, 4 and between multiple words (more than two, multi-way) in Chapter 5, 6. Two-way lexical relationships are basic. They occur in almost all sentences and have been the dominating subject in knowledge/semantic representation studies. Nevertheless, a great portion of sentences contain three or more lexical terms with mutual dependency that do not reduced to two-way relations. An example would be *She cut the cake* vs. *She cut the cake with the knife*. In the former sentence, there are only two lexical terms and therefore there is only one two-way lexical relationship, i.e. *cut-cake*. However, the latter sentence involve three lexical terms *cut, cake*,and *knife* that interact with each other. Importantly, as I will discuss with more details in 5, the relationship between the three words does not reduce to two-way relations in a straightforward sense. This phenomenon has been noticed by psycholinguists for long, and studies have shown that the multi-way lexical relationship affect people's understanding of the sentences (Bicknell et al., 2010; Kamide et al., 2003; Rayner et al., 2004). Unfortunately, the topic has been less explored in knowledge/semantic representation modeling, and as a result, most of semantic models are not able to handle multi-way lexical relationship (which I will show in Chapter 6).

While sharing some similarities, multi-way relationship contrast to two-way relation by the **structure** it requires. It is simple for a two-way relation, as there are only two words and one single relation. But as long as we get to the number '3', it becomes exponentially harder even to explicate the multi-way relation. What does it mean for a three-way relation between *cut, cake*,and *knife* in the sentence *Mary cut the cake with the knife*? In a sense, people might phrase this as: the lexical choices of any two roles/slots affect the lexical choice of the one left (or less intuitively, the other way round). Specifically, this says: the choices of the two lexical items *cut* and *cake* on the verb and the object slots influence the choice of *knife* on the instrument slot. As we frame the situation, we have already started to build up some structure. In this case, the relationship between three words is reformulated as the relationship between a two-word verb phrase and a noun, that is, some syntactic structure. In other words, it seems that we generically chunk the words into some structure when formulating and representing the multi-way lexical relationships. These structures can be semantic or syntactic, which are usually considered as 'linguistic knowledge' (Chomsky, 1965; Jackendoff, 2002). The point is, these linguistic knowledge concerning language structure seems to be useful when we try to analyze multi-way lexical relationships. This leads to a central argument of this dissertation (in contrast to many other semantic modeling works): Linguistic knowledge such as syntactic and semantic structures might be necessary to incorporate into knowledge/semantic representations, as long as these representations target not only at two-way lexical relations, but also multi-way lexical interactions in more complex sentences. I examine the role of linguistic structure in semantic representation in Chapter 5 and 6.

1.2.4 Generalization and Paradigmatic Relationship

Although I primarily examine syntagmatic relationships, paradigmatic relationship can be important as well. This is due to their role in constructing 'indirect' syntagmatic relationships and generalization. There are some syntagmatic relations we are more familiar with, as the lexical terms frequently co-occur in sentences. However, most of possible syntagmatic relations are rarely observed. For example, we are familiar with *cut cake* but not as much to *cut cookie* and even less to *cut juice*; We easily accept *cut cake with knife* compared to *cut cake with saw* or *cut cake with shoe*. Regardless of familiarity, it is intuitive that we can judge on the plausibility of these lexical combinations, in other words, evaluating these syntagmatic relations. How does this happen? Why we may find *cut cookie* better than *cut calibration*? One possible answer to the question is that we generalize from what we are familiar with to the unfamiliar ones: We generalize from *cake* to *cookie* and *juice*. In this case, paradigmatic relationship play a role: it would be better if a model form an effective scale on the similarities between the lexical terms. For example, *cake* should be more similar to *cookie* than to *juice*. It turns out that effectively representing both paradigmatic and syntagmatic relationships is a challenging task. I investigate this topic thoroughly in Chapter 3 and 5.

1.2.5 Summary

This dissertation mainly focuses on modeling syntagmatic lexical relationships that are quantitatively evaluated. Multi-way syntagmatic relations might require incorporating linguistic (structure) knowledge into the representation, so that it may account for sentence comprehension involving interaction of three or more lexical terms. Paradigmatic relationships will be attended to as they may help inferring on unfamiliar syntagmatic relations. In the next section, I review previous semantic models by focusing on their ability of representing the interested type of lexical relationships.

At the end of this section, I leave a final note about the lexical relationship specified in this section. The knowledge about lexical relationships is necessary but sufficient to language comprehension. Aside from the qualitative aspect of the relations, we need to note that these lexical relationship knowledge is probably the most **superficial** part of world knowledge. For example, consider the lexical relationship *cut-cake*. All we say here is that *cut* and *cake* may bear a strong syntagmatic relationship (by quantitative measure), such that they are very likely to co-occur in the same sentence to collectively build up a meaningful event. However, there are more world knowledge about why there is such relation. To know this, one needs to know about the exact manner of the verb *cut*, and some feature of CAKE such as it is a concrete object, and it normally has soft texture. While it is very likely that the evaluation on these detailed world knowledge give rise to the lexical relation *cut-cake*, it is still fair to say that the mere knowledge of the lexical relationship itself may directly contribute to the understanding of the sentence. Therefore, I consider the lexical relationship specified in this section a lower bound of the set of knowledge that we need for language comprehension. While 'deeper' world knowledge definitely worth scholarly endeavors, I start from the superficial layer, the interface between world knowledge and linguistic knowledge.

1.3 Classic Semantic Models

A series of works has carved out the landscape of Semantic Representation research: The multi-dimensional space in Osgood (1952) have provided the representational prototype for modern distributional semantic model; Semantic network proposed in Collins and Quillian (1969) is the pioneer for the subsequent graphical models; and the feature comparison model in Smith et al. (1974) is the first attempt modeling semantic memory with feature. I review these earlier models with the focus on how they represent and access to lexical relationships, and how these foundational works have influenced later semantic models. However, while the classic models have established the framework of semantic/knowledge representation, they are limited in accounting for more complex language comprehension. Moreover, they are also less explicit about how the semantic structures could have been formed in a psychologically grounded way. I comment on these limitations to elicit the discussion on more recent semantic models.

1.3.1 Semantic Differentiation

The semantic differentiation (Osgood, 1952) model is one of the earliest model to represents concepts in a multi-dimension vector space. Osgood (1952) constructed a multi-dimensional semantic space, in which the dimensions are scales on selected polar pairs such as *good-bad* and *cold-hot.* In the initial attempt, 50 such polar pairs were selected as the scale dimension. Concepts were rated by human participants on the 50 dimensions, such that each concept is represented as a point in the multi-dimensional semantic differentiation space, with the coordinates indicating its goodness, coldness, etc. These scaling dimensions can be considered as the 'features' used to evaluate the concepts, and in this sense the representation is similar to that in the Feature Comparison Model (Smith et al., 1974), which will be introduced right after.

A primary 'use' of the semantic differentiation model is to evaluate lexical relationship. As an illustration, Osgood (1952) compared two words: *eager* and *burning.* It turned out that *eager* had higher value for 'good' and 'fresh', while burning was relatively 'hot' and 'dry'. In other word, the semantic differentiation model indicates that *eager* is 'fresher' than *burning* while *burning* is 'dryer' than *eager.* The comparison essentially concerns the relation between *eager* and *burning*, and in the model, the relation is derived from the coordinates of the concepts in the relevant scaling feature dimensions.

1.3.2 Feature Comparison

The feature comparison model (Smith et al., 1974) is essentially a semantic processing model/theory motivated by answering questions like *Is robin a bird?* or judging whether the proposition *A canary can fly* is true. These tasks can be considered as evaluation of the relation between lexical concepts (*robin* and *bird*), and the relation between a concept *canary* and a feature *can fly.* In the Model, concepts are associated with defining and characteristic features, which are compared in separated stages to judge a proposition (stage 1 for characteristic features and stage 2 for defining features). The hypothesis is, if two concepts share a lot of characteristic features, e.g. *robin* and *bird*, judgements can be made within stage 1. People need to go to stage 2 and compare defining features when the characteristic features do not align well, e.g. *penguin* and *bird.*

Concerning the representation, although Smith et al. did not explain how concepts are represented relative to the features, it was suggested that the concepts were vectors on the feature dimensions. In this way, the representation was similar to that in Semantic Differentiation. However, it was different from its precedent in the way that the concepts were not compared separately on the 'feature' dimensions, instead, a similarity score was computed and taken as the measure of the distance between the concepts. In turn, these similarities were used to account for processing time in semantic decisions. For example, as *robin* shares many characteristic features with *bird*, their similarity score would be high. As a result, the judgement *Is robin a bird* can be made after the stage 1 process (by only comparing characteristic features).

1.3.3 Semantic Network

Collins and Quillian (1969) aimed at similar semantic judgements as Smith et al. (1974). However, they represented the concepts in another structure, i.e. a discrete hierarchical tree, making it the first graphical representation of semantic memory. Collins and Loftus (1975) reformed the hierarchical tree into more general network structures. These models have represented the semantic information in a graph, where concepts are encoded as nodes, and semantic relations as links between the concept nodes. To be more specific, Collins and Quillian (1969) encoded concepts (which are also categories) and their subcategories as parent and child nodes on the tree, linked by a directed edge representing the IS_A relation, (e.g. concept *canary* is linked to *bird* with the edge IS_A). Moreover, each concept (category) and the features shared by all instances in the category (e.g. concept *canary* is associated with feature *can sing*). In the later version of semantic network (Collins & Loftus, 1975), nodes still represent concepts, which are connected by undirected edges bearing similar 'qualitative' relationships.

Similar to Semantic differentiation and Feature Comparison model, there is a way to evaluate the quantitative relation between concepts in a network. The intuition is that, concepts that are 'closer' in the network should be more related to each other, and they should be more likely to be retrieved and processed together. For example, in the hierarchical tree structure (Collins & Quillian, 1969), *canary* is closer to *bird* than to *animal*, and therefore bird features such as *can fly* and *have wings* are accessed faster than general animal features like *can breathe*.

1.3.4 Representational Structure and Access to Semantic Relations

The early models are different in terms of representational structure: **vector space** for Semantic Differential and Feature Comparison Model; **graph** for semantic network. Almost all subsequent models have adopted either of the two representations, or can be considered as a mixture of the two (Connectionist models). The two representational forms have led to two ways of accessing semantic information. In graphical models, qualitative relational labels can be explicitly marked on the edges (Collins & Loftus, 1975), and quantitative relations are accessed by network measures (e.g. graphical distance). Specifically, Collins and Loftus (1975) proposed an algorithm for accessing information and evaluating lexical/semantic relationships: the spreading-activation process. Activation starts from a source node, and spread to neighboring nodes constantly through the links. In this way, concepts are accessed in the order by their distance to the source node, and so are the semantic relations encoded in the links. A pair of concepts are linked by a 'composite relation' between them, which is an assembly of all possible qualitative semantic relations the pair can bear. In vector space models (from now on referred to as spatial models), quantitative relations such as semantic relatedness or similarity can be defined by vector metrics.

Regardless of the representational structure, the early modeling endeavors have targeted on the paradigmatic relationships between concepts/categories. The constructed knowledge structure and the information access process can account for human behavior data in simple semantic tasks, such as answering 'Is robin a bird?'. However, these models may not account for phenomena involving more complex language. Meanwhile, they focused primarily on the structure and process *per se.* without attempts to explain where the structures are from. Before introducing how these issues are addressed by more recent models, I briefly expand on this limitation of early models.

1.3.5 Limitations of Classic Semantic Models

The first major limitation of the early models is their restrained account for broader language processing, as the models have neglected words other than nouns, and more complex linguistic expressions. In Semantic Differential and semantic network, the represented concepts were primarily nouns, more precisely 'entities'. Verbs and adjectives were usually taken as predicative features (e.g. *can sing* and *have wing* for *canary*). They were used to represent the nouns, while the representation of the predicates was not the focus. [1] However, from a more linguistic view [2], predicates such as verb and adjective should not be considered as mere features of nouns/entities. They are meaningful on their own, and important constituent of linguistic expressions. If the goal is to represent knowledge not only concerning entities, but also meaning of natural language as a whole, at least all open class words should be included in the representational structure. Moreover, the notion of concept should not be restricted to words, but also complex expressions like phrases, e.g. *cut cake*. They should be included in semantic models as they incorporates linguistic structures on top of the word-level information, and directly contribute to sentential meaning. Without inclusion of open class words and complex linguistic expression, it is impossible for a semantic representation to account for general language comprehension.

Another issue in the early models is that they have no attempt to ground the formation of the structure by any psychological process. The semantic differentiation model formed the representation based on human's rating. The associative semantic networks (Collins & Quillian, 1969; Collins & Loftus, 1975) and the defining/characteristic features of the concepts (Smith et al., 1974) were both hand-crafted. While the structures aimed at how knowledge are represented in mind, the psychology theory about development of such structure was placed secondary. Nevertheless, the missed aspect is essential for a primary reason: A theory of how the structure could have been formed based on input can be informative on the exact form of the end-state representation. It is possible that some representations may never be formed due to the limited input or learning mechanism.

These fundamental drawbacks have been addressed, to different extents, by a recent line of works, i.e. **Distributional Semantic Models**. These models resemble Semantic differential in the way that concepts are represented in a vector space. The contrasting difference is that distributional models are trained on naturalistic linguistic corpus by doing statistical counts and predictions, and the semantic representation formed along the process reflects the distributional features of the concepts (words). In this way, all distributional models have a clear story of how semantic structure may have been formed from the raw input. Furthermore, as distributional models are trained on linguistic corpus, there is no principal constraint on the concept (word) to represent. Compared to the hand-crafted precedents which are at most conceptual framework of human knowledge, distributional models have the potential of representing any concept with a linguistic expression, and accounting for more complex phenomena in language comprehension.

[1] An exception is the Semantic differential model, in which both entities (e.g lady) and predicates (e.g. polite) are represented in the multi-dimensional space

[2] The early models primarily focused on concept, rather than word, as the former usually embodies multi-modal information and is what knowledge is about, while the latter is often considered as a linguistic expression of the former. On one hand, this view is well justified: Linguistic expression is only one facet of concept, therefore should not be conflated with concept itself. On the other hand, the neglect has led to limited range of concepts concerned in the models, i.e. concepts that are expressed by a single noun or relatively simple linguistic expression.

1.4 Distributional Semantic Models

Distributional Semantic Models form semantic representation of words or larger linguistic units based on their distribution pattern of in textual corpus. The distributional pattern may refer to how a word co-occur with other words, or in which pages the word has occurred within the corpus in the encoding phase. After certain abstraction processes, words are represented as a point in the abstract distributional vector space, in which each dimension can be considered as an abstract distributional feature reflecting how words co-occur with each other in the corpus. In this way, distributional models share the vector space representational form with Semantic Differential and Feature Comparison Model. What they drastically differ from the earlier models is their primary focus on language. As the represented entities are not only concepts but also words, the relations accessed through vector space metrics can be considered as quantitative lexical relationships. I expand on this point in 1.4.3

Most existing distributional models fall into two classes: representational distributional models and task-based distributional models. Representational models form representation directly from the text input, through unsupervised 'word count'. On the other hand, task-based models usually have a connectionist architecture which is trained on a corpus to solve certain tasks. Useful semantic representations can be derived from components (e.g. hidden layer or input layer) of the connectionist architecture after training. Although differed by how representation is formed, the two classes of distributional models have common advantages and limitations. I will review the two model classes and discuss their commonalities.

Clarification for the term 'distributional model' is needed before moving on. In general, 'distributional' refers to the sense that the representation structure is built on how words co-occur in the corpus. It does not inform on the representational form. Nevertheless, almost all existing distributional models represent concepts as vectors in a multi-dimensional Euclidean space. Therefore, it is widely assumed that distributional models are vector space models. However, this dissertation is a proposal of representing distributional semantics in graphical forms. As a result, in this dissertation the term 'distributional model' refer to all semantic models that encodes distributional data, not restricting to vector space representation. The only exception is this section. However, as section 1.4 reviews existing distributional models (dominated by spatial representations), the term 'distributional model' hereafter in this section only refers to spatial distributional models.

1.4.1 Representational Distributional Models

Representational Distributional Models directly form vector representations for words, with different ways of encoding information. In Hyperspace Analogue to Language (HAL) model (Lund & Burgess, 1996), a co-occurrence matrix is derived in which the rows and columns are both word types in the corpus. The row vectors are considered as the representations of the words, with the columns the context word dimensions. Each cell in the matrix is the co-occurrence frequency of the represented word and context word. For example, the mini-corpus *The dog chased the cat* have four word types *the, dog, chased, cat*, resulting in a 4 by 4 co-occurrence matrix, with the four dimensions corresponding to the four words in order. Suppose only adjacent forward and backward co-occurrences are counted, then *dog* has one co-occurrence with *the* and chased, but no co-occurrence with *cat*, therefore the word vector for *dog* is $(1, 0, 1, 0)$ in this corpus.

Alternatively, a word can be represented by the document it occurs in, e.g. in the Latent Semantic Analysis (LSA) model (Landauer & Dumais, 1997). A 'document' refers to a certain part of the corpus, which can be a section, a page, or a paragraph, etc. All word types in the corpus can be represented by the vector showing the frequency of the word in each document. Suppose the word *cake* has only occurred twice on the second page and once on the third page of a 5-page corpus, then it has the representation $(0, 2, 1, 0, 0)$, with each page as a document. In practice, the dimensionality of the vectors can be huge, given the scale of the corpus. As a result, the singular value decomposition (SVD) process is usually applied to reduce the dimensionality of the vectors. The reduced vectors no longer represent the exact occurrences of words in documents, but still reflect the abstract distributional pattern: words with similar reduced vectors have similar occurrence in documents and set of surrounding words (Rubin, Kievit-Kylar, Willits, & Jones, 2014).

Representational models either encodes word-word (WW) co-occurrence(Jones & Mewhort, 2007), e.g. *bound of encoding aggregate language environment* model (BEAGLE), or word-document (WD) co-occurrence. Differed by the exact encoding, all models form a vector space and evaluates lexical/semantic relations by vector space metrics. The relationships represented in distributional models are mostly quantitative, obtained through taking vector metric measure on the word pairs, e.g. cosine-similarity, Euclidean distance (L^2, city block) or correlation. These quantitative relationship have succeeded in various semantic tasks, such as judging synonym (Landauer & Dumais, 1997) and forming semantic categories (Lund & Burgess, 1996).

The representational models have some shared encoding parameters which may affect the nature of representation. As we have seen, given a corpus and the co-occurrence type, the vector representation is a function of the range of co-occurrence. For WW models, it is the direction and size of the co-occurrence window: e.g. whether counting forwardly, backwardly or in both directions, and how large the window is. For WD models, it is the range of document: a page, a paragraph, or a sentence. In addition, to avoid the effect of word frequency, the co-occurrence counts are usually normalized, and the normalization formula may as well affect performances in downstream tasks. Different encoding parameters give rise to different vector representations, which in turn influence the quantitative semantic relations derived from the vectors. (Bullinaria & Levy, 2007, 2012).

1.4.2 Task-Based Distributional Models

On the other hand, tasked-based models are trained on linguistic corpus to solve certain task. In most of these models, a connectionist network consisting of multiple fully connected layers is inputted with linguistic data as sequences of words. Each word is represented as either localist or distributive activation on the input units (input layer). The activation is transmitted to the output layer through intermediate layered connections, and the distributive activation on the output layer is taken as the response of the network. Learning happens when the output response mismatch with the expected response: An error is computed and propagated 'backwardly' to the lower layers of the network (backward propagation), and the connection weights are tuned in the direction to reducing the error. With more and more training, the network architecture gradually captures the statistical pattern of the input, reflected on the converging connection weights. After training, semantic representation of lexical items or sentence 'gestalts' can be derived from the distributed activation on various part of the network (e.g. output layer activation, or upward weights from the input layer), given the targeted word or sentence input.

Earlier neural network models have been implemented to solve various semantic computation tasks. For example, in the parallel distributed processing (PDP) approach on sentence comprehension (McClelland et al., 1989), a neural network was trained on an artificial corpus in which sentences (as word sequence) were paired with their thematic role assignments. The goal was to solve the thematic role assignment task: Given a sentence, the model needed to specify the thematic role of all its constituents. Modern task-based model instead are trained to uncover the 'blocked' words in a given sequence.

Another type of neural network models are trained to solve a relatively more straightforward task:

predicting the next symbol in a sequence. This approach at least traces back to simple recurrent network (SRN) model, which can integrate previous and current information of the input sequence in the hidden layer of the network (Elman, 1990, 1991, 1993). In this way, SRN can form a holistic representation of the sequence, and use the representation to predict the next coming word. More recent language models like Word2Vec (Mikolov et al., 2013) have been trained on similar type of tasks. In Word2Vec, each input word sequence is marked with a target word, and either the target or the 4 surrounding words are blocked. The task of the model is to predict the target or the surrounding context, which is considered as an extension of the widely used 'predicting next word' task.

Despite the different problems to solve, the task-based models are able to form representations featuring semantic structures embedded in the input. In the PDP model (McClelland et al., 1989), weights on the connections between the input units and the hidden layer units were taken as the lexical representations of the words. A cluster analysis of the lexical representations has demonstrated a clear semantic structure: the 'eating' verbs were closest to each other, followed by the 'drinking' verbs, and then related actions (e.g. *spread* and *stir*, which collocated with *bread* and *coffee*). The similar analysis exploring semantic structure formed in the task based networks have been conducted in prediction based models trained on artificial corpus (Elman, 1990, 1991) and more recent models which were trained on naturalistic corpus. Huebner and Willits (2018) have trained multiple connectionists network to predict missed words in the sequences, from a child directed speech corpus, and have also used input layer upward going weights as lexical representations. The resultant representations have shown a delicate and complex semantic structure with fine-grained categorical and taxonomic knowledge. Trained on a larger corpus, the Word2Vec model (Mikolov et al., 2013) is able to form analogues based on the lexical representations. Difference between word vectors was taken as their quantitative relation. It turned out some word pairs that formed analogues had comparable vector differences, such as country-capital pairs, e.g. *Germany:Berlin* and *Russia:Moscow*. These evidences show that while the task-based models are primarily trained to solve problem, they may form effective representations reflecting semantic structure in the input.

Before talking about the advantages and challenges of modern distributional model, I need to note a family of tasked-based models: Transformers, which has given rise to the most powerful language AIs to date. These models are distributional models as they are trained on linguistic co-occurrence. The difference between the Transformer based models and all other task-based language models is their attention mechanism (Vaswani et al., 2017), which helps them capture the complex lexical dependencies in natural language. This computational capability have be amplified with larger and larger model and training size, resulting in powerful generative language models such as BERT (Devlin, Chang, Lee, & Toutanova, 2019), and the early GPT series (Radford et al., 2019; Brown et al., 2020). While there are voices that the state-of-the-art representative of the Transformer family, i.e., GPT-4, have at least partially opened the door of Artificial General Intelligence (Bubeck et al., 2023), the understanding of model mechanism towards the achievement is minimal, if there is any. To be more specific, it is less clear what representations in the models have led to the achievement. Despite an attempt to open the black box in Chapter 6, I will in general exclude Transformer based large language models (LLM) from the set of semantic models concerned in this dissertation, especially in the discussion on distributional models' challenges followed immediately. The challenges applies to almost all existing non-Transformer distributional models, while it is controversial whether those are still challenges to modern LLMs, due to the lack of understanding on the large models.

1.4.3 Advantages and Challenges of Distributional Models

Compared to earlier semantic models, distributional models have provided mechanisms of forming representations from naturalistic language input. Two relevant benefits follow immediately. First, in distributional models, all linguistically expressed concepts can be represented, e.g. verbs, adjective, phrases. As a result, a broader range of lexical relationship can be accessed and evaluated. Second, distributional models generically depict a learning process on the semantic structure, such that the modeling works may in turn implicate on theory of language acquisition (Elman, 1993; Huebner & Willits, 2021b, 2021a).

Despite the considerable improvements compared to early works, current distributional models still face challenges such as: i) representing indirect semantic relations; ii) representing and generalizing complex lexical dependencies. As I argue in detail, these issues are relevant as they are among the core goals of semantic representation and matter for language comprehension.

Indirect Relation

First of all, current distributional models have trouble forming indirect semantic relations, while there have been evidences showing such capability in human (Balota & Lorch, 1986). The capability of forming indirect relation is relevant in language processing due to human's productivity in generating meaningful expressions (Fodor & Pylyshyn, 1988). People not only judge whether an expression is grammatical, but also make sense of the expression, e.g. *cut cake* is more plausible than *cut cookie*, and *cut lamp* makes least sense. However, due to combinatorial explosion, the expressions we have seen only take up a small portion of the grammatical expressions that we can form. (e.g. *cut cake* have probably been heard, while *cut cookie* and *cut lamp* are rare in language, yet are judged as grammatical). Then, one goal of semantic representation, is to help preferential judgements on the unobserved yet grammatical expressions, based on the limited input. Most models are able to learn the *cut-cake* co-occurrence provided in the corpus. The problem for the semantic model is to capture the graded preference for the phrases have not occurred in the corpus (provided that all words have occurred). As I will explain in detail in Chapter 3, such inference is based on the successful representation of both syntagmatic and paradigmatic relationships. I will show that canonical vector space representations struggle in representing indirect relations 3, and provide a formal theoretical explanation in 4.

Complex Dependencies

The second challenge faced by current distributional model is representing multi-way lexical relationships. For example, the model needs to tell that *cut cake* is different but similar to *cut pie*, different but related to *bake cake*. In addition, it is supposed to relate *cut cake* to instrument nouns like *knife*, but not *ax* or *scissor*, which all associate to the verb *cut*. As I will investigate more in Chapter 6, representational and task based distributional models have respective difficulties on this topic. Most representational distributional models primarily focus on word representations, and it is not easy to effectively form representation of more complex expressions, like phrases, from the word vectors (Mitchell & Lapata, 2010), especially when the resultant phrase vectors are tested on syntagmatic relations, e.g. between *cut cake* - *knife*. Alternatively, task based model can form holistic representation of sequences as an implicit composition of words (McClelland et al., 1989; Elman, 1990). While the recurrent networks can successfully learn observed multi-way leixcal relations, such as *cut cake* - *knife*, I will show that they have trouble generalizing the learned relation to novel lexical combinations (Chapter 6).

I argue that these challenges on existing distributional models is largely due to the vector space representational form. As I will show in later chapters, the spatial representation and the canonical vector space metrics may limit the model in forming indirect relations and addressing multi-way relations. Notice that the issues are not caused by the distributional encoding of linguistic data. Regarding this, I propose a novel approach that use distributional linguistic data to build a knowledge structure in graphical form – Distributional Graph – instead of the canonical vector space. To measure the graded lexical relationship, I adopt spreading-activation algorithm on the graphical structure, inspired by classic and recent works (Anderson, 1983; Collins & Loftus, 1975; Deyne, Navarro, Perfors, & Storms, 2016; Rotaru, Vigliocco, & Frank, 2018). Before developing a theory Distributional Graph, I will first review its non-distributional graphical predecessors: focusing on the advantages of the graphical representational structure, the spreading-activation process, and disadvantage of these models for being non-distributional.

1.5 Graphical Model and Spreading Activation

Graphical model for semantic representation (i.e. semantic network) can be traced back to Quillians and Collins (1969), in which concepts (categories) and predicative features are represented by nodes, and semantic relations by links between the concept nodes. The featured characteristic of graphical representational structure is the explicit encoding of relation: the relation between concepts has to be defined in order to form the structure (the link between concepts). The relations defined on the links are mostly qualitative, such as categorical subordinates (Collins & Quillian, 1969), conceptual semantic similarity (Collins & Loftus, 1975), proposition-argument (Anderson, 1983), formal lexical relation such as synonym and hypernym (Miller, 1992), and empirical association norm (Nelson, McEvoy, & Schreiber, 2004). Moreover, network metrics can be used to quantify relations on graphs. For concepts directly connected by a link, their relation can be encoded quantitatively as weights on the links, such as associative strength (Deyne et al., 2016), and distributional similarity (Rotaru et al., 2018). For connected concept pair with no direct link, their quantitative relation can be measured by graphical distance (Kumar, Balota, & Steyvers, 2019) and other measures defined on network structure (Deyne et al., 2016; Rotaru et al., 2018). The potential benefits by the graphical representational structure comes from the indirect relation made and dictated by the discrete topology, accessed through the spreading-activation process. I will return to the structural benefits after discussing the spreading-activation process.

1.5.1 Spreading Activation on Semantic Graph

To date, the idea on spreading activation goes back to a series of works by Allan Collins and Ross Quillian, which have been summarized and extended in Collins and Loftus (1975). The extended theory has conceptualized a semantic network in which links between nodes as shared property between the concept nouns. Within the semantic network, activation starts from a concept node that is being processed (e.g. during reading) and spreads to the neighboring nodes via the attached links, tagging the neighbors by the source name, and the neighboring nodes repeat the process on and on. Although it has been assumed that only one concept can be processed at a time, activation may remain on the nodes while decaying over time. As a result, when language is being processed, a node can receive and sum up the activation from multiple sources, and concatenating the paths from different sources to the intersected node would result a path between the sources. Once the accumulated activation pass certain threshold, the path leading to the intersecting activation will be evaluated. While it remains less clear how the evaluation is carried out, the work has proposed a basic framework on how information on a network could be accessed through the spreading-activation regime. However, this early attempt lacked details leading to computational implementations.

Anderson (1983) has added on top of the conceptual proposal with a mathematical formulation. The network consists of propositions as 'unit' node and the arguments and predicates of the propositions as 'element' node. Each unit node only has links to its elements and so does the element nodes. A cognitive unit consists of one unit node and element nodes (the predicates and arguments) attached to it. Different cognitive units may share elements, so that the cognitive units are connected into a network. A cognitive unit is encoded in the long term memory when a trace of the proposition (event) is established, with its unit and element nodes assigned with strength 1. After the initial encoding, the strengths of units and elements accumulate every time the nodes are processed. Therefore, the strengths of unit nodes reflect the frequency of the proposition or event being processed, and the strengths of element nodes reflect the frequency of corresponding concepts.

The computational formulation of spreading activation on the semantic network is based on these frequency contingent node strengths. The activation initiated on the source node is proportional to its strength (frequency), thus, more frequently processed concepts receive more initial activation. The activation spreads to neighbors proportional to the relative 'significance' of the shooting node to the receiving node. To be more precise, suppose a_i is the current activation on node i, then it spreads to its neighboring node j, with $a_i \cdot \frac{s_i}{\sum_{k \in N_j} s_j}$, where s_i is the strength of node i, and N_j is the set of neighboring node of node j. As a result, more activation would spread from the shooter to the receiver if the shooter is more relevant to the receiver. When an event (represented in propositional form, with its elements included in the long term memory, e.g. the semantic network) is being processed, its elements become the source of activation, which shoot to the neighboring nodes. As it allows activation reverberates (activation spreads from node 1 to node 2 and flows back to node 1), activation level on the finite network will finally converge to an asymptotic state. This converging level is taken as the determining factor for processing and retrieval time for the proposition if it has been encoded in long term memory. In this way, the computational model shows a clear interaction between processing and the semantic structure.

While in earlier works, spreading-activation is modeled as a cognitive process to access concepts in long term memory, more recent models have used it metaphorically as a measure to compute quantitative lexical relationships. In contrast to Anderson (1983), the recent models have reflected the strengths on links. In Deyne et al. (2016), the link strengths are empirical word association strengths. In the norming study, participants are asked to report the first words coming to mind given a target word. A link is formed between two words, if one is frequently responded as the associated word to the other, and the weight on the link reflect the response frequency. Alternatively, Rotaru et al. (2018) used vector representations of words in distributional model to form a similarity graph by linking two words if their vector similarity is above zero (the links bear lexical similarities). Regardless of how the link weight is derived, these models have primarily focused on the word-word lexical relations, and used spreading activation as a tool to access them quantitatively. To be more specific, the semantic relatedness from a source word to a target word is calculated as the activation that initializes on the source and spreads to the target. During the process of spreading, a node spreads its activation to its neighbors proportional to the weight of the link attaching to the neighbor, in other words, the shooting nodes send more activation to the neighbor that it prefers, rather than the neighbor that prefers it. Notice that the source node and the target node can be indirectly connected, and their relatedness is guaranteed by spreading activation.

1.5.2 Benefit of Graphical Structure

The access to indirect relatedness on graphical structure through spreading activation is beneficial. For example, in Rotaru et al. (2018), the strengths on the similarity graph are the semantic similarity between vector representation of the words, and the adjacency weight matrix is the word-word similarity matrix that reflects the spatial similarities in distributional models. Equivalently, the similarity matrix can be considered as 1-step activation spreading on the similarity graph, which measure the similarity between concepts only by the direct link between them. If two concepts have a link (their similarity is greater than zero), then the similarity would be the weight on the link, otherwise, it is zero. However, in the similarity graph, two concept nodes may be connected by other indirect paths of arbitrary lengths. The general spreading activation process provides access of similarity through those indirect paths. Rotaru et al. (2018) has found that after adding the similarities accessed through the indirect paths, the model's lexical similarity better accounts for human semantic similarity judgements. The finding is shared in related works (De Deyne, Navarro, Perfors, & Storms, 2016; Kumar et al., 2019).

Although the benefit is realized by the spreading-activation, it arises from the graphical structure. In spatial representation, the semantic relation between two concepts is always directly accessed by the vector space metric, while in a network, two nodes may have no direct link, but connected through an indirect path. The network topology first **allows** the indirect relations between concepts to be evaluated. Moreover, it **dictates** the evaluation to go through the intermediate nodes and links on the path. As a result, it makes the indirect relation as a collection of other relations. Such a process results from the discrete topology, and is a feature unique to graph. It can be argued that in a vector space, two concepts can be indirectly evaluated by going through 'intermediate' concepts. However, the approach is problematic, as (i) it is a non-trivial task to decide which intermediate concepts count, and (ii) the decision of intermediate concepts and relating the two concepts to them is essentially a process of constructing a graph, and therefore the approach by itself is already graphical. As I will formulate in Chapter 4, the traverse through indirect paths on graph is analogical to some higher-order similarity in vector space, which is usually not accessed through canonical spatial metrics.

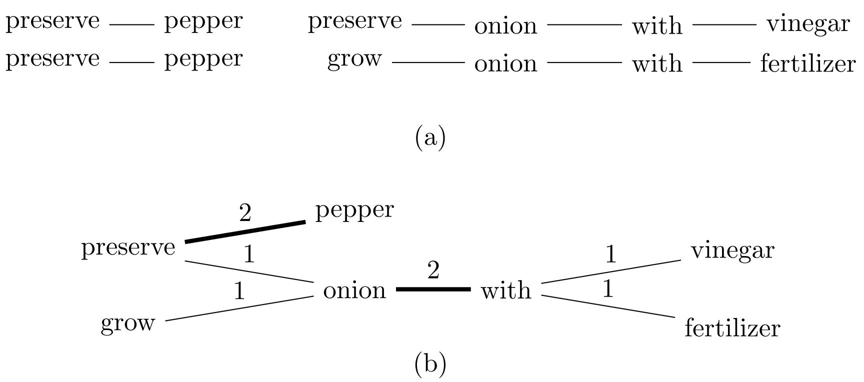
1.5.3 Issue for non-Distributional Graphs

Nevertheless, the existing spreading activation and graphical models (except for the recent similarity graph) may not be ideal since they are not distributional. In more recent studies, graphical structures are constructed from scaled-up normative data (Deyne, Navarro, Perfors, Brysbaert, & Storms, 2019; Miller, 1992; Nelson et al., 2004) and they have shown the norm-based models outperforms distributional models in some semantic tasks (Deyne, Navarro, Collell, & Perfors, 2021; Kumar, Steyvers, & Balota, 2021). However, these endeavors have provided no mechanism about how these semantic structures arise from naturalistic input. Moreover, similar to the earlier models, existing graphical approaches have been constrained on the scope of concepts it can represent and the range of language processing data it may speak to. If the graphs can encode distributional information from large-scale linguistic corpora, they will have the potential to represent more complex lexical relationships, and will be more readily to compare with the spatial distributional models.

Chapter 2

Distributional Graphs

Distributional Graph inherits the basic conception of semantic network (Collins & Loftus, 1975), which represents concept or word by node, and relationship between concepts by edges linking the nodes. Particularly, two concept nodes in a distributional graph are linked by an edge if they are related distributionally. The distributional relation may refer to a wide range of co-occurrence relations: co-occurrence of words in a corpus, co-occurrence of objects in a physical world, or co-occurrence of a word and its physical-world referent. A clarification is that the relations encoded in the edges of a distributional graph may not be abstracted. In other words, they are raw co-occurrence, and the abstraction happens on the graph after the structure is formed (by network metrics). For an example of encoding abstracted distributional relation, Rotaru et al. (2018) have encoded similarity between word vectors in the edges between word nodes. In that case, the information encoded on the edges are abstracted distributional relations between words, and therefore is considered a graphical model encoding distributional data, but not as a distributional graph proposed in this dissertation. While noticing 'co-occurrence' can be defined broadly, I will focus on co-occurrence of linguistic expressions in a corpus. To sum up, I refer to distributional graph as a graphical structure encoding raw co-occurrence between words in linguistic corpus.



(a)

(b)

Figure 2.1: Formation of a distributional graph from mini corpus *preserve pepper, preserve pepper, preserve onion with vinegar, grow onion with fertilizer.* (a) word chains formed by encoding adjacency co-occurrence, (b) network formed by joining word chains.

Usually, corpora for Distributional Graph are collections of linguistic expressions (e.g. phrase and sentence), rather than a sequence of words. For instance, the mini corpus '*preserve pepper, preserve pepper, preserve* *onion with vinegar, grow onion with fertilizer*' is treated as a collection of phrases allowing repetition. In general, the expressions are punctuation free, but there might be variations. Alternatively, the corpus can be considered as a sequence of symbols, including words and punctuation. The different treatments on the corpus may lead to different representational structures (See chapter 3). Without further notice, the linguistic corpus refer to the collection of expressions version.

A distributional graph is formed with two steps: (i): converting linguistic expressions into graphical form; and (ii) joining the graphical forms. In the first step, a linguistic expression, e.g. word, is transformed to a node, and nodes are linked following certain encoding method. The simplest way is to form a word chain, in which two words are linked by an undirected edge only if they are adjacent in the expression (Figure 2.1a). After the first step, all expressions in the corpus are presented in a graphical form. In the second step, the graphical expressions are joined by the shared nodes and links. For example, in Figure 2.1a the two four-word phrases have two words in common: *onion* and *with*, alongside with the link between the words, and the two two-word phrases are identical. The repetitive structures are combined, resulted in a unique word nodes for each word type, and accumulated weights (marked by the numbers above the edges) reflect the number of repetitive edges in the graphical expression collection (Figure 2.1b). For example there are two *preserve-pepper* phrases and only one *preserve-onion* in the corpus, as a result, the weight for the former relation is 2, and the latter 1 in the derived distributional graph. Combining the two steps, I refer to the process of constructing distributional graphs as **Connecting Language**.

What is the structures formed by connecting language? First, it is a semantic network, a representation of concepts/knowledge. More importantly, it is a more 'linguistic' representation of knowledge: (1) The nodes are not just concepts, but also linguistic units, e.g. word/phrase; (2) The encoded relations between the concepts are linguistic co-occurrence, which may even incorporate syntactic/semantic structures relevant to language processing. These 'linguistic' features distinguish Distributional Graph from traditional semantic networks that represent static concepts. As the model encodes 'linguistic forms' of the concepts, it is possible to access and evaluate concepts dynamically generated in diverse linguistic forms, and the relationships between the generated concepts. On the other way round, since the model encodes the concepts through linguistic structure in graph topology, effective operations on the graph may help capture the complex lexical relationship in language use, and account for phenomena in complex language processing. The above mentioned access/evaluation is actually a measure of semantic relatedness in the graphical structure, which will be implemented with a spreading-activation based algorithm.

The chapter is organized as following. First, in Section 2.1, I define and introduce the spreading-activation based measure of semantic relatedness on networks in general. I develop and discuss two different types of distributional Graphs: Co-occurrence network and Constituent Tree Network in Section 2.2 and 2.3. In the last section, I briefly summarize the theoretical motivation and prospective advantages of distributional graph. This part is left at the end, as it would be easier to explain the ideas after the modeling details have been presented.

Before moving on, it is necessary to clarify the notations and terms that will occur. Words will be denoted by lowercase English letters, and composed expressions like phrase and sentence will be denoted by uppercase letters. All linguistic expressions are lemmatized, and only nouns, verbs, adjectives, adverbs and prepositions are encoded. The graphs are distributional graphs by default and therefore the edges are always weighted, no matter whether the weights are explicitly marked. The notation for a linguistic expression simultaneously denote its corresponding graphical object. For example A will denote both the phrase *grow pepper* and its corresponding node in the graph. $\mathrm{SR}(\cdot,\cdot)$ will denote the semantic relatedness between two objects (two linguistic expressions/concepts, or the corresponding nodes), and it is sensitive to the argument order. SR(A, B) denotes the semantic relatedness from node A to node B, and is by definition different from SR(B, A). The difference and its psychological implications will be explained in next section.

2.1 Spreading-Activation based Semantic Relatedness on Graphs

Spreading-Activation was first proposed as a conceptual process of accessing and retrieving information from semantic networks (Collins & Loftus, 1975). More recently, the conception has been applied metaphorically to define and measure relatedness (similarity) on semantic graphs. There are two reasons why spreading-activation can be adopted for a relatedness measure on network structures. First, the spreading-activation process may psychologically ground the relatedness measure. Intuitively, two concepts or words are considered related if one can be accessed/primed given the access to the other. This intuition was reflected in the spreading-activation process in Collins and Loftus (1975): An attended concept node sends out activation while its related concepts receive activation and are accessed. A quantitative refinement follows: Relatedness between two concepts can be measured by the likelihood of accessing to one concept provided access to the other. It has been conceptualized that the likelihood of access is determined by the amount of activation received by the concept node (Collins & Loftus, 1975). Therefore, activation received by a node from a sending source can be generically taken as the relatedness between the two concepts.

The other reason is that spreading activation can capture more information embedded in the network, compared to simpler measures such as graphical distance. Here, I first list three graphical factors that reflect (distributional) semantic information in corpus on the graph: the graphical distance, strength on edges, and number of paths between nodes. I will explain how the three factors affect the semantic relatedness on graphs, and then show how the spreading-activation measure takes care of the three factors.

2.1.1 Three factors to Relatedness on Graph

The most generic way to measure the relatedness between two nodes on a graph is by their graphical distance, defined as the length of the shortest path(s) between the nodes. Intuitively, smaller graphical distance imply greater semantic relatedness: in Figure 2.2 (a), node a is more related to b compared to c. What does smaller distance in a distributional graph reflect? If two words co-occur in a sentence, it is likely that they are close in the distributional graph: They are encoded within the same graphlet. If two words do not typically co-occur, but they both co-occur with some other words, then the two words might be a little far from each other: They are encoded in different graphlets and joined through the shared words. Further, if two words do not even share any co-occurrence, it is likely that they are connected with more graphlets, resulting in larger graphical distance on graph. In this way, the graphical distance roughly capture the 'directness' of word relations. Relation strength (weights on edges) is the second factor when considering relatedness, which usually represent the co-occurrence frequency in distributional graphs. For example, in Figure 2.2 (b), e has co-occurred 9 times with d but only once with f. The intuition follows that e and d should have stronger relatedness. In distributional graph, the relation strength basically reflect the (normalized) co-occurrence frequency. If two words co-occurs a lot, it is reasonable to grant them a strong relationship, reflected by a large edge weight in the graph. Finally, in graphical structures, it is possible that multiple paths exists between nodes (e.g. in Figure 2.2c, g is connected to h through 2 paths, while i and h only by one path with equal length to the $g - h$ paths. In this case, h should be more related to g if the strengths of the edges on the paths are identical. In a distributional graph, the number of paths can be considered as 'shared (distributional) features' or 'shared contexts' between the words/concepts. If two words share lots of features, their relation should be strong reflected by multiple paths between the word nodes. Ideally, a network measure should simultaneously capture the three factors that reflect distributional information in corpus.

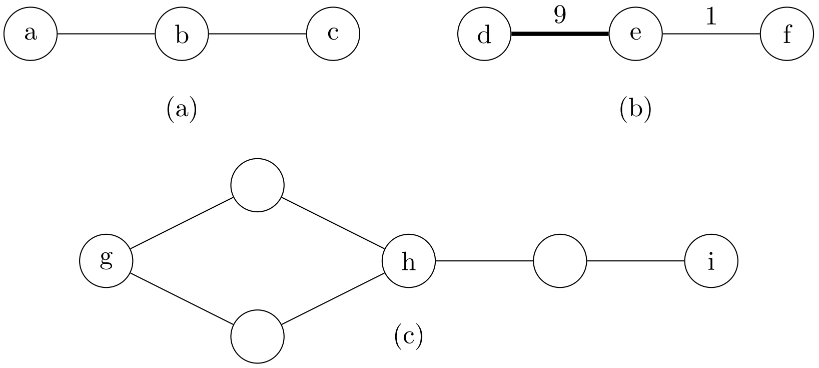


Figure 2.2: Three factors affect relatedness on graphs: (a) graphical distance, b is closer to a compared to c; (b) edge weight, $d - e$ is stronger than $e - f$; (c) number of paths, h has more paths to g compared to i.

However, in most cases, the three factors: graphical distance, edge weight, and number of paths interact with each other. For example, consider SR(k, j) and SR(k, l) (Figure 2.3), the three factors have counteractive effects on the two relations: k has graphical distance 1 to l but 3 to j, which makes it closer to l compared to j. However, there are two length-3 paths between k and j which strengthens the $k - j$ relation. Moreover, although the paths are long between k and j, the edges on the paths are much stronger (reflected by the frequency weight) than the direct edge between k and l, which further complicates the picture. Since this kind of counteraction is common in distributional graphs derived from naturalistic linguistic corpus, a measure that integrates the three aspects is necessary for defining the semantic relatedness on graphs.

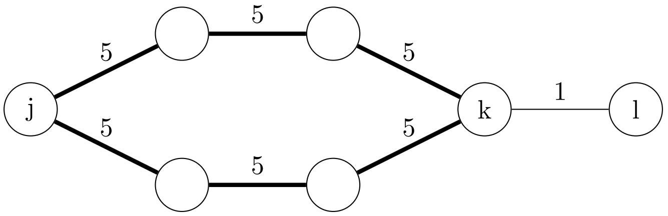


Figure 2.3: Graphical distance, edge weight and number of paths counteract with each other concerning pairwise relatedness on graph.

2.1.2 Spreading-Activation Measure

In classic works (Anderson, 1983; Collins & Loftus, 1975), and recent modeling studies (Deyne et al., 2016; Rotaru et al., 2018), the spreading-activation based relatedness is conceptualized as the activation spread from a source node to a target node. In the activation process, nodes with non zero activation spreads all its activation to its neighboring nodes at each discrete time tick. In this way, once a node is activated, all nodes connected to this activated source will receive some activation after a while, and the amount of activation received by the target node from the source node is taken as the semantic relatedness from the source to the target. I adopt the similar conception, and the formal definition will be unfolded with three steps: (i) Activating neighbors, (ii) Relatedness through Spreading-Activation on a single path, and (iii) Formal definition of semantic relatedness on a distributional graph.

At each moment, nodes with activation spreads its activation to the neighboring nodes proportional to the weight on the linking edges. For example, in Figure 2.4a, node A has edges to C,D,E with weight 3, and edge to B with weight 1. As a result, A would spread its activation to the neighbors proportional to these weights, ending up in the normalized weighted outward edges in Figure 2.4b, so that less co-occurred items become less related. The normalization can be repeated for all nodes in the graph, resulting in the normalized (directed) graph (Figure 2.4c), where weights on the directed edges denote the proportion of activation flown from a node to its neighbor linked by the edge. B, D, E, F has outward weight of 1, since each node only has one neighbor in the graph. A verbal example of normalized graph is provided in Figure 2.4d. The proportional spreading leads to asymmetric relatedness between node pairs, which echoes empirical data (Tversky, 1977). In Figure 2.4d: lawnmower is much more related to grass than grass relating to lawnmower.

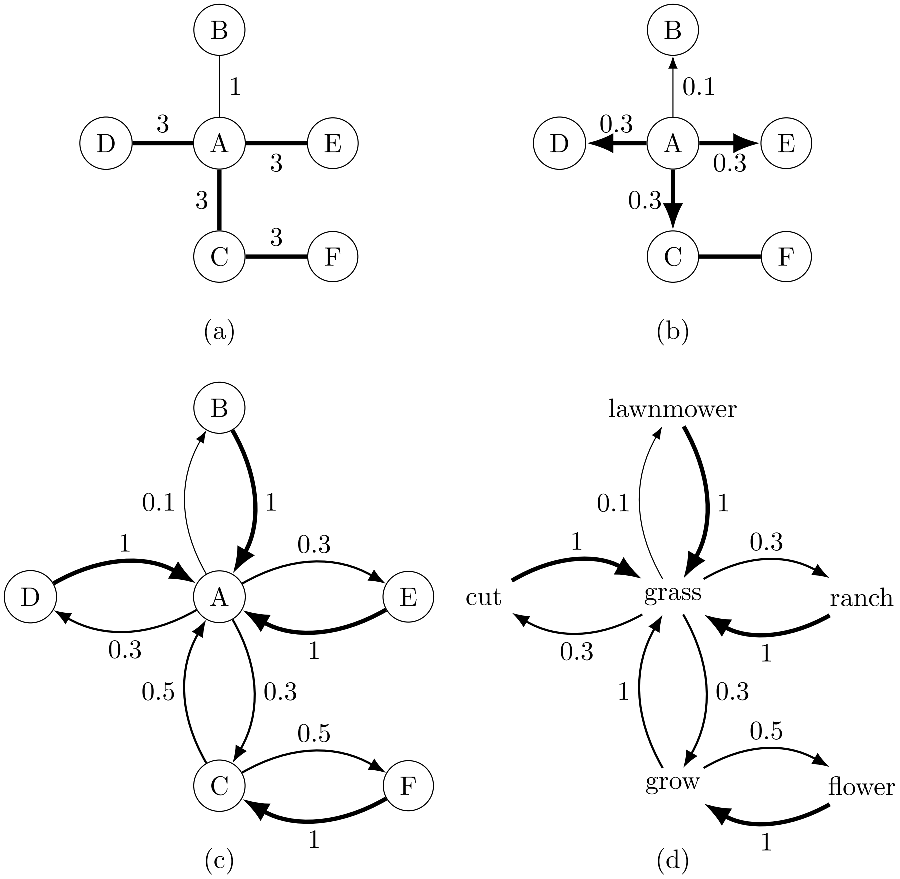


Figure 2.4: Activation spreading to neighboring node: (a) Node A has edges linking to neighboring nodes with different frequency weights. (b) Normalized outward weights denote proportion of activation for node A. (c) Normalized outward weights denotes proportion of activation for all nodes in the distributional graph, featuring asymmetric relatedness. (d) Verbal example of the normalized graph: *Lawnmower* has weaker relatedness from *grass* compared to other words, while *grass* is less related to its neighbors than its neighbors relating to itself.

How a node spreads activation to its neighbors provides a way to define the relatedness between adjacent concept nodes in a distributional graph. Nevertheless, an important feature of graphical structure is the access to indirectly connected nodes. On a graphical structure, more than one paths may exist between two nodes (Figure 2.5). I first define the semantic relatedness from a source node to a target node through a particular path connected the nodes, based on which the total semantic relatedness of the node pair will be defined.

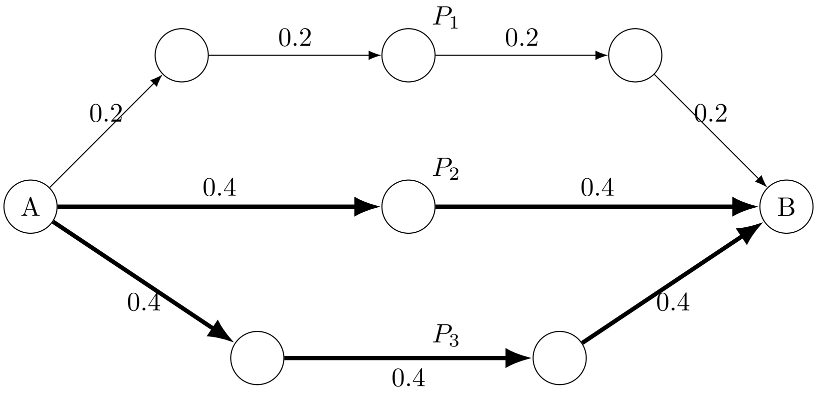


Figure 2.5: Semantic relatedness through three paths: Only normalized edges on focused paths from A to B are included, with the outward weights on the directed edges.

Suppose A and B are two nodes in graph $\mathbf{G}$ connected by a path P (from A to B), the semantic relatedness from A to B by path P, denoted by $\mathrm{SR}^P(A, B)$ is defined as follow:

$$\mathrm{SR}^P(A, B) = \prod_i \omega_i \tag{2.1}$$

where ω_i is the normalized weight of the i'th edge on the path L. This definition of path-wise relatedness takes care of both the graphical distance and the edge weights on the paths. Almost all weights are smaller than one, in other words, the activation diffuses on every edge as it travel through a path. As a result, the shorter the path, the more activation will reach the target node through the path, and the higher the edge weights on the path, the more it prevents the activation from flowing away, and the two factors altogether leads to greater relatedness from the source to the target. An illustration is provided in Figure 2.5, where three paths from A to B vary by graphical length and edge weights. The shortest path (P_2) with higher weights has the greatest relatedness ($\mathrm{SR}^{P_2}(A, B) = 0.16$), while the longest path (P_1) with lower weights is the weakest ($\mathrm{SR}^{P_1}(A, B) = 0.0016$).

Finally, the semantic relatedness from A to B, denoted by $\mathrm{SR}(A, B)$ is defined as the sum of semantic relatedness from A to B by a collection of paths:

$$\mathrm{SR}(A, B) = \sum_{P \in \mathcal{P}_{A,B}, L(P) \leq d(A,B)+n} \mathrm{SR}^P(A, B) \tag{2.2}$$

where $\mathcal{P}_{A,B}$ is the collection of paths from A to B on $\mathbf{G}$, $L(P)$ refers to the length of path P, $d(A, B)$ is the graphical distance between A, B, and n specifies the upper boundary of path length, which is selected by modelers due to specific needs. In this way, $\mathrm{SR}(A, B)$ include activation flown on all paths from A to B with length no greater than $d(A, B) + n$. For simplicity, the relatedness is denoted by $\mathrm{SR}(A, B)$ without marking the length upper bound. The contribution from longer paths are ignored as they carry relatively less activation, and are psychologically uninformative (Deyne et al., 2016). Defined in this way, the semantic relatedness has covered the third factor: number of paths connecting the node pairs. As in Figure 2.6a, assuming all edge weights are identical, then g and h are more related, since they have more paths. A verbal example is provided in Figure 2.6b, where *grow* is more related to *plant* compared to *cut*.

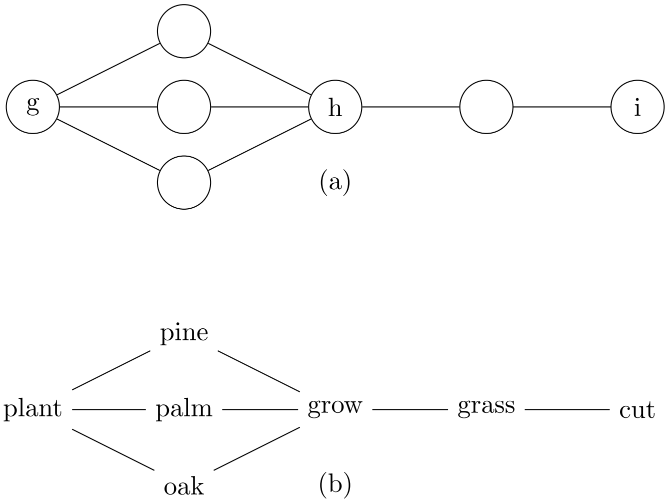


Figure 2.6: Relatedness affected by number of paths: (a) h has more paths to g compared to i, thus stronger relatedness to g, (b) verb example, *grow* is closer to *plant* than to *cut*.

2.1.3 Asymmetry of the Spreading-Activation Based Measure

It is noticed that such an spreading-activation based relatedness measure is asymmetric, in other word, $\mathrm{SR}(A, B)$ is not the same as $\mathrm{SR}(B, A)$. As illustrated in Figure 2.7, $\mathrm{SR}(A, B)$ can be a lot greater than $\mathrm{SR}(B, A)$ due to the weights on the normalized directed edges. As a result, the notion of relatedness **between** A and B become ambiguous, and the direction has to be clarified. While the modeling characteristic is helpful in accounting some psychological phenomenon (Tversky, 1977), it may come across troubles when considering relatedness **between** A and B, without strong theoretical or empirical motivation for the directedness. In this case, one needs to address which direction should be chosen, and whether the choice is proper. In each specific model to introduce, the asymmetry issue will be mentioned and handled respectively.

2.1.4 Spreading-Activation and Random Walk

The activation measure introduced in this article resembles the random-walk approach (Deyne et al., 2016), in which a cognitive process is implemented by accessing lexical knowledge in graphical structures. The two approaches (random walk and spreading-activation) vary by the hypothetical activation process but largely overlap on the mathematical substances of the computational implementations.

In the random walk approach, a random-work process is defined on the network: a node is initially activated and it activates one of its neighbor (the activation randomly walk to the neighbor) at the next discrete time tick, with the probability proportional to the locally linked edge weights. In this way, at each time point, there is only one node being activated, and every node has a probability of being activated. On the other hand, the spreading-activation measure assumes the activation spread to all its neighbors in the network, and therefore is more similar to a priming process rather than an activation process. Despite the difference in the underlying process, the two approaches share commonalities: At each moment, the

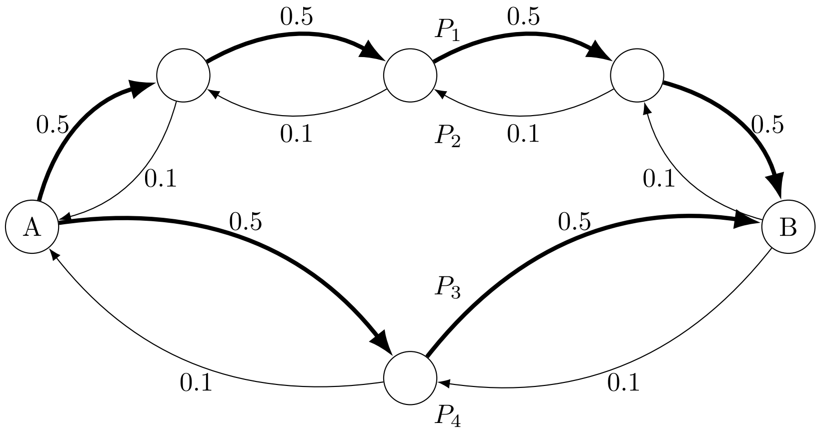


Figure 2.7: Asymmetric relatedness: relatedness from A to B accounted by P_1 and P_3 (thicker paths), and that from B to A by P_2 and P_4.

probability distribution of a random walk is equivalent to the amount of activation distributed in the network. To further present the similarity and subtle differences between the models, I give a formal description of the random walk approach.

In the random walk approach, the adjacency matrix of the network is first row-normalized, resulting in a transformation probability matrix $\mathbf{P}$. This step exactly mirrors the local normalization of each node in a distributional graph, such that each row in $\mathbf{P}$ corresponding to the outward degrees in the normalized graph (Figure 2.4) b. Then, the random-walk process is formally described by:

$$\mathbf{G_{rw}}^{(r=k)} = \sum_{0 \leq i \leq k} (\alpha \mathbf{P})^i \tag{2.3}$$

where r denote the path length and $k \geq 0$ is the length of the longest path allowed in the random walk, or equivalently, the discrete time tick in the spreading-activation process. α denote a decay rate of activation while walking to a neighboring node. The process start at $k = 0$, and every node is activated with activation 1. Then, at each time point, the activation on each node A randomly 'walks' to one of its neighbor with the transition probability on the corresponding row (for A), and the activation decays with the rate α. In this way, $\mathbf{G_{rw}}^{r=k}$ denotes the accumulation of the probability for a node being activated up to time k. Alternatively, in the activation-spreading scenario, $\mathbf{G_{rw}}^{r=k}$ denotes the accumulated activation that each node has received until time k, with the initial state that every node is activated with activation 1.

Computationally, the random walk approach is different from the spreading activation measure in three ways. First, it has the decay rate α which penalties longer paths. Second, it admits recurrent activation (activation once flown away that flow back), and lastly, it has a unified length upper bound k, so that for all nodes, only paths shorter than k are counted. In contrast, in the spreading-activation approach, the length upper bound is contingent to the geodesic distance between the node pair. For example, suppose that $d(A, B) = 2$ and $d(A, C) = 3$, then in spreading-activation approach, an upper bound modifier n indicate that for the node pair (A, B), effective paths can be no more than length $2 + n$, and no more than $3 + n$ for (A, C).

These differences result from specific modeling purposes, but they reflect similar modeling considerations. The random walk approach is implemented to access the semantic similarity on network, in which each word is represented as a vector with the entries semantic relatedness to other words. The relatedness is calculated by considering paths up to length k, so that a node pair will have relatedness 0 if their graphical distance is greater than k. Such a set-up is reasonable given the goal of measuring similarity, and the upper bound k is the way to filter extensively long paths. On the other hand, spreading activation focuses on each relatedness between every single word pairs, and the universal upper bound k will lead to trivial result (zero relatedness) for distant pairs, therefore it adopts a flexible upper bound contingent to the graphical distance between node pairs, while preventing longer paths just like the random walk approach.

Moreover, in the random walk approach, an infinite length upper bound is set up in the implementation:

$$\mathbf{G_{rw}} = \sum_{k=1}^{\infty}(\alpha\mathbf{P})^k = 1 - \alpha\mathbf{P}^{-1} \tag{2.4}$$

in which it supposes the time for random walk is long enough, so that the probability distribution will reach the equilibrium $\mathbf{G_{rw}}$. In this case, the decay rate α would be necessary to filter the meaningless longer paths. However, the current spreading-activation approach presumes a limited time range activation, and longer paths are avoided by the upper bound modifier n. In future works, the decay rate will be included for the spreading-activation measure if it turns out that the cognitive process fits better to the case that the equilibrium is reached. Similarly, in the current set up, $\mathrm{SR}(A, B)$ is defined without recurrent activation, nevertheless, counting recurrent activation might speak better to cognitive theories and empirical data. In this case, I leave the option open and may add recurrent activation to the semantic relatedness between node pairs.

Provided the differences between the two approaches (rooted from different initial modeling purposes), I emphasize that they largely converges on the mathematical substances, and the proposed cognitive processes communicate to each other.

2.2 Co-occurrence Graph

Distributional graphs encode some kind of co-occurrence relation on the node links, and Co-occurrence Graph is the most basic type. While other types of distributional graphs may explicitly encode more structured (syntactic or semantic) relations on the link, co-occurrence graphs encode only ‘raw co-occurrence’. Formation of a co-occurrence graph from linguistic input has been shown in Figure 2.1: In step 1, expressions are converted into sub-graphs, which are joined by shared word nodes in step 2. The construction of co-occurrence graph from sub-graphs, e.g. step 2, is shared by all distributional graphs, and the difference between various types of distributional graphs lies in the graphical form of the linguistic expressions. In co-occurrence graphs, expressions are converted such that words within certain distance (in text) are linked by an edge. I refer to these graphical expressions co-occurrence clusters, and the form of these clusters varies due to the co-occurrence ‘window size’ (e.g. the upper bound of distance in the expression that allows a co-occurrence count) and other encoding factors.

In terms of information encoded, Co-occurrence Graph can be thought an analogue of the HAL model which also encodes the elementary word-word co-occurrence. The difference between HAL and Co-occurrence Graph is the representational structure: while HAL aligns the co-occurrences in a row to form distributional feature vectors for words, Co-occurrence Graph encodes the co-occurring words as neighbors in a network, which can be easily accessed through spreading activation. In this section, I expand on this point and show that just like spatial distributional models, co-occurrence graph is indeed ‘distributional’: It encodes distributional pattern of words and represent it in a graphical form, from which semantic relatedness can be extracted. Furthermore, I show how the graphical representation diverge from spatial models in terms of its way of encoding indirect relation from the distributional data. This feature of co-occurrence graph is also evident in other types distributional graphs. In general, the Co-occurrence Graph model assumes very few linguistic pre-processing, as the encoding process requires not much more than raw word count. Nevertheless, there are still encoding parameters that may affect the form of co-occurrence cluster, the joined structure, and furthermore, the semantic relations represented in the network. I use some examples to show the effect of these modeling parameters and explain its indication on assumptions of the representational system.

2.2.1 Co-occurrence Graph is Distributional

While distributional pattern of words is encoded as vector dimensions in spatial distributional models, they are encoded as node neighborhoods in co-occurrence graph. Given the mini corpus *preserve pepper, preserve onion, grow onion, grow pepper, fresh pepper, fresh onion*, a co-occurrence graph encoding immediate adjacency is shown in Figure 2.8.

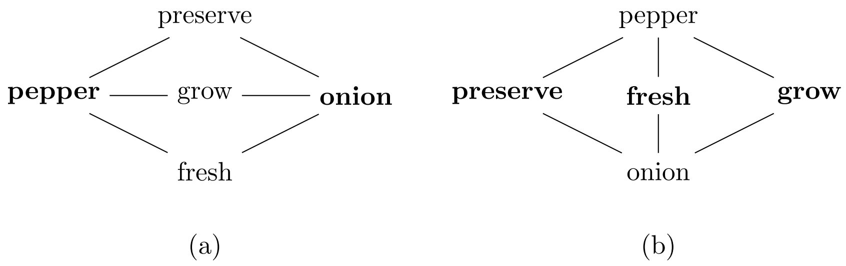


Figure 2.8: Represent co-occurring words as word features in co-occurrence graphs: (a) Treat co-occurred verbs and adjectives as features of nouns; (b) Treat co-occurred nouns as features of verbs and adjectives

In the representation, words are characterized by other words co-occurred with it, which can be considered as features of the represented words. This feature-concept relationship is mutual, as *preserve, grow, fresh* can be taken as features of the two nouns *pepper* and *onion* (Figure 2.8a), and alternatively, the nouns can be considered as features of the adjective and the verbs (Figure 2.8b). Notice that Figure 2.8a and 2.8b depict the same structure from different perspectives. In this sense, the concept-feature representations for all words are simultaneously presented in the graphical structure, which is similar to the vector encoding in spatial co-occurrence models, e.g. HAL.

Such a graphical representation can be considered as an extended implementation of the conceptual framework in Collins and Loftus (1975). In the conceptualized semantic network, nodes are concept nouns linked by shared features, which is exactly Figure 2.8a. With the co-occurrence graph approach, the 'feature encoding' is operationalized by the co-occurrence count in a graphical form. Since distributional encoding does not differentiate concepts by grammatical category, a benefit follows that not only nouns, but all open class words can be homogeneously encoded and evaluated in the same structure. To be more specific, not only nouns sharing co-occurred verbs and adjectives have stronger relations represented in the network, but also the verbs taking similar nouns and modified by overlapping adverbs. This distributional benefit resonates with spatial distributional models, yet has been absent in previous graphical approaches.

2.2.2 Indirect Relations in Co-occurrence Graph

One feature of graphical structure is the capability of encoding indirect relations through paths. On top of the graphical feature, the distributional encoding of linguistic input in co-occurrence graph leads to representation of much richer and language grounded indirect relations, compared to other semantic networks. For example, in classic associative networks, *lion* is indirectly connected to *stripe* by the intermediate concept *tiger*, while in co-occurrence graph, nouns, verbs (e.g. *cut* and *chop*), adjectives (e.g. *happy* and *lucky*), adverbs (e.g. *fast* and *swiftly*) are all indirectly related through their shared collocations, which contributes to the paradigmatic relations within different types of words.[1] This advantage also arises from the encoding of distributional linguistic data, as in its spatial counterpart (e.g. similar nouns or similar verbs have shared co-occurrence entries, which lead to higher similarity evaluated through vector space metrics).

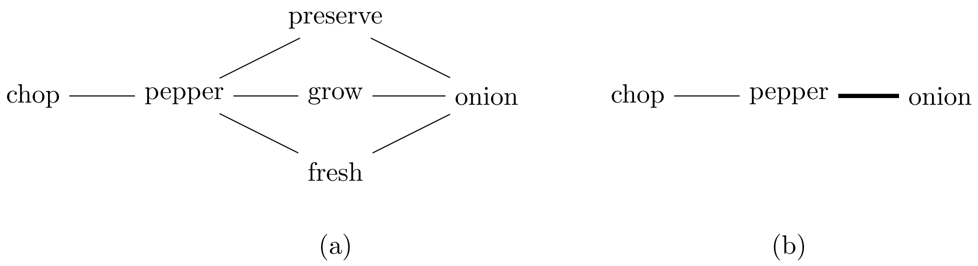


Figure 2.9: Represent indirect and direct relations in co-occurrence graphs: (a) *Chop* co-occurs with *pepper*, but not with *onion* and is indirectly connected to *onion* through three paths; (b) The three paths between *pepper* and *onion* can be 'abstracted', and taken as a strong similarity between the noun, and in turn shows the strength of the indirect relation between *chop* and *onion*.

One notable capability of co-occurrence graph is the representation of relations a little more indirect than the paradigmatic relationships (e.g. onion-pepper). For example, suppose that a verb *chop* co-occurred with *pepper* has been added to the mini corpus. The verb is immediately connected to *onion* through the indirect paths, although it has not collocated with the noun. The relation between *chop* and *onion* can be inferred based on the similarity between *pepper* and *onion* (the thickened edge in Figure 2.9b). This indirect relation is encoded in the graphical structure and can be quantitatively accessed by the spreading-activation measure. Moreover, representation of the longer indirect relation does not counteract with representations of other direct and indirect relations, such as the paradigmatic relation between *pepper* and *onion*, and the direct co-occurrence between *pepper* and *grow*. Instead, all indirect relations, no matter how indirect it is, are always built on the most primitive direct co-occurrence. As a result, relations with different levels of indirectness can be simultaneously represented and evaluated in the structure, which is not always guaranteed in other distributional models. More detailed investigations of this representational characteristics will be included in Chapter 3, with a controlled comparison between spatial and graphical distributional models.

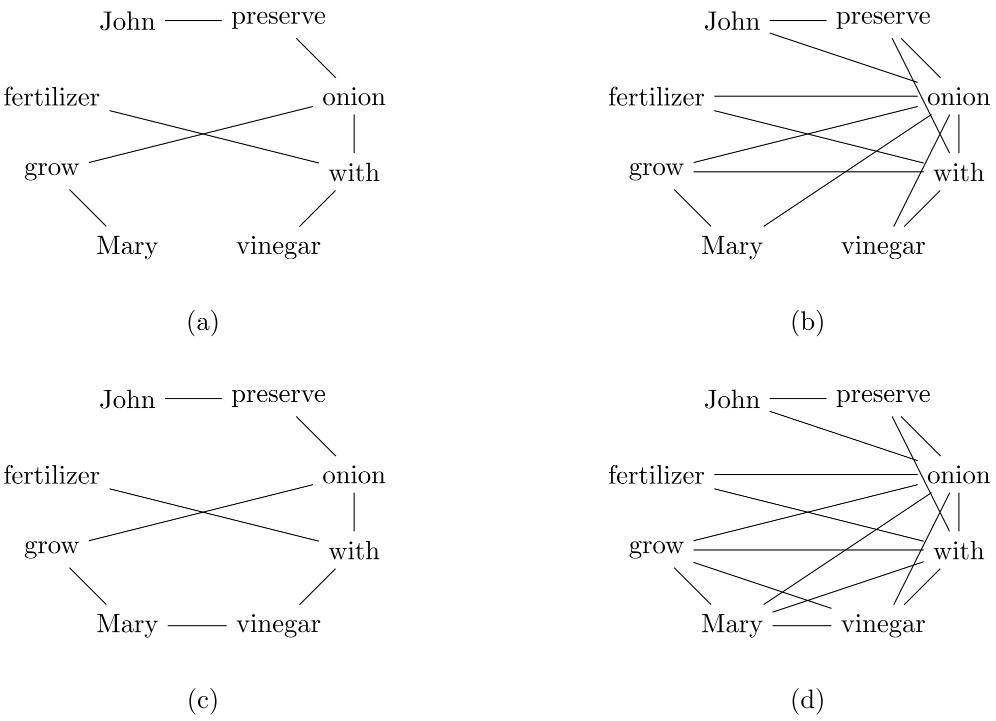
2.2.3 Effect of Encoding Factors

Since co-occurrence models mostly only take word counts, the encoding process does not require structured input. As a result, an input can be taken as sequences of words, while the boundaries for sentences and phrases are not necessary. Viewing the linguistic input as a long sequence rather than a set of expressions give rises to multiple encoding factors may affect the form of graphical structure and representational properties. These factors include expression boundary, co-occurrence window size, window weight, window type (direction), to list a few. In this section, I define these encoding parameters and explain the effects of them on co-occurrence graph with two examples: expression boundary and co-occurrence window size.

[1] I need to clarify that in co-occurrence models, paradigmatic relations are indirect by nature, as replaceable words seldom co-occur with each other, ending up with few co-occurrence encoding.

Expression Boundary is concerned with whether or not co-occurrence across expressions are counted. The expression may refer to sentence or phrase, and once the boundary is set up, the co-occurrence across expressions may not be counted. Suppose *John preserve onion with vinegar. Mary grow onion with fertilizer.* are two adjacent sentences in a corpus. If there is no sentence boundary, then the co-occurrence across two sentences can be counted, e.g. *Mary* co-occurring with *vinegar*. Otherwise, such co-occurrence may not be counted. A co-occurrence window refers to the sliding window for counting co-occurrence, and it leads to three encoding parameters. First, **window size** determines the maximal distance between two words whose co-occurrence may be counted. For size one, only adjacent words are counted. For size two, the adjacent and 'one-step farther' words may be counted, and so on. Second, **window type** concerns the direction of the window, which can be **forward**(only counting words after the represented word), **backward**(only counting words before the represented word) or **summed** (counting both sides of the represented words). Lastly, **window weight** is the parameter to differentiate nearby and distant co-occurrences. There are many types of weights, and the most commonly used are **flat** (e.g. all co-occurrence receive one count) and **linearly inverse** (e.g. the co-occurrence count is a linear inverse of the distance, such that co-occurrences with distance n in the text receive count $1/n$).

The effects of **window size** and **boundary** are illustrated with the following example. Four co-occurrence graphs based on the mini corpus *John preserve onion with vinegar. Mary grow onion with fertilizer* are shown in Figure 2.10, varied on the window sizes and choice of boundary. The graphs in the left column have window size one and the right column graphs have size two; the upper row has sentence boundary and the lower row has no boundary. All graphs have forward and flat co-occurrence windows.



(a) (b) (c) (d)

Figure 2.10: Effect of sentence boundary and window size on the form of co-occurrence graph: (a) window size 1 with sentence boundary; (b) window size 2 with sentence boundary; (c) window size 1 without sentence boundary; (d) window size 2 with sentence boundary

.

When only adjacent co-occurrence is encoded (Figure 2.10a), the graph is relatively simple. Two sentences are chained together, and only overlapped by the co-occurrence between *onion* and *with*. The structure becomes much more complicated when window size is two, as more distant co-occurrences are encoded by direct links: Both humans and the instruments are directly connected to the object noun *onion*, and both verbs are directly linked with the preposition *with*. Allowing counts across sentence boundary brings a few more connections, so that *vinegar* becomes connected to *Mary*, and even the verb *grow* (in Figure 2.10d), which are from different events.

In general, different choices on window size and expression boundary determine the amount of direct links in the network. Larger window size and absence of expression boundary create more edges, and make concepts closer to each other (higher relatedness). However, whether or not the added edges are detrimental or beneficial to the representation depends on the evaluation. In the mini corpus, *vinegar* and *grow* are brought together with the larger window size and absence of the sentence boundary. However, *grow* and *vinegar* are not related by any means in the corpus, such that they should have a relatively weak relation. In this case, window size one with sentence boundary seems to be a better way of encoding. However, in other cases, adjacent sentences could be semantically related, so that connecting words across the boundary would be beneficial.

The models though should have no judgement on the parameters or adjust the parameters flexibly. They do the counting once the parameter values are fixed. The discussion only shows that when a certain parameter is determined, it will lead to a semantic structure that is different from ones derived from other parameter set-ups. This in turn affects how semantic relations are represented in the structure and how models perform in semantic tasks. Therefore, modelers should be aware of the effects of encoding parameters, and manipulate them when building co-occurrence models, to rule out the possibility that model behavior depends on choice of encoding parameter values.

2.3 Constituent-Tree Network

The distributional graphs introduced in previous sections are built exclusively on words. While the models are capable of representing word-word lexical relations, they might fall short in handling more complex lexical relations and representing complex expressions such as phrase and sentence. As that is going to be shown in Chapter 5, co-occurrence graph may struggle in learning and generalizing on multi-way lexical relationships. I will argue the failure may be attributed to lacking explicit encoding of linguistic structures in the graphs. This motivates the development of Constituent Tree Network (CTN), the type of distributional graph that explicitly encodes linguistic structure.

2.3.1 Constituent Tree

Similar to other distributional graphs, CTNs are built from graphical linguistic expressions. While in other distributional graphs, expressions are transformed into chains, co-occurrence clusters or dependency trees, in CTN, they are converted into constituent tree.

The constituent tree of an expression is nearly isomorphic to its syntactic parse tree, yet with the following differences: (i) Grammatical categories labeling non-terminal nodes are substituted by the corresponding lexical expressions; (ii) The original lexical terminal nodes are excluded. An illustration of the syntactic tree and constituent tree of the phrase *preserved the pepper with vinegar* is in Figure 2.11a,b. Furthermore, the expressions are usually lemmatized due to the interest in open class words. As in the example, *preserved the*

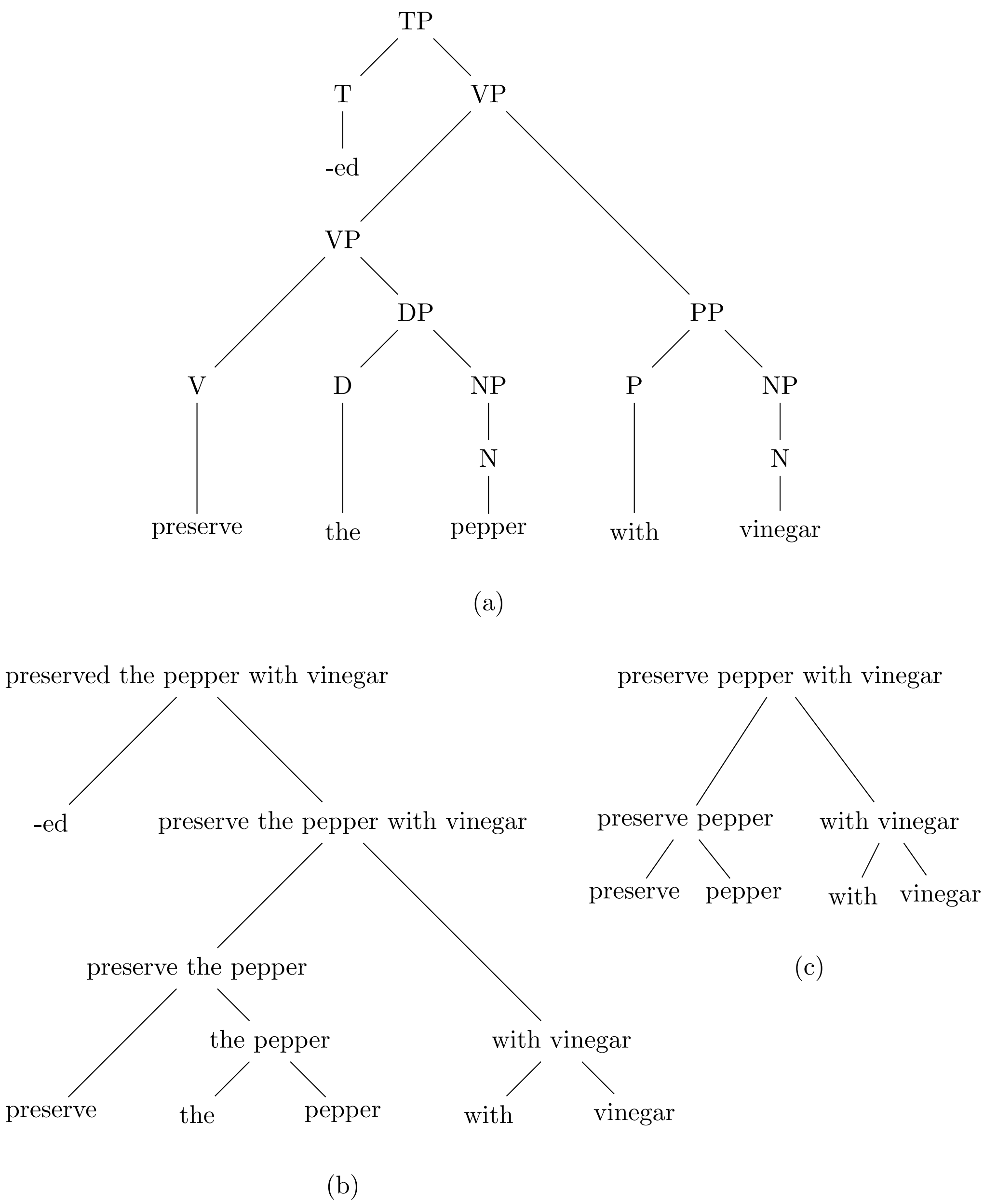


Figure 2.11: Structures for the expression *preserved the pepper with vinegar*, or the lemmatized form *preserve pepper with vinegar*.(a) Syntactic tree. (b) Constituent Tree. (c) Simplified Constituent Tree.

pepper with vinegar becomes *preserve pepper with vinegar*, and the corresponding simplified constituent tree is presented in Figure 2.11c. Without notice, the expressions are lemmatized: All inflections are excluded and only preposition, noun, verb, adjective and adverb stay in the expression. Constituent trees will be, by default, in the lemmatized form (Figure 2.11c).

Although constituent tree and syntactic tree are almost isomorphic, they have different representational focus. Syntactic trees show how expressions are parsed by generative grammar, while constituent trees present the subordinate relationships between linguistic expressions (and the concepts they correspond to), which are taken as a form embedding syntactic and semantic structural information. Since the constituent trees have lexical expressions for nodes, instead of grammatical categories, joining the trees would give rise to representation of concepts in a semantic network.

2.3.2 CTN as a Structure for Long-term Semantic Memory

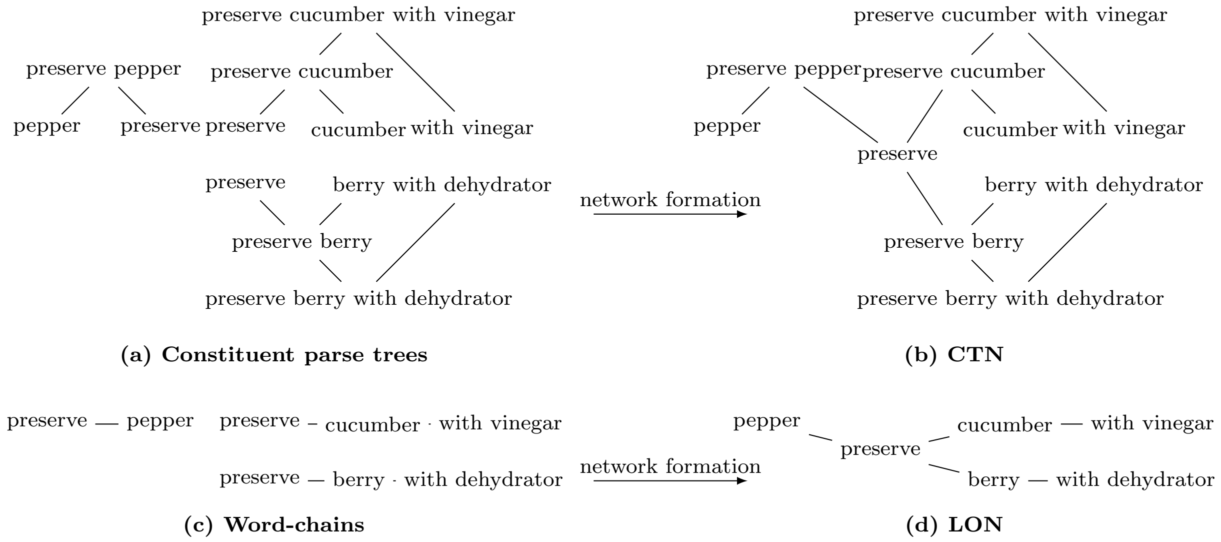
Given the graphical forms of the expressions, a CTN is formed by joining the sub-graphs at the shared lexical nodes (Figure 2.12 a-b), in a way similar to other distributional graphs (e.g. the co-occurrence graph in Figure 2.12 c-d). Despite the common computational procedure, there are two major differences between CTN and Co-occurrence Graph as representations of long-term semantic memory: the form of co-occurrence in the structure and the assumed linguistic capability of the representational system.

First, just like Co-occurrence Graph, CTN encodes co-occurrences of words in linguistic corpus. The difference lies in the pattern of the encoding: instead of encoding the words by the linear sequential order, CTN groups the words by their constituent relations in the expression. To be more clear, in a co-occurrence graph, two words are directly linked if they are adjacent in the input sequence, while in CTN, two linguistic units are immediately connected if one contains the other. From this aspect, the co-occurrence captured in a co-occurrence graph is also represented in the corresponding CTN, but in a more structured format.

One view of CTN is that the structure not only represent lexical items, but also complex expressions like phrase and sentence. This view has conflated encoding and representation: While there are phrase and sentence nodes in CTN, these nodes should be taken as constituency 'bridges' for words, but not representation of the complex expressions. CTN represents words. The constituent structure should be considered as the way CTN represent the complex relationships between multiple words that build up larger expressions (e.g. phrase). It should not be taken as a model that represents complex expressions in the long-term memory. For example, neither the node *preserve berry*, nor the sub-tree of *preserve berry* in the CTN in Figure 2.12b is a representation of the concept. The role of the phrase node and its links to the subordinate words is to provide the structural grouping of *preserve* and *berry*, showing that the two words has been grouped in the input data, and can be bound together to form a holistic concept during online language processing. Labeling of phrasal nodes by the expressions in CTN is for the purpose of illustration (Figure 2.13b), and a more precise presentation may not show the linguistic labels of the phrases (Figure 2.13c). The blank phrasal nodes indicate that the phrasal concepts are not statically represented in the semantic network like the lexical terms, and they probably do not have a corresponding phonological representation in the lexical network proposed in Collins and Loftus (1975). The blank phrasal nodes only contribute to the structure, so that lexical items can be grouped together dynamically to function as a holistic meaningful unit when such grouping is needed in language comprehension and production. I will show in Chapter 5 that the constituent structure encoding benefits in representation of multi-way lexical dependency.

From the perspective of memory encoding, the constituent structure encoded in CTN can be considered a type of memory trace that embeds certain syntactic information. It is similar to other memory traces as an exemplar instance to record the knowledge in input and form the long-term semantic memory. Its distinct feature lies in the specific type of information encoded in the trace: the constituent structure of the expression. The encoding requires syntactic adequacy: the system is assumed to have mastered constituent parsing to encode constituent structures. This brings it to the second difference between CTN and other distributional models.

The linguistic prerequisite for CTN and Co-occurrence Graph differ dramatically. While the latter needs mere word count, the former requires considerable capabilities in constituent parsing. Therefore, the two models are set up for different psychological phenomena. Co-occurrence Graph, but not CTN can be used



(a) Constituent parse trees

(b) CTN

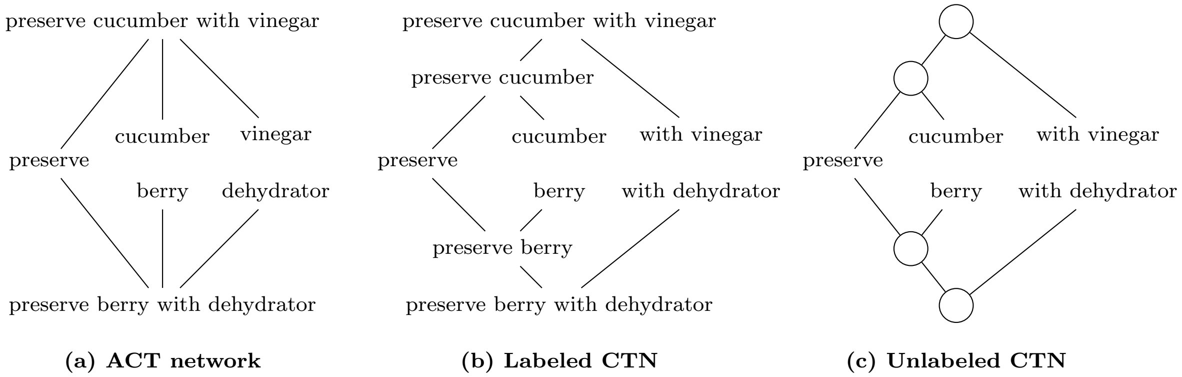
(c) Word-chains

(d) LON

Figure 2.12: Formation of the network structure in the Constituent Tree Network (CTN) and the Linear Order Network (LON) — a type of co-occurrence graph which only encodes adjacent co-occurrences — given the mini corpus *'preserve pepper', 'preserve cucumber with vinegar', 'preserve berry with dehydrator'*. **(a)** The input to the CTN consists of constituency-parsed trees for sequences in the mini corpus. **(b)** The network structure of the CTN is formed by joining the constituent trees at shared nodes. **(c)** The input to the LON consists of word-chains, formed by connecting adjacent words in the mini corpus. **(d)** The network structure of the LON is formed by joining word-chains at shared nodes.

to model early semantic structures. Nevertheless, CTN can still be applied to account for the formation of semantic structure after mastery of syntax. When lacking syntactic capability, the representational system may only encode linguistic input by sequential word count, and the trace left in long-term memory is in the form of co-occurrence clusters. With the development on grammar, the system becomes more competent at syntactic processing and might be more likely to do constituent parsing when processing linguistic input. Formation of CTN is built on the hypothesis that the constituent parsing occurred during language processing would leave traces (e.g. the constituent trees) for building up the long-term semantic memory. Need to mention that CTN is a model for semantic memory representation, and it is not responsible for explaining how acquisition of syntax and constituent parsing has happened. The representation of structured knowledge in CTN should be considered as a result of syntactic maturation, but not a cause of it.

In general, both Co-occurrence Graph and CTN can be considered as descendant of the semantic network in ACT model (Anderson, 1983). In ACT, a cognitive unit is a proposition converted to a sub-graph: a proposition node connecting to its constituent predicate and arguments. Propositions are joined by the shared predicates and arguments to form the semantic network (Figure 2.13a). In the ACT network, each cognitive unit is the accumulation of traces of corresponding events. This idea is followed by Co-occurrence Graph and CTN that operate on linguistic input. The distributional graphs are more 'linguistic' compared to ACT: They respect more to the linguistic forms rather than the abstract logical form of the event. Within distributional graphs, Co-occurrence Graph is less structured than ACT: Words are linked by the ordered co-occurrence in the input sequence. In contrast, CTN is more structured than ACT: the predicates and arguments are not only connected to the proposition node (as in ACT), but also hierarchically grouped by constituency. Therefore, while all three structures are able to account for basic word-word lexical relations, ACT and CTN may further speak to semantic phenomena in sentence processing, and CTN potentially give rise to a more detailed account for multi-way dependencies among words.



(a) ACT network **(b) Labeled CTN** **(c) Unlabeled CTN**

Figure 2.13: ACT network and CTN as joined traces of linguistic structures. **(a)** An ACT network as joined propositions by shared 'element' nodes. **(b)** A CTN formed by joining the constituent trees at shared nodes. **(c)** The same structure to (b), without labeling the non-word nodes. The complex concepts are not 'represented' directly in the network, rather, they are indirectly represented as potential groupings of the lexical terms, which can be accessed when certain combination of words are presented.

2.3.3 Spreading Activation and Phrasal Semantic Relatedness

It is assumed that the spreading-activation processes on different distributional graphs are identical. On a network, the semantic relatedness from a source node S to a target node T is considered as the amount of activation received by T from S over time. In co-occurrence graphs, the semantic relatedness between concepts can be directly implemented by the semantic relatedness defined between nodes, as the concerned concepts are words and uniquely map to nodes in the graphs. However, in CTN, the interest is in broader concepts, e.g. *red car*, *preserve onion with vinegar* or any concept with a linguistic expression. Although there are phrase and sentence nodes in CTNs, semantic relatedness involving complex expressions are not directly defined by the activation traveled between the corresponding nodes in the network. In this section, I give the formal definitions of spreading-activation based semantic relatedness on CTN, and explain the motivation of the definitions as relating to language processing.

Before the formal definition, it is necessary to talk about what kind of relations are of interests. Evaluation of words is more straightforward, as all words to evaluate are in the vocabulary, and have already occurred in the corpus. There would be no representation of a word, if it has not occurred in the input. However, a 'complex' concept (expressed in a complex expression) may have not occurred in the corpus, even if its constituent words have occurred. Due to the productivity of natural language (Fodor & Pylyshyn, 1988) and combinatorial explosion, the observed linguistic expressions can be a small subset of the meaningful expressions that people are able to process. In this dissertation, the interested concepts are not only those whose linguistic forms **have occurred** in the corpus, but all concepts whose linguistic form **can be constructed** from the lexical terms in the input. For example, given *fancy car* and *sad story* occurred in the corpus, the evaluation would cover *fancy story* and *sad car* as well as the observed phrases.

Moreover, in CTN, all interested concepts should be constituency-parsed: It is either a word, or an expression with a certain constituent parse. For example, the phrasal expression in linear form 'decorate cake with icing' at least has two meaningful constituents parses, one in the sense that 'the cake is decorated by icing' and the other 'the cake being decorated has icing on it'. Every expression under evaluation should be unambiguous in terms of its parse and for now the expressions under consideration should not have repetitive tokens (e.g. 'dog chase dog' and 'old boy treat young boy' are not of interests). Without notice, all constituent parses are binary.

The definition is carried out in three steps: First, relatedness between words on CTN is defined, followed by relatedness from an arbitrary structure to a word, and finally relatedness from an arbitrary structure to an arbitrary structure. Similar to relatedness defined in general distributional graphs, word-word lexical relatedness in CTN is also direction sensitive. That is, relatedness from a source S to a target T, denoted as $\mathrm{SR}(S, T)$ can be different from $\mathrm{SR}(T, S)$ in principle. Words would be denoted by lower-case letters and complex expressions by capital letters. Given a CTN $\mathbf{G}$, the semantic relatedness from w_1 to w_2 on G is defined as:

$$\mathrm{SR}(w_1, w_2) = \sum_{P \in \mathcal{P}_{w_1,w_2}, L(P) \leq d(w_1,w_2)+n} \mathrm{SR}^P(w_1, w_2) \tag{2.5}$$

which is exactly the definition 2.2 for relatedness between nodes in a general distributional graph. Then consider a structure A consisting of word tokens $w_1, w_2, ...w_k$ (in other words, the constituent tree of A has $w_1, w_2, ...w_k$ $(k \geq 1)$ as leaves. Then the semantic relatedness from the expression A to a word w is defined as:

$$\mathrm{SR}(A, w) = \prod_{1 \leq i \leq k} (\mathrm{SR}(w_i, w))^{\alpha_i} \tag{2.6}$$

where α_i is a weight for the depth of w_i in the constituent tree of the structure. Suppose in the constituent tree T_A for expression A, word w_i has depth d_i, e.g. the distance between w_i and the root of T_A is d_i, then the weight depth α_i for w_i in A is defined as:

$$\alpha_i = \frac{2^{d_i}}{\sum_{1 \leq j \leq k} 2^{d_j}} \tag{2.7}$$

In this way, deeper words is in the constituent structure of A matters more in any semantic relatedness from A. Now consider another arbitrary expression B, with constituent words $w'_1, ...w'_l$ $(l \geq 1)$, associated with depth weight $\beta_1, ...\beta_l$ in B, then the semantic relatedness from A to B is defined as:

$$\mathrm{SR}(A, B) = (\prod_{1 \leq i \leq l} \mathrm{SR}(A, w'_i))^{\beta_i} \tag{2.8}$$

which can be spelt out as:

$$\mathrm{SR}(A, B) = (\prod_{1 \leq i \leq k, 1 \leq j \leq l} \mathrm{SR}(w_i, w'_j))^{\alpha_i \beta_j} \tag{2.9}$$

Such a definition has covered semantic relatedness from arbitrary structures (without duplicated words) to arbitrary structures in CTNs, which weighs more on deeper constituent words and phrases in the structures. In this way, it is able to distinguish different structures with constituent words, e.g. the two parses for *decorate cake with icing* and *dog chase cat* versus *cat chase dog*.

Two explanations are needed for the definitions. First the relatedness are reduced to activation between word nodes, but not taken directly on phrase or sentence nodes despite the explicit encoding of these complex expressions. The reason are two folds. For one reason, the model need to evaluate expressions that have not been encoded in the CTN. For example, suppose *cut* and *pie* have occurred in the corpus, but not the phrase *cut pie*, there would be no phrase node in the CTN for *cut pie*. Nevertheless, since its constituent words must have been encoded in the network, a definition reducing to word-word relatedness is always legitimated. Moreover, for phrases that have been encoded in the network, different definitions would lead to different representational capability of the model. In short, defining phrasal relatedness directly by the activation from it to others results in a summation of its lexical relatedness but not the product of the lexical relatedness as given in 2.6 and 2.9. This brings to the second explanation to make. The question is: Why phrasal relatedness is defined as the product of lexical relatedness rather than their sum.

The most direct answer is that multiplication always counts the effect from both constituents, while summation is very likely to reduce the phrasal relatedness to lexical relatedness of one of its constituent. For example consider SR(*cut cake, knife*), in the case that SR(*cut, knife*) $=$ 0.3 is a lot larger than SR(*cake, knife*) $=$ 0.03 and SR(*grass, knife*) $=$ 0.01, a summation definition leads to SR(*cut cake, knife*) $=$ 0.33 and SR(*cut grass, knife*) $=$ 0.31. In both cases, the phrasal relatedness is approximately SR(*cut, knife*), and thus the phrasal relatedness is reduced to lexical relatedness. Such a situation is not ideal for the situation that *cut cake* and *cut grass* do have significant difference in terms of collocation with *knife*. Alternatively, the multiplicative definition would circumvent this issue. Regardless of the scales of the lexical relatedness, the product of lexical relatedness maintains the magnitude effect of both components in the phrasal relatedness, e.g. SR(*cut cake, knife*) $=$ 0.1 and SR(*cut grass, knife*) $=$ 0.05. The different scales for lexical relatedness result from the spreading-activation measure: the activation spread through a path is exponential to its length, which means linear difference in terms of graphical distance usually leads to exponential difference in terms of semantic relatedness. The phenomenon is common in distributional graphs. As a result, to prevent the phrasal relatedness from reducing to word-word relatedness, the definition need to be by multiplication instead of by summation.

2.4 Summary

In general, Distributional Graph is a modeling approach aiming at representation of semantic memory and language meaning. The structures are formed from memory traces in linguistic forms (word, phrase or sentence) after trivial (word co-occurrence) or sophisticated (constituent tree) syntactic processing. Encoding of the co-occurrence gives rise to a distributional representation of word relationships, while concatenation of the traces leads to indirect relation between not co-occurred words. Furthermore, the spreading-activation based measure of semantic relatedness defined on Distributional Graph may produce a graded measure of word-word and phrasal relations. The benefits of distributional representation of indirect lexical relations will be the focus of chapter 3. In addition, incorporating linguistic structures in the graphs may lead to a representation that can handle complex multi-way lexical relationships. In chapter 5, I show that explicit encoding of the constituent structure in CTN leads to its capability in compositional generalization that requires a representation of complex lexical dependencies.

Chapter 3

Linear Order Network

3.1 Introduction

Representing and processing semantic information is fundamental to language. The sequence 'babies sleep' is more easily processed than 'cars sleep', and the sequence 'ideas sleep' is even more difficult to process. Because all three sequences are grammatical and share the same syntactic structure, it is most natural to explain the difference in ease of processing at the semantic level. While neither cars nor ideas can sleep, it is easier for most people to metaphorically imagine a car sleeping than an idea sleeping. This example illustrates that semantic relatedness between words is an important part of people's ability to process and understand language.

The large number of words in natural languages, and the number of different ways that words can be related and paired, presents a daunting challenge for modeling semantic relatedness. Distributional models of semantic memory have been quite successful at modeling coarse-grained semantic tasks such as categorization (Landauer, Foltz, & Laham, 1998; Lund & Burgess, 1996; Huebner & Willits, 2018) and semantic priming (Griffiths, Steyvers, & Tenenbaum, 2007; Kumar et al., 2019; Landauer et al., 1998; Mandera, Keuleers, & Brysbaert, 2017). And despite the success and wide applicability of both co-occurrence-based vector space models and more recent neural network models like Word2Vec (Mikolov et al., 2013), BERT (Devlin et al., 2019) and GPT-3 (Brown et al., 2020), these models still have known shortcomings. For example, these models often fail at rudimentary language tasks involving structured relations or compositionality (Gershman & Tenenbaum, 2015; B. M. Lake & Murphy, 2020; Marcus, 2020). And while the semantic coherence of large language models like GPT-4 is truly impressive, the fact that they require orders of magnitude more data in order to achieve that performance raises serious questions about their feasibility as models of human semantic representation. As a consequence, questions remain about the capacity of these kinds of models to represent the various kinds of semantic relations that humans use to represent and comprehend language.

In this chapter, I address the question of what kind of semantic representations and processes might best support the human ability to produce graded semantic plausibility judgments of multi-word sequences. I discuss the advantages and disadvantages of representing word co-occurrence information and word similarity information in high-dimensional vector spaces, versus other approaches that represent this information in a connected graph (Anderson & Bower, 1974; Collins & Loftus, 1975; Deyne et al., 2016; Gentner, 1975; Rotaru et al., 2018). I then present a set of experiments designed to test the capacities of these models at predicting quantitative differences in semantic plausibility of predicate-argument pairs.

The chapter is organized as follows: First, I describe the importance of lexical semantic relatedness in determining the semantic plausibility of a sentence. Second, I review the distributional approach to semantic modeling, and describe some features or properties that differentiate how lexical semantic relatedness is acquired by different semantic models. The major properties examined are the representational structure of the model (graphical vs. spatial), and the type of information that is encoded (co-occurrence vs. similarity). Third, I discuss the construction of an artificial corpus that is built for the purpose of training and evaluating our models. Fourth, I report model performances in a selectional preference task that requires learning of semantic relations instantiated in the training data and making inferences about unobserved relations. Fifth, I explore the individual contributions of representational structure and information encoding type and their interaction to performance in this task. Finally, I discuss the results more broadly in the context of models of semantic development. To note, this chapter has been published as an article in the Psychology Review journal, titled 'Spatial versus graphical representation of distributional semantic knowledge', which can be found in https://psycnet.apa.org/record/2024-24081-001?doi=1.

3.1.1 Simple Sentences and Syntagmatic Relations

The study of semantic knowledge representation can be approached from multiple perspectives. One important starting point is establishing what behavioral or empirical phenomenon one is trying to model or explain. Historically, researchers have examined a diverse array of phenomena, including the learning of word meanings, semantic priming, categorization and typicality effects, judgements about factuality or plausibility, and sentence production and comprehension. In this paper, I focus on how representations and processes underlying lexical semantic relatedness contribute to plausibility judgments of multi-word sequences.

There have been many proposals for how to characterize the nature and kinds of relations that can affect the semantic plausibility of multi-word sequences. One of the most foundational distinctions is between words that have a syntagmatic relationship and words that have a paradigmatic relationship (de Saussure, Bally, Sechehaye, Reidlinger, & Baskin, 1960; Sahlgren, 2006). To recall, words that are syntagmatically related are words that can "go together" in language, operationally defined as linguistic co-occurrence or thematic relatedness. Syntagmatic relatedness is most often and most easily used to describe noun-verb relations (*drink-coffee*, *walk-dog*), but can also be used to describe adjective-noun relations (*hot-coffee*, *brown-dog*) and noun-noun relations (*cup-coffee*, *leash-dog*) where relatedness is defined as co-occurrence or joint participation in the same event. Syntagmatic relatedness can be distinguished from paradigmatic relatedness, which links words that are substitutable with one another in the linguistic structures in which they occur. For example, *tea* is paradigmatically related to *coffee*, because *tea* and *coffee* can be substituted with minimal impact on the meaning of the sentences or their semantic plausibility.

Syntagmatic and paradigmatic relations are related concepts. For instance, under many theories (like those using some form of distributional learning), syntagmatic relatedness can be inferred based on a combination of paradigmatic and syntagmatic relatedness. For instance, based on previous knowledge of the paradigmatic relatedness of *dog-puppy*, and the syntagmatic relatedness of *dog-leash*, one can infer that *puppy* and *leash* are also likely to be syntagmatically related.

When judging the semantic plausibility of sentences, syntagmatic relatedness is probably more influential. The sentences *Mary drank the coffee* and *Mary walked the dog* are plausible, and *Mary drank the dog* and *Mary walked the coffee* are not, because they mismatch in their syntagmatic but paradigmatic relations. In other words, it seems to make more sense to describe the plausibility of sentences in terms of how well the words *drink* and *coffee* go together (compared to *drink* and *dog*), and not in terms of their substitutability.

3.1.2 Selectional Preference

In this chapter, I am interested in how well different distributional semantic models acquire the selectional constraints on noun-verb pairs from linguistic data. I will use the term 'selectional preference' to describe constraints that determine which word pairs result in semantically plausible combinations (e.g. *babies sleep*) and which do not (e.g. *ideas sleep*). For example, *baby* is a better argument than *idea* for the verb *sleep*. Selectional preference is a quantitative (i.e. graded) phenomenon, and therefore requires a numerical score for each predicate-argument pair (Erk, Padó, & Padó, 2010). Thus, it is critical that model judgments are derived on a continuous as opposed to discrete ("related vs. unrelated") scale.

We can derive a measurement of selectional preference from most semantic models equipped with quantitative measures of lexical relatedness. For example, many models represent words as vectors, and their relatedness as distances in a vector space (Osgood, 1952; Smith et al., 1974; Griffiths et al., 2007; Landauer & Dumais, 1997; Jones & Mewhort, 2007; Mikolov et al., 2013; Huebner & Willits, 2018). Other models represent semantic relations in a graphical structure (network or tree-like), with connections that vary in strength, and/or with a spreading activation mechanism that allows for a quantitative degree of relatedness between words (Collins & Quillian, 1969; Collins & Loftus, 1975; Elman, 1990; McRae, de Sa, & Seidenberg, 1997; Miller, 1992; Nelson et al., 2004; Steyvers & Tenenbaum, 2005; Rumelhart & Todd, 1993; Rogers & McClelland, 2004). These models, despite varying considerably in their operational definitions of semantic relatedness, all provide a way to measure the relatedness between arbitrary word pairs. Therefore, it is possible to use model-derived relatedness scores as a proxy for selectional preference.

In this chapter, I use model-derived selectional preference judgments to evaluate the representational capabilities of different distributional semantic models. Importantly, to perform well in the evaluation, a model must not only assign higher semantic relatedness to word pairs that frequently co-occur, but also to word pairs that are more semantically plausible despite not having been observed during training (e.g. *cars sleep* is more plausible than *ideas sleep*). By comparing a model's selectional preferences to the corpus on which it was trained, we can make inferences about which models or properties of models are most useful for acquiring syntagmatic knowledge, and deploying that knowledge to make inferences about semantic plausibility.

Broadly speaking, this chapter presents a systematic comparison of many distributional models of semantics, with a specific focus on their ability to represent syntagmatic relations and using their learned representations to infer the semantic plausibility of observed and novel word sequences. There are four major differences between this approach and that of previous studies. First, the current work systematically explores differences between graphical models built from language-internal distributional data and more traditional spatial models built on the same amount and kind of data. While several graphical models have been proposed in the semantic modeling literature, they have rarely been compared to spatial models while systematically controlling for differences in their training data, learning algorithm, and other modeling parameters. Second, I conducted a systematic comparison of model properties and parameters to better understand their individual contributions and their interactions. The third difference is a focus on the quantitative rather than qualitative nature of semantic relatedness. Previous work has focused on qualitative analyses, such as distinguishing "related" versus "unrelated" word pairs (Bullinaria & Levy, 2007, 2012; Erk et al., 2010; Huebner & Willits, 2018) However, in this chapter I am more interested in the ability of models to reproduce a gradient of selectional preference that can be used to rank-order multiple word pairs by semantic plausibility. For example, given the pairs '*trap rabbit*' (observed), '*trap boar*' (unobserved, more plausible), '*trap water*' (unobserved, less plausible), previous evaluations required that a model only produce relatedness scores that differentiate the observed pairs from the unobserved. That is, a model only needed to correctly judge '*trap rabbit*' > '*trap boar*' and, separately, '*trap rabbit*' > '*trap water*'. In contrast, to succeed in our evaluation, a model must produce the correct rank-ordering '*trap rabbit*' > '*trap boar*' > '*trap water*'.

3.1.3 Interested Modeling Parameters

When semantic models are used to predict data and their performances are compared, it is not always clear why a particular model outperformed another. Unlike well-controlled experiments, computational models are complex, and usually vary in many ways at once. Which of these differences actually lead to differential performance?

In this chapter, I closely examine the effects of two of these dimensions, the **Representational Structure** (space vs. graph), and the **Encoding Type** in those spaces and graphs: word co-occurrences vs. distributional similarities based on word co-occurrence. Thus, I create four main classes of models: a co-occurrence space model, a similarity space model, a co-occurrence graph model, and a similarity graph model. The motivations for systematically exploring these two dimensions are both practical and theoretical. From a practical standpoint, graphical models are under-represented in contemporary semantic modeling, especially in research on automated information extraction from large naturalistic language data. Spatial models predominate in this area due to the simplicity and efficiency of algorithms used for training on large, unstructured data-sets. This state of affairs exists primarily due to practical reasons (e.g. availability of efficient algorithms for training and/or inference), but does not reflect theoretical or empirical advantages of spatial over graphical models. As such, the paucity of graphical models is potentially concealing as of yet unknown benefits that would result if they could be made to perform as efficiently as contemporary spatial models. Specifically, graphical models are rarely trained on or constructed with word co-occurrence data, and even less rarely using large corpora of natural language. To address this paucity of research, I propose a graphical model trained on distributional linguistic data (word co-occurrence). In the upcoming sections, I discuss theoretical considerations for our proposal. In particular, I will argue that graphical models, while under-represented in the literature, may be useful for addressing limitations on the representational abilities of contemporary spatial models.

3.1.4 A Limitation of Spatial Models

All spatial models represent words as vectors, whose dimensions may be populated with features specified directly by the modeler, from norming studies, or from naturalistic data like linguistic corpora. For example, the dimensions of a feature-based vector for *fish* might consist of the proportion of raters who judged *fish* on some feature dimension (e.g., 'can-fly', 'can-swim', 'has-beak'). Most proposed spatial models derive their semantic information from linguistic data from a naturalistic corpus, such as how often *fish* co-occurs with other words, or the number of times *fish* occurs in each of a set of documents. Most spatial models normalize these co-occurrence counts in some manner, such as converting co-occurrences to pointwise mutual information values (Bullinaria & Levy, 2007).

Given the large number of dimensions in models trained with vocabulary sizes that are often in the tens of thousands, dimensionality reduction is typically performed after vectors are populated with co-occurrence counts. Typically, this involves using an algorithm like Singular Value Decomposition (Landauer & Dumais, 1997), Latent Dirichlet Allocation (Blei, Ng, & Jordan, 2001), or Random Vector Accumulation (Jones & Mewhort, 2007). In addition to reducing the size of the vectors, dimensionality reduction also serves several other useful purposes. These procedures reduce the sparsity of semantic vectors (which, if unreduced, typically contain mostly zeros). Another consequence is that dimensionality reduction serves as a method for generating more abstract representations, since the resulting dimensions serve as latent variables that aggregate information in multiple rows or columns with similar covariance. For example, in a co-occurrence matrix where *robin*, *eagle*, and *crow* all co-occur with *wing, fly, feathers*, and *beak*, the columns for these words can end up being combined into one or more abstract latent dimensions on which words related to birds load highly compared to other words like *airplane* or *penguin* which share fewer features with birds.

Despite their widespread use in both NLP applications and cognitive modeling, spatial models suffer important limitations with respect to accounting for the full range of human semantic abilities. One critical issue is that spatial models cannot distinguish - in a principled fashion - between different types of semantic relations (such as syntagmatic vs. paradigmatic) in the same semantic space. Vector distance in a single semantic space typically represents some combined measure of multiple different types of relations, or emphasizes one or more relation types more strongly than others. To make this point clear, consider a semantic model whose similarity scores are used to guess the right answer to a multiple-choice test, where the cue is fast and the choices are speedy, slow, brown, and pointy. If the question is "What is the synonym?", versus "What is the antonym?", the right answer changes, and the semantic model cannot possibly use the most similar word to get both answers correct.

For a model to make principled distinctions between different kinds of relations, it would require one vector space for each relation type - an inelegant solution, especially if the number of relations one wishes to represent is large. Defenders of spatial theories of semantic cognition could question the necessity of principled distinctions between different types of lexical relations. However, their psychological reality is supported by the demonstrations that humans represent syntagmatic and paradigmatic relations and use them to construct indirect semantic relations - a signature of human cognition (Balota & Lorch, 1986; McNamara & Altarriba, 1988; Chwilla & Kolk, 2002).

The idea that spatial models struggle to distinguish different types of relations and to form indirect relations based on those distinctions is supported by their poor performance on tasks that require indirect relations. For example, Peterson, Chen, and Griffiths (2020) examined the performance of spatial models (using Word2Vec and GloVe) on a relational analogy task, of the form *king:man :: queen:woman*. This type of evaluation was first reported by Mikolov et al. (2013), on the basis that a model used to account for the structure of human semantic memory should be able to represent higher-order similarities, such as the similarity between king-man and queen-woman. Peterson et al. (2020) measured the similarity between word pairs (represented as vector differences) and correlated these scores to human judgements. Consistent with the idea that spatial models cannot explicitly represent indirect relations, the authors found that the models did not perform consistently across a diverse set of analogy types. While the models successfully predicted human ratings for the relation type CASE (e.g., *soldier-gun*, *plow-earth*), they performed poorly on other relation types, such as SIMILAR (e.g., *car-auto*, *simmer-boil*), CONTRAST (e.g. *old-young*, *buy-sell*), and NON-ATTRIBUTE (e.g. *fire-cold*, *corpse-life*). Of note, the spatial models performed well with syntagmatic relations (*soldier-gun*, *plow-earth*), but poorly with paradigmatic relations (*simmer-boil*, *old-young*), and especially poorly with those that indirectly bind the two (e.g., fire-cold can be decomposed into *fire-warm* and *warm-cold*).

It is possible that the failure of spatial models to succeed across all relational analogy types is because their representational substrate is suboptimal for flexibly combining different types of relatedness among words, and to use such combinations to infer the strength of indirectly related word pairs. The experiments below are designed to test this limitation explicitly. Strong performance on the selectional preference task requires 1) the ability to represent both paradigmatic and syntagmatic relatedness in the same model, and 2) leveraging both kinds of relatedness simultaneously to predict indirect relatedness. I show that spatial models tend to represent either syntagmatic or paradigmatic relatedness, and that their failure to represent both in a principled manner limits their ability to infer indirect relatedness.

3.1.5 A Limitation of Graphical Models

In graphical models, words correspond to nodes in a graph, and relations among words are represented as edges between nodes. One advantage of the graphical structure over spatial models is their straightforward encoding of indirect relations. For example, the indirect relation stripe-lion can be represented as a chain of edges that connects stripe to tiger, and tiger to lion.

While this property of graphs makes them promising for inferring indirect relations, previously proposed graphical models suffer from a limitation not shared by existing spatial models: In contrast to spatial models, existing graphical models are typically populated either with hand-specified relations (Collins & Quillian, 1969; Steyvers & Tenenbaum, 2005) or with normative word association data (Deyne et al., 2016; Kenett, Kenett, Ben-Jacob, & Faust, 2011; Kenett, Beaty, Silvia, Anaki, & Faust, 2016; Kenett, Levi, Anaki, & Faust, 2017; Kumar et al., 2019; Steyvers & Tenenbaum, 2005). Due to the differences in the materials used as input to graphical models compared to spatial models, which are often trained on large corpora as opposed to normative association data, it is impossible to make strong conclusions about whether differences in capabilities of the two model types are due to representational structure or information source. In addition, these normatively formed semantic networks are derived from established relations in human semantic memory and are thus useful only for characterizing the end-state of semantic development, rather than the process by which the semantic network is formed. Put differently, most graphical models of semantic knowledge were developed to account for the structure of semantic memory, not its development. In this chapter, I am concerned with the latter: How can we build semantic networks from language input with both efficiency and developmental plausibility in mind?

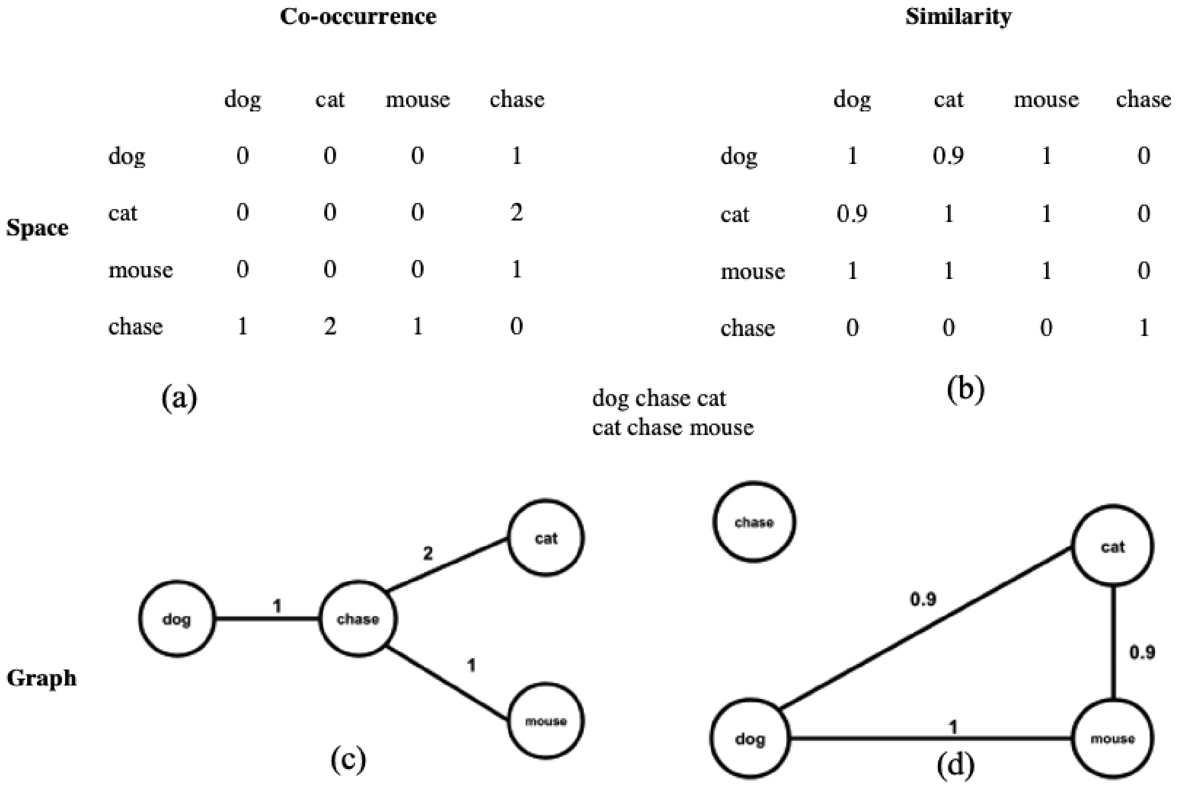
The lack of contact with learning and developmental processes is a substantial issue for graphical models. The methodological gap between language input and linguistic representations must be filled for graphical models to be trained on large, naturalistic corpora. There have been few recent investigations of this issue. That said, while some work on semantic networks by Hills, Maouene, Riordan, and Smith (2010), Deyne et al. (2016), and Rotaru et al. (2018) has examined the construction of graphs directly from corpus data, these models were not trained on the scale at which spatial models are often trained. This makes existing graphical models unsuitable for comparison with many spatial models trained on corpora consisting of many millions and even billions of words. Given the current state-of-the-art, the best spatial models may outperform the best graphical models simply because they were trained on much larger corpora. Such a comparison however would be more valid if graphical models could be trained on data that is equally large as, and in a manner comparable to, the standard method for training spatial models.

3.1.6 Combining Spatial and Graphical Models

To summarize, while spatial models can be efficiently trained on large corpora of natural language, graphical models excel at making inferences about indirect semantic relations. A similar point was recently made by Kumar, Steyvers, and Balota (2022), who wrote that:

> ...it seems most likely that modern distributional models (specifically multimodal DSMs) provide a promising account of learning meaning from natural environments, whereas semantic network accounts provide useful conceptual tools to probe these representations and the processes that operate upon these representations. (p.19)

Given the complementary strengths of the two modeling approaches, a system that can take advantage of both strengths would be an important contribution to the field of semantic modeling. Here, I argue that such an integration can be accomplished. I show that the Co-occurrence Graph (Linear Order Network, or LON) model inherits the complementary strengths of graphical and spatial models but does not suffer the limitations of the latter. Specifically, the data structure of the proposed model is graphical, but the mechanism for deriving the graphical structure is based on the same co-occurrence counting methods used to construct many spatial models from corpus data. I then test the feasibility of using co-occurrence frequency, and word similarity as the basis for connecting word nodes. The latter is the subject of prior work (Rotaru et al., 2018), and the current comparison builds upon and extends this work.



Co-occurrence

Space

	dog	cat	mouse	chase
dog	0	0	0	1
cat	0	0	0	2
mouse	0	0	0	1
chase	1	2	1	0

(a)

Similarity

	dog	cat	mouse	chase
dog	1	0.9	1	0
cat	0.9	1	1	0
mouse	1	1	1	0
chase	0	0	0	1

(b)

dog chase cat
cat chase mouse

Figure 3.1: A schematic illustration of the construction of the four model types investigated, with text in center (dog chase cat/cat chase mouse) the toy corpus for illustration. (a) spatial co-occurrence representation in matrix form where rows are word vectors whose elements contain the frequencies of co-occurrence with words that label the columns. (b) spatial similarity representation derived from (a), where each entry is the similarity between co-occurrence word vectors. (c) graphical co-occurrence representation derived from co-occurrence matrix (d) graphical similarity representation derived from the similarity matrix in (b).

Before describing the procedure used to train the proposed model, I first illustrate the way in which spatial distributional models are typically created. Given a toy corpus consisting of two sentences 'dog chase cat. cat chase mouse', a four-by-four matrix of bi-directional word-word co-occurrences with adjacent words (i.e., with window size = 1) can be formed, shown in Figure 3.1a. Here, the entry (i, j) in the matrix is the bi-directional co-occurrence between i'th and j'th word. From this word co-occurrence matrix, a word similarity matrix (shown in Figure 3.1b) can be constructed using some pairwise comparison between pairs of row vectors (such as the cosine of the angle between vectors). Thus, within the spatial framework, two kinds of models can be constructed: The relatedness of words i and j can be obtained either from the co-occurrence count in cell (i, j) of the co-occurrence matrix, or the similarity score in cell (i, j) of the similarity matrix. Henceforth, I will refer to these two types of models as Co-occurrence Space and Similarity Space models, respectively.

The same matrices can be reused to construct corpus-derived graphical models. Because both the columns and rows refer to words we wish to represent, the co-occurrence matrix and the similarity matrix can each be considered an adjacency matrix of a graph. When deriving adjacency information from the co-occurrence matrix, the words become nodes in the graph, and are connected by an undirected weighted edge proportional to the value of the corresponding entry in the co-occurrence matrix (provided an entry is non-zero). I refer to the resulting model as a Co-occurrence Graph (illustrated in Figure 3.1c). Similarly, we can construct a graphical model in which the weights of edges are derived from the similarity score of the words in the similarity matrix. I refer to the resulting model as a Similarity Graph (illustrated in Figure 3.1d). In total, we consider four types of models that vary along two dimensions, namely Representational Structure (graph vs. space) and Encoding Type (co-occurrence vs. similarity) in the other.

3.1.7 Artificial Corpus

Semantic models built from linguistic corpora have several advantages. They have the practical advantage that obtaining a corpus is cheaper and easier than obtaining an equally sized normative dataset, and much easier than hand-labeling semantic relations. And perhaps most importantly, corpus-based models have a theoretical advantage, in that their structure and complexity is critical to distribution-based theories of knowledge representation. However, semantic models built from large naturalistic corpora have a matching disadvantage: the size of the corpus, and the complexity of the information contained in it, can make it difficult to understand precisely what aspects of the input contribute to the success of a model. Popular neural network models such as GPT-3, Word2Vec, and BERT, are trained on natural language corpora containing millions or billions of words, represented across millions or even billions of parameters. This can make understanding what kind of knowledge they have acquired very difficult.

There are two different strategies for researchers aiming to develop better models of human semantic knowledge. The first (and most common) approach is to focus on their fit to empirical data, such as sentence reading times, eye-tracking and EEG data obtained during sentence processing, semantic priming data, and normative judgements (of relatedness, similarity, categorization, and semantic facts). The second approach, and the one that is pursued in throughout this dissertation, is to design artificial datasets that are created to highlight specific formal scenarios and can be used to test a model's formal capabilities.

Using an artificial corpus has many advantages when trying to understand the basic workings of complex distributional semantic models, such as large language models. The first and most obvious is that it allows one to precisely control the language, such that the only important sources of variability in the language are those that match the theoretical question that is being tested. This allows for control of the vocabulary size, the syntactic structure, and the semantic relationships in the language, allowing for more controlled tests of the models' abilities. A second advantage is that limiting the size and complexity of the language allows the models to be more interpretable. Instead of needing to understand and interpret billions of parameters, we can deal with models that have only dozens or hundreds. This makes understanding what the model can do much easier. There is a growing number of studies that have made use of artificial language corpora to understand the representations learned by complex models (Asr & Jones, 2017; Elman, 1990, 1991, 1993; Perruchet & Vinter, 1998; Ravfogel, Goldberg, & Linzen, 2019; Ri & Tsuruoka, 2022; Rubin et al., 2014; Clair, Monaghan, & Ramscar, 2009; Tabullo et al., 2012; Wang & Eisner, 2016; White & Cotterell, 2021; Willits et al., 2015).

3.2 Method

3.2.1 An Artificial Corpus to Describe A Simulated World

To generate the linguistic corpus, I first create a miniature world simulation, which consists of agents that have goals, and events that occur as those agents set out to achieve their goals. I then generate a linguistic corpus composed of simple sentences that describe the sequences of events that occur in the world. The events in the simulated world take place in temporal order which is determined by pre-designed intrinsic event structures. Although both the world and the language generated from it are simplified compared to the physical world and language in reality, it has the advantage of being completely under control. Unlike many artificial grammars which generate sentences that are semantically isolated from one another, consecutive sentences in our corpus relate semantically with one another, due to the predefined event structures.

Unlike previous works (Erk et al., 2010) that used selectional preference to evaluate semantic models trained on naturalistic corpora, I incorporate selectional preferences into the artificial corpus. To be more specific, I manipulate which nouns can be agents or patients for each verb, with what frequency or probability. The detailed procedure followed for generating the events in the world and then for generating the corpus that describes those events can be found in Appendix A of the journal article. Here, I briefly summarize the structure of the world, and the corpora generated from it for examining the models.

The Simulated World

The Agents and objects (organized into a taxonomy) take actions in the simulated world. The full set of agents and objects (and the categories to which they belong, noted in all caps) as well as the full list of actions (and the arguments for those actions) are shown in Table 3.1. As listed in Table 3.1, there are 5 categories of agents and 5 categories of objects. A member of AGENT (referred to simply as agent) in the world may appear as the agent of a verb, while other entities may not. Each category consists of three entities, also referred to as members. For the category CARNIVORE, one out of three members is randomly selected, and for other categories two out of the three members are randomly selected so that there are 19 entities (nouns) across the 10 categories when the world is initialized. And there are 20 possible actions (a fixed set of verbs) in each simulation.

The set of actions that can be performed by an entity is dictated by its category (e.g., HUMAN). A selection of these rules are shown in Table 3.2. Table 3.2 can be used to determine which nouns may occur with which verbs, in either the agent or patient role. These rules manifest as selectional preferences in the corpus. Need to note, we labeled the nouns in the corpus by both their name, e.g., ‘tiger’, ‘water’, and their thematic role. For example, in the sentence ‘tiger chase rabbit’, the full labels for the entities are ‘tiger$_a$’ and ‘rabbit$_p$’. In contrast, in the sentence ‘rabbit sleep’, the full label for the noun is ‘rabbit$_a$’. Consequently, ‘rabbit$_a$’ and ‘rabbit$_p$’ are taken as distinctive word types in the corpus and by each semantic model. The

Table 3.1: Categories of Objects in the Simulated World. Upper-case words denote categories of entities, while lower-cased words denote entities or actions in the simulated world. Brackets are used to group entities that belong to the same category. Parentheses are used to group entity categories involved in action, as either agent (in first position), or patient (in second position).

Animate Agent	**Inanimate object and Location**
AGENT = [HUMAN, NONHUMAN]	FOOD = [NUT, FRUIT, PLANT, AGENT]
HUMAN = [Mary, Kim]	NUT = [walnut, cashew almond]
NONHUMAN = [CARNIVORE, HERBIVORE]	FRUIT = [apple, pear, peach]
CARNIVORE = [wolf, tiger, hyena]	PLANT = [grass, leaf, flower]
HERBIVORE = [S_HERB, M_HERB, L_HERB)]	LIQUID = [water, juice, milk]
S_HERB = [rabbit, squirrel, fox]	Location = [river, tent, fire]
M_HERB = [boar, ibex, mouflon]	
L_HERB = [bison, buffalo, auroch]	

Intransitive action	**Transitive action**
rest(AGENT)	go_to(AGENT, LOCATION)
search(AGENT)	chase(AGENT, AGENT)
lay_down(AGENT)	drink(AGENT, LIQUID)
sleep(AGENT)	eat(AGENT, FOOD)
wake_up(AGENT)	reach(HERBIVORE, PLANT)
get_up(AGENT)	catch(CARNIVORE or HUMAN, HERBIVORE)
yawn(AGENT except S_HERB)	peel(HUMAN, FRUIT)
stretch(AGENT except S_HERB or L_HERB)	crack(HUMAN, NUT)
	throw(spear)_at(HUMAN, L_HERB)
	shoot(HUMAN, M_HERB)
	trap(HUMAN, S_HERB)
	stab(HUMAN, S_HERB)
	butcher(HUMAN, NONHUMAN)
	gather(HUMAN, FOOD)
	cook(HUMAN, NONHUMAN)

motivation for this was to preserve the semantics of the world, which is based on asymmetries with respect to which entity can perform which action as either agent or patient. The co-occurrence encoding, however, is invariant to such distinctions in situations in which word-order does not correlate with thematic role. For example, ‘rabbit’ and ‘tiger’ in the transitive sentence ‘tiger chase rabbit’ are coded identically when counting co-occurrences, despite their difference in thematic roles. If the corpus has many sentences like ‘tiger chase rabbit’, ‘rabbit’ will be distributionally similar to ‘tiger’, despite ‘rabbit’ not being allowed to perform the chase action, as dictated by the rule book (in Table 3.2).

Implicit in this table are semantic facts which we may use to compute the target preferences for our selectional preference task. For example, ‘tiger’ is more similar to ‘Mary’ than to ‘rabbit’ when they are in the agent role. This semantic fact in the world is quantified by comparing the row vectors associated with each entity in Table 3.2.

The Corpus

Corpus generation proceeded as follows: At each time step, agent entities took turns performing actions contingent on their drive levels and the event structure in which they were situated. If an agent successfully carried out an action, a sentence describing the action was generated and added to the corpus using the

Table 3.2: A Subset of Rules and Hypothetical Frequencies of Events. The rules govern what entities were allowed to perform which actions in the simulated world. The subscripts a and p denote whether a word was an agent or patient in each sentence. The co-occurrence rules differed based on agent/patient status, and therefore so did the co-occurrence frequencies.

	Rules				Hypothetical Frequencies			
Noun	crack	chase	eat	drink	crack	chase	eat	drink
Mary$_a$	1	1	1	1	2	3	5	6
tiger$_a$	0	1	1	1	0	2	7	4
rabbit$_a$	0	0	1	1	0	0	3	3
Mary$_p$	0	1	0	0	0	6	0	0
tiger$_p$	0	0	0	0	0	0	0	0
rabbit$_p$	0	1	1	0	0	10	7	0

formulas S1 = Agent + IntransVerb and S2 = Agent + TransVerb + Patient. The formula that was chosen depends on the verb type. Every sentence was followed by an optional utterance boundary marker (a minor parameter, see Table 3.3). I used the 10 simulations to generate 10 different corpora, with differences that resulted from which specific members of each category were selected during random initialization of entity locations and drive values. Despite this variation, the general semantic structure of the world was extremely consistent across runs. On average, each corpus had 14330 sentences and 50204 word tokens. I experimented on different corpus sizes and decided that the current size was sufficient to encode the stipulated semantic constraints. The vocabulary consisted, on average, of 10 agent nouns, 17.9 patient nouns, and 22.9 verbs of which 14.9 are transitive.

Due to the semantic constraints posed in the simulated world (left, Table 3.2), only a subset of verb-noun combinations are allowed to occur in the corpus. The frequency of their occurrence in the corpus reflects the number of times their associated events occur in the simulation. To provide a sense of their frequencies, the same table (on the right) also shows the frequencies of their occurrence derived from one hypothetical run of the simulation. Due to the tight coupling between the world and the corpus, the distributional statistics in the corpus can be said to be grounded in the statistics of the simulated world. As a consequence, this enables us to use the generated distributional data not only for training of semantic models, but as the criterion for testing the representational capabilities of the semantic models.

3.2.2 Experimental Design for Testing Semantic Models

As described, this chapter is mainly concerned with comparing the relative performance of spatial versus graphical models (controlling for the difference between co-occurrence and similarity data) to represent the semantic structure of the simulated world. However, there are a number of other parameters and factors that must be fixed when building co-occurrence matrices, and which have been shown to have a large effect on their performance (Bullinaria & Levy, 2007, 2012; Sahlgren, 2006). Because these parameters are not of the primary concern here, I built a set of models that varied along these minor parameter dimensions, and then computed the mean and best performing model for each of the four main conditions, but did not further investigate the theoretical and empirical effects of the minor parameters. Consequently, I trained 216 different co-occurrence space and graph models, each varying on one of six minor parameters, related to corpus pre-processing or the calculation of co-occurrences. These manipulations are shown in Table 3.3.

Table 3.3: Minor Parametric Variations. All combinations led to 216 different models based on how co-occurrences were counted and normalized.

Parameter	Options
Periods included as words in the corpus	yes, no
Co-occurrence cross sentence boundary	yes, no
Co-occurrence window size	1, 2, 7
Co-occurrence window weight	flat, inverse
Co-occurrence window direction	forward, backward, summed
Normalization	non, row-log, PPMI

These 216 models were also run for the similarity space and graph models. However, two additional parameters were varied for the similarity models, shown in Table 3.4, namely (i) whether the co-occurrence matrix was reduced via SVD, or left in its original form, before similarities were computed, and (ii) which similarity metric was used (the distance between vectors, the cosine of the vectors' angle, or the correlation between the vectors). While not being the primary interest of this work, I included these manipulations to strengthen confidence in the primary conclusions. Further details on how these parameters affected the creation of the co-occurrence matrices and the calculation of the co-occurrence and similarity scores can be found in Appendix C in the journal article. For more information about effects of minor parameters on the performance of spatial models, readers may refer to Sahlgren (2006), Bullinaria and Levy (2007), Bullinaria and Levy (2012), and Rubin et al. (2014).

Our experimental design is composed of 6 minor parameter dimensions (Table 3.3), each nested in one of two major parameter dimensions (Table 3.4), resulting in 3024 model variations to examine. I trained all 3024 model variations on the artificially generated corpus, and obtain, for each model, a semantic relatedness table that contains relatedness scores for every word pair in the vocabulary. Note that the same noun may occur twice in the vocabulary if it is used both as an agent and as a patient. I evaluate each model using the selectional preference task in which a model's semantic relatedness (SR) scores is compared to the target scores derived from the corpus statistics. How semantic relatedness is computed for different models is discussed in the upcoming sections.

Table 3.4: Number of models in each of our major parameter conditions. For each data structure, there are 7 encoding types, based on different methods for computing similarity.

Data Structure	Encoding Type						
		Unreduced Similarity (no SVD)			Reduced Similarity (SVD)		
	Co-occurrence	Distance	Cosine	Correlation	Distance	Cosine	Correlation
Spatial	$n = 216$	$n = 216$	$n = 216$	$n = 216$	$n = 216$	$n = 216$	$n = 216$
Graphical	$n = 216$	$n = 216$	$n = 216$	$n = 216$	$n = 216$	$n = 216$	$n = 216$

3.2.3 Computing Semantic Relatedness

For the co-occurrence space models, semantic relatedness was calculated in the following way. Relatedness between two words in the co-occurrence space at indices i and j was calculated as the simple co-occurrence value in the co-occurrence matrix (after normalization, for models that included it). One complication is that these co-occurrence matrices were not always symmetric. For example, for models that track co-occurrences in the forward direction only (from the word in row i to the subsequent word in column j), the cell (i, j) encodes how often j followed i, and the cell (j, i) encodes how often i followed j. As I use these co-occurrence values to predict relatedness in ordered sentence contexts, I always used the cell that corresponded to the appropriate order given the sentence. For example, if trying to measure the semantic plausibility of the sentence 'Mary$_a$ trap rabbit$_p$', I used the cell corresponding to the frequency of 'rabbit' occurring after 'Mary$_p$'. Due to this asymmetricity, from now on I denote $\mathrm{SR}(w_1, w_2)$ as the semantic relatedness from word w_1 to word w_2, in that order. In this case, $\mathrm{SR}(\mathrm{Mary}_a, \mathrm{rabbit}_p)$ denotes the semantic relatedness from 'Mary$_a$' to 'rabbit$_p$', evaluated by the cell $(\mathrm{Mary}_a, \mathrm{rabbit}_p)$ of the co-occurrence matrix. For the similarity space models, relatedness was computed by taking each word's co-occurrence row vector, and computing its similarity to the other word's vector, using the similarity metric (cosine, distance, or correlation) for that model. For both Similarity graph and Co-occurrence graph, the semantic relatedness $\mathrm{SR}(w_1, w_2)$ followed the definition 2.2, with the upper bound modifier $n = 0$:

$$\mathrm{SR}(w_1, w_2) = \sum_{P \in \mathcal{P}_{w_1, w_2}, L(P) \leq d(w_1, w_2)} \mathrm{SR}^P(w_1, w_2) \tag{3.1}$$

In other words, $\mathrm{SR}(w_1, w_2)$ is defined as the amount arrived at w_2 initialized from w_1, through the shortest paths between the two words.

3.2.4 Evaluation

To evaluate each model, it needs some way to establish the "right answer" that determines which words should be syntagmatically related. There are, in principle, three ways this could be done. The first would be to say relatedness is a binary feature, defined as whether the words were allowed to occur according to the grammar that was used to generate the simulated world's events. Under this definition, ' Mary$_a$' and 'eat' would be related (because Mary can eat), and 'rabbit$_a$' and 'shoot' are not related, because in the simulated world, rabbits cannot shoot. This particular evaluation was not the choice, as the interests lied in seeing which models could produce the graded forms of relatedness that humans demonstrate. A second option would be to use the actual frequencies in the corpus. This would produce graded differences but is less interesting as the co-occurrence space model would be best at this, because that is literally what that model is tracking. Such an evaluation would also lack the interesting property of demonstrating any form of generalization beyond the data that humans do show. In the simulated world, Mary can shoot boars, but cannot shoot rabbits or water. Nonetheless, we might expect that real humans would find it more plausible that Mary can shoot rabbits than that she can shoot water. This is because, in the simulated world, rabbits are more similar to boars than to water, in terms of the other semantic roles they can fill. Using simple co-occurrence frequencies as the gold standard against which the models are compared would not allow us to test a model's ability to make these generalizations (or, more accurately, would punish them for doing so).

To decide the "right answer" (i.e., the semantic plausibility of a noun-verb pair), I opted for a third option, namely a similarity-based procedure. In this procedure, the target scores were directly derived from the corpus co-occurrence statistics. Importantly, however, the implementation differed depending on whether a word pair is directly or indirectly related. In the former case, the target relatedness score is based on the co-occurrence frequency of a given word pair; in the latter case, the target relatedness score is based on the distributional similarity between a given noun and the nouns that co-occurred most frequently with a given verb. The higher the similarity between a noun that did not previously co-occur with a verb and nouns that did, the higher the target relatedness score. The method of quantifying semantic relatedness is a simplified version of prior methods based on similar ideas (Erk et al., 2010).

Computing the Target Relatedness

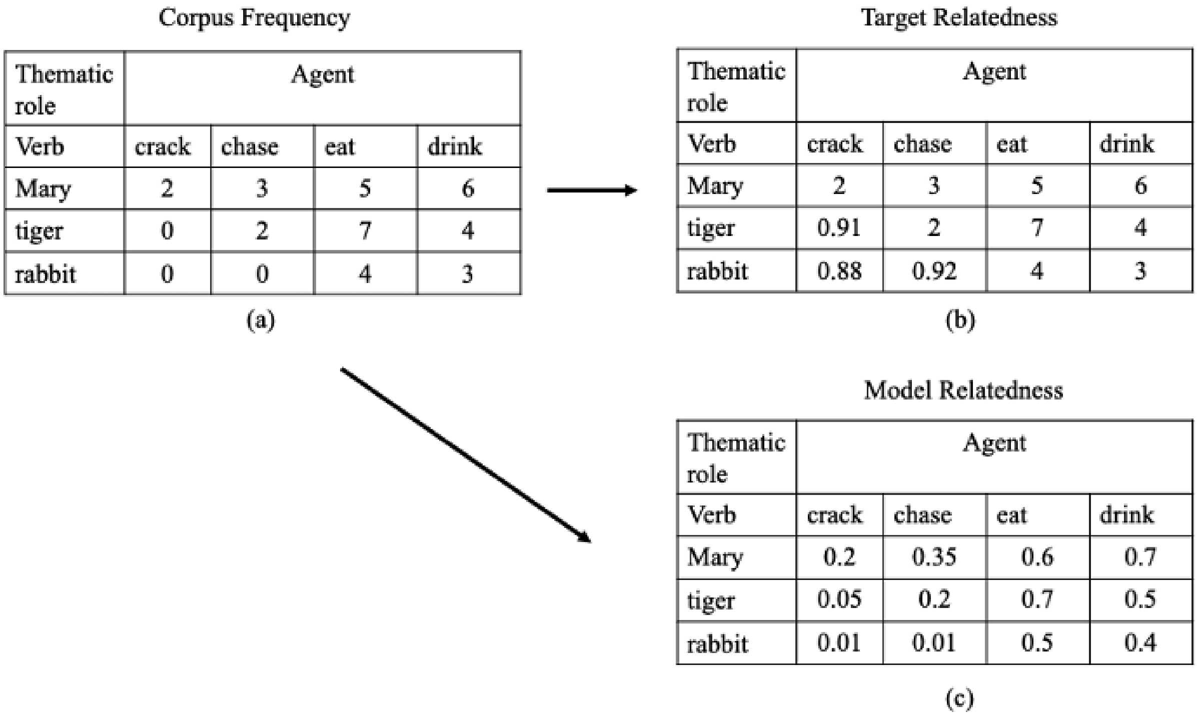
To illustrate how I computed target relatedness, consider the relatedness between the verb 'crack' and all nouns that have occurred in the corpus in agent position. The relatedness score is computed differently depending on whether a noun co-occurred with the verb or not. For nouns that co-occurred with 'crack' in agent position, the semantic relatedness is simply the co-occurrence frequency. This is illustrated in Figure 3.2. Because Mary$_a$ co-occurred with crack three times, the target relatedness for the pairMary$_a$-crack is 2 (Figure 3.2b). The computation is different for nouns that did not co-occur with a given verb in the corpus. For instance, 'tiger$_a$' did not co-occur with 'crack' (it is not allowed to be the agent of 'crack'). To quantify the relatedness of the pair tiger$_a$-crack, I obtained the nouns that did co-occur with crack in the corpus in agent position, and calculated the average of the cosine similarity between 'tiger$_a$' and each of those nouns. Note that the vectors used for similarity computation are the row vectors in Figure 3.2a. Given the example presented in Figure 3.2, I computed the cosine similarity between 'tiger$_a$' and 'Mary$_a$' , which resulted in a value of 0.91 (Figure 3.2b). When there are two nouns that co-occurred with a given verb, I repeated the procedure, and averaged the resulting cosine similarities. For instance, the relatedness of rabbit$_a$-chase is the average of the cosine similarity of tiger$_a$-chase and Mary$_a$-chase, which ends up being 0.88.

In this way, the relatedness of a noun and verb that did not co-occur is always smaller than the relatedness of a noun and verb that did co-occur, since the co-occurrence frequency for a pair that co-occurred in the corpus is at least 1, while the cosine similarity is upper bounded by 1. In general, relatedness is highest for pairs where the noun frequently co-occurred with the verb, less high for pairs where the noun co-occurred with the verb less frequently, and lower still for pairs where the noun did not co-occur with the verb. Importantly, among the latter group of pairs, semantic relatedness is graded, such that pairs where nouns are more similar to nouns that co-occurred with the verb score higher than pairs where the noun is less similar to nouns that co-occurred with the verb.

Comparing Target and Model Relatedness Scores

Figure 3.2 illustrates the process of how corpus-derived and model-derived relatedness scores are compared, using hypothetical data. I used Spearman correlation to evaluate the semantic relatedness scores produced by the model (Figure 3c): each column in Figure 3.2c is correlated with the corresponding column in Figure 3.2b. The resulting correlations are averaged to obtain the performance of a single model. As an example, consider the column representing the verb 'crack' to agent nouns in Figure 3.2b and 3.2c, namely [2, .91, .88] and [.2, .05, .01], respectively . The Spearman correlation of two column vectors is 1. The average across columns, and across all 10 corpora is the performance reported in the results. By averaging across multiple corpora, high performance would be a strong evidence that the model is successful at learning, representing, and inferring the fine-grained semantic relatedness between nouns and verbs.

Verbs differed widely in the proportion of nouns that did and did not co-occur with them in the corpus. For instance, the verbs 'eat' and 'drink' co-occurred with many nouns in agent position in the artificial corpus. This is illustrated in Figure 3.2. The nouns 'Mary', 'tiger' and 'rabbit' all co-occurred with eat and drink at least once in agent position. Other verbs are more selective; for instance, verbs like 'chase' and 'catch'



Corpus Frequency

Thematic role	Agent			
Verb	crack	chase	eat	drink
Mary	2	3	5	6
tiger	0	2	7	4
rabbit	0	0	4	3

(a)

Target Relatedness

Thematic role	Agent			
Verb	crack	chase	eat	drink
Mary	2	3	5	6
tiger	0.91	2	7	4
rabbit	0.88	0.92	4	3

(b)

Model Relatedness

Thematic role	Agent			
Verb	crack	chase	eat	drink
Mary	0.2	0.35	0.6	0.7
tiger	0.05	0.2	0.7	0.5
rabbit	0.01	0.01	0.5	0.4

(c)

Figure 3.2: An Illustration of the Model Evaluation Procedure. (a) First, a selection of noun-verb co-occurrences are counted in the corpus. (b) Using this information, distributional similarities between nouns are computed using pairwise cosine between row-vectors.(c) Next, model-derived semantic relatedness scores are computed for the same selection of noun-verb pairs. Both the corpus-derived and model-derived relatedness are flattened to a single column vector, and their correlation is computed. The resulting value is indicative of a model's ability to represent fine-grained semantic relatedness.

co-occurred with a smaller set of nouns in agent position. The same is true of pairs where the noun was in patient position.

Given that these two situations require different kinds of inference by the model, the model evaluation was split into two experiments. In Experiment 1, I evaluated the performance on verbs that directly co-occurred with every noun. In Experiment 2, I evaluated performance on pairs in which the verbs co-occurred (directly) with some but not all nouns. I call the word pairs in these two experiments 'directly related stimuli' and 'indirectly related stimuli', respectively. As an illustrative example in Figure 3.2a, the left half, i.e. word pairs that include 'crack' and 'chase' are indirect stimuli, as they do not directly co-occur with all agents. On the other hand, the right half of Figure 3a, i.e. the word pairs that include 'eat' and 'drink' are direct stimuli. To evaluate a model, I compared each verb column in the model relatedness table to the corresponding column in the corpus-derived target relatedness table using the Spearman correlation. I averaged Spearman correlation across all direct stimuli and indirect stimuli separately, as the measures of a model's performance in Experiment 1 and 2, respectively.

What the average Spearman correlation tells us is how close a model's relatedness judgments are to the target relatedness. There are many alternative methods I could have chosen, and there is no straightforward single "correct" method, especially considering we cannot yet precisely formulate the computational procedure that underlies semantic inference in human. That said, I tried multiple alternatives to forming the target relatedness (the verb-noun matrix normalized with PPMI/row-log, or no normalization; and similarity calculated with either cosine or 2-distance). The results did not differ qualitatively from those presented here.

To perform well on this task requires inferences based on two sources of information. First, a model should rank highest those noun-verb combinations which occur most frequently in the corpus. For example, 'Mary$_a$' is the only agent of the verb 'crack' in the corpus, and thus a model should prefer 'Mary$_a$' over other nouns for the agent role in the event 'crack'. Second, good performance also requires handling unobserved noun-verb combinations, such as unobserved agents of 'crack'. Although neither 'tiger$_a$' nor 'water$_a$' can be the agent of the action 'crack', they should not necessarily be judged as equally bad agents. One way to make these finer-grained judgements is to leverage indirect evidence, such as the similarity between 'tiger$_a$' and 'Mary$_a$', and 'water$_a$' and 'Mary$_a$'. For example, the entity 'tiger$_a$' is more similar to 'Mary$_a$' because many of the actions it performs are also performed by 'Mary$_a$'. Based on corpus statistics - which reflects such world statistics - a model should infer that 'tiger$_a$', compared to 'water$_a$', is more likely to carry out an action performed by 'Mary$_a$'. Notice that such an inference cannot rely on direct observation, but rather requires the integration of multiple observations, which I refer to as indirect evidence. In this example, a good model is expected to rank unobserved agents of 'crack' based on their similarity to 'Mary$_a$' (an observed agent of 'crack'). For example, a good model should assign a higher score to the relationship between 'tiger$_a$' and crack compared to 'walnut$_a$' and crack. In sum, good performance on the selectional preference task requires more than just tracking observed co-occurrences; instead, a model must leverage indirect evidence, namely the distributional similarity between two entities, and combine this information with the verb-noun co-occurrence, to infer plausible, but unobserved verb-noun pairs.

3.2.5 Summary of the Experimental Procedure

To summarize, I performed the following steps. First, I generated an artificial corpus. Next, I computed 216 co-occurrence matrices, one for each model trained in a minor condition (Table 3.4). For each co-occurrence matrix, I generated six similarity matrices, and used these seven matrices as the seven spatial models. To obtain the corresponding graphical models, I used the co-occurrence or similarity matrices to create undirected graphs with edges between all nodes whose matrix values were non-zero. In cases where a matrix was not symmetric, I calculated the edge weights from the sum of the matrix with its transpose, to ensure that the resultant graph is undirected. For each model, I computed the semantic relatedness score for all verb-noun pairs. This resulted in a total of $216 \times 14 = 3024$ verb-noun semantic relatedness tables, one for each model.

Next, the target relatedness scores were derived from the verb-noun co-occurrence table (Figure 3.2a,b). Then, for each model, I correlated its semantic relatedness scores for a given column in the model table with the corresponding column in the target table (Figure 3.2b,c) using the Spearman correlation. This resulted in a performance score for each verb in each model (separated by thematic role). These correlations were separated to distinguish direct and indirect word-pair stimuli. Finally, I averaged model scores across word pairs within each experiment (direct word pairs for Experiment 1, vs. indirect word pairs for Experiment 2). The final result was two performance scores for each model (direct and indirect respectively).

I repeated the above procedure 10 times, one for each corpus, and reported the average score for each model variation. All code for this paper including generation of the world and the models is available at https://github.com/UIUCLearningLanguageLab/Humans.

3.3 Hypothesis

The primary question here is which of the four model classes (spatial vs. graphical, crossed with co-occurrence vs. similarity) would be most successful at judging the semantic relatedness of previously observed (direct) and unobserved (indirect) syntagmatic relations. I hypothesized that graphical models should be more successful at inferring the relatedness of indirect pairs, while performing equally well at judging related pairs relative to their spatial counterparts. The reasoning is presented below in detail.

First, the **co-occurrence space** models should perform extremely well at predicting the semantic relatedness of observed pairs, such as Mary$_a$-crack, because that is precisely what this class of models represents. More precisely, because the target semantic relatedness scores are directly derived from co-occurrence frequency, they are extremely similar to the relatedness scores produced by the co-occurrence space model. In contrast, co-occurrence space should be less successful at inferring the relatedness of indirect word pairs. The latter is true of any distributional model that has no abstraction or inference mechanism enabling integration across multiple observations.

Second, I predicted that **co-occurrence graph** models should be equally good at predicting the semantic relatedness of direct word pairs, and, importantly, are likely to be better at predicting the semantic relatedness of indirect word pairs relative to all other models. Co-occurrence graphs directly represent syntagmatic relatedness between direct word pairs (Mary$_a$-crack) with a direct edge between the two nodes. And while indirect word pairs (like tiger$_a$ and crack) are not directly connected in the graph, they are nonetheless linked by two intermediate nodes (tiger$_a$-chase-Mary$_a$-crack). Combined with a spreading-activation procedure for computing semantic relatedness, this means that co-occurrence graphical models should be able to produce a non-zero relatedness score that is sensitive to the edge strength between each intermediate word and the surrounding network topology. That said, whether a particular co-occurrence graphical model will succeed in inferring the relatedness of indirect pairs is dependent on the particular combination of minor parameters (e.g. window size, normalization type). Therefore, I do not expect that all instances of the co-occurrence graph will achieve high performance. It is likely that most of the co-occurrence graphs with large window sizes or without normalization will not result in a topology useful for inference based on activation-spreading. Furthermore, I suspect that a co-occurrence graphical model with a window size of 1 should be able to capture both the direct and indirect relations better than any other model. The reason is that nouns and verbs always occur in adjacent positions in the training corpus (i.e., there are no intervening items in the word strings presented to our model). This does not mean this is the optimal window size for all tasks and learners, including humans; I return to this point in the discussion.

In sum, I predicted that there would be (1) a large amount of variation in performance due to the vast modeling space, (2) that both the spatial and graphical co-occurrence models perform equally well in predicting the relatedness of direct word pairs, and, importantly, (3) that the highest performance overall (Experiment 1 and 2) should be achieved by a restricted set of co-occurrence graph models.

Because the task requires inferences about syntagmatic relatedness, and because similarity models capture word substitutability (i.e., paradigmatic relations like rabbit$_a$-tiger$_a$), **similarity models** should perform overall less well than models that track co-occurrence directly. The reason is slightly different depending on whether the model is a graph or a space. As mentioned before, spatial models tend to specialize in one type of similarity, such that encoding one usually is at the cost of others. While similarity spaces might perform well at inferring the relatedness of Mary$_a$-tiger$_a$, this same ability will likely interfere with the model's success in predicting the relatedness of Mary$_a$-crack. On the other hand, while similarity graphical models are in principle able to infer different kinds of similarity, they should not be able to recover direct syntagmatic relatedness given that their computational primitive is one order of similarity above syntagmatic relatedness. Therefore, I predicted that similarity (both spatial and graphical) models will have lower performance compared to co-occurrence models when judging the relatedness of both direct and indirect word pairs.

It should be emphasized that the primary interest is not merely to identify the model that scores highest on the tasks, but rather to use performance scores as a tool for understanding the representational abilities of different types of semantic models. In particular, I was interested in which theoretical properties, and combinations thereof, enable a model to perform well. As a result, I use performance in two ways: (1) as a filter for identifying those models that warrant follow-up analyses and comparisons, and (2) to verify that the hypotheses hold up against a large amount of variation in other model parameters.

3.4 Result

3.4.1 Experiment 1

In Experiment 1, I examined the ability of models to judge the semantic plausibility of direct pairs (i.e., noun-verb pairs that occurred in the training corpus). The results are shown in Figure 3.3. As described in the Methods, there were two conditions varying data structure (spatial vs. graphical, shown in blue and orange in Figure 3.3), and seven conditions varying encoding type (shown as different violin plots in Figure 3.3). For each of these 14 combinations I ran 2160 model instances (10 randomly generated versions of the corpus for each of 216 different combinations of the minor parameters). The performance of a model averaged across all 10 corpora is shown as a line in a violin plot. The thickness of a violin at a given x coordinate indicates the number of models with a performance close to that indicated by the x coordinate. The truncation at the left and right edges of a violin indicates the minimum and maximum performance of models from that group, respectively.

The figure reveals an enormous amount of variation in performance on this task. Most models resulted in a relatively poor performance, between -0.5 and +0.5. Models that surpassed +0.5 still varied along a number of parameters, such as data structure and encoding type. Notably, many similarity models in that group used distance as a similarity metric. That said, the top-performing models were those that encoded co-occurrence, and many of them achieved perfect performance (see bottom right of Figure 3.3). As predicted, both spatial and graphical models are equally represented in this group of top performing models. The perfect and near-perfect performance of the co-occurrence models is not surprising. After all, the target semantic relatedness scores of direct word pairs are identical to co-occurrence frequency. The co-occurrence models therefore directly represent the information needed to perform well on this task, and do not require sophisticated inference.

Finally, ai noted that dimensionality reduction via SVD generally reduced model performance (e.g., comparing the violin plots labeled ‘distance’ and ‘reduced-distance’). There are several reasons for this: First, the raw co-occurrence count is the target semantic relatedness, so any departure (e.g., via dimensionality reduction, etc.) from co-occurrence necessarily results in worse performance unless there is a singular value encoding that co-occurrence, and only that singular value is used to calculate similarity. But because the training corpus is built from a simulated world in which all actions are diagnostic of the semantic relatedness structure among entities, the dimensionality reduction by SVD likely removed more signal than noise. To illustrate why models that encode similarity performed worse than those that encode co-occurrence, consider Table 3.5. In the left part of this table, I compared the best performing co-occurrence and the best performing

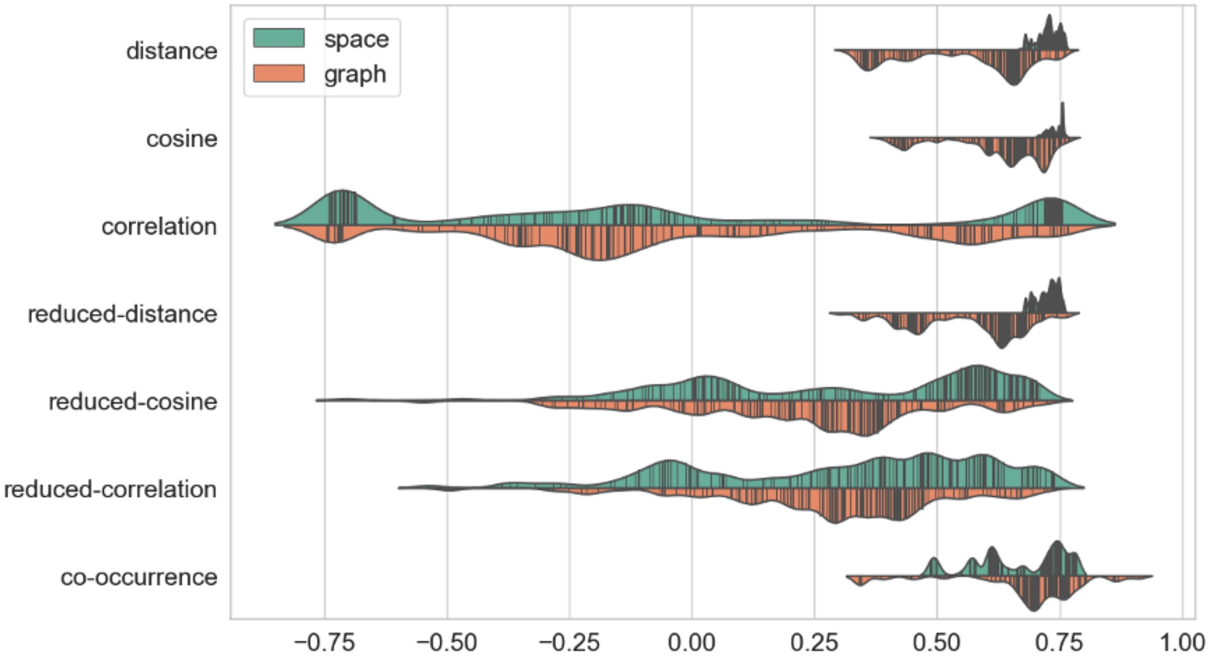


Figure 3.3: Average Model Performance on Selectional Preference Task for Directly Co-occurring Words.

similarity model on judging which nouns are better agents of the verb ‘drink’ in one of the 10 corpora. In that particular corpus, the performance of the top similarity model was 0.733, while the performance of the top co-occurrence model was 1.0. This means the top similarity model did not reproduce the correct rank-ordering of observed agents of drink according to co-occurrence frequency. The reason is that the top similarity model does not directly use co-occurrence, but, rather, must rely on word-word similarity (a transformation of co-occurrence). This transformation, as the results suggest, does not perfectly preserve co-occurrence information. It should be noted that the pattern of results presented in Table 3.5 is not specific to the verb ‘drink’, but is representative of other verbs. See Appendix E in the journal article for a list of the performance of the top models on each word pair.

3.4.2 Experiment 2

In Experiment 1, I showed that many co-occurrence models capture the semantic relatedness of word pairs that are directly observable in the training data. Next, I compared the ability of distributional models to infer the semantic plausibility of word pairs that were not directly observable in the training data. In Experiment 2, I investigated the problem by evaluating models on indirect word pairs. I was particularly interested in the spatial and graphical co-occurrence models that achieved perfect performance in Experiment 1. While each was able to represent the co-occurrence pattern of observed word pairs equally well, would they differ in their ability to infer the syntagmatic relatedness between nouns and verbs that did not directly co-occur? As stated above, I predicted that the proposed co-occurrence graph models would surpass their spatial counterparts. In Experiment 2, models were evaluated on the same 10 randomly generated corpora used in Experiment 1.

First, to get a better understanding of the overall performance across all model types, I plotted the distribution of model performance using a violin plot. The results are shown in Figure 3.4. As in Experiment 1, there is enormous variation in performance both within and between model types. In general, similarity

Table 3.5: Two case studies comparing the best similarity and best co-occurrence model in Experiment 1 (left), and between the best graphical and best spatial model in Experiment 2 (right). Left panel: the semantic relatedness between agent nouns and *drink*, in target, the best similarity and the best co-occurrence models (the rank of the noun's relatedness score in parenthesis) in Experiment 1. Right panel: semantic relatedness between agent nouns and *trap*, and the ranks of nouns, in Experiment 2.

	agent noun + 'drink'				agent noun + 'trap'		
noun	target	similarity	co-occurrence	noun	target	graph	space
tiger$_a$	293(1)	.7605(3)	.0915(1)	Kim$_a$	62(1)	.255(1)	.255(1)
wolf$_a$	289(2)	.7597(4)	.0912(2)	Mary$_a$	53(2)	.245(2)	.245(2)
Kim$_a$	81(3)	.7655(2)	.071(3)	wolf$_a$	.83(3)	.0553(3)	0(3)
Mary$_a$	74(4)	.7657(1)	.069(4)	tiger$_a$	.82(4)	.0551(4)	0(3)
squirrel$_a$	70(5)	.7586(7)	.068(5)	ibex$_a$	.769(5)	.300(6)	0(3)
boar$_a$	64(6)	.7597(4)	.0672(6)	boar$_a$	.768(6)	.304(5)	0(3)
rabbit$_a$	63(7)	.7575(9)	.0669(7)	bison$_a$	.765(7)	.0233(10)	0(3)
ibex$_a$	61(8)	.7589(6)	.0664(8)	buffalo$_a$	.764(8)	.026(7)	0(3)
buffalo$_a$	58(9)	.7574(10)	.064(9)	rabbit$_a$	.756(9)	.0238(9)	0(3)
bison$_a$	57(10)	.7583(8)	.059(10)	squirrel$_a$	.755(10)	.0244(8)	0(3)

space models perform relatively well on this task, while similarity graph models perform relatively poorly (many achieve an average performance below +0.5). Some co-occurrence space and graph models surpassed +0.5, performing better than a large proportion of other similarity models. Within that group, I found that space models clustered at approximately +0.4 and +0.75, and most graph models were more evenly distributed between +0.5 and +0.75. Furthermore, I found that there is a small minority of co-occurrence graph models that performed much better than all other models, achieving near perfect performance (the small orange hump at bottom right corner of Figure 3.4).

There are a number of other interesting patterns of performance in the results, including (1) the much higher degree of variability in the performance of graphical models in response to changes in the minor parameters relative to spatial models, (2) the extremely low variability and generally good performance for similarity space models without SVD and distance or cosine as the similarity metric, and (3) the generally (though not universally) worse performance of models using SVD compared to those that did not. But for the remainder of the chapter, I will focus on questions related to the theoretical framework and the predictions I derived from it for those models clearly standing out from the rest in terms of combined performance in Experiment 1 and 2.

Top performers

To determine whether, as predicted, the co-occurrence graph was in the top-performing models, I obtained the modeling parameters of the top 20 performers in Experiment 2. The average performance on indirect word pairs for the top 20 models is shown in Figure 3.5, and the parameters of the top 6 models are shown in Table 3.6. As predicted, I found that the top 5 models were co-occurrence graphs with a window size of 1. Because the same models also performed at ceiling in Experiment 1, these observations demonstrate that the proposed approach based on combining a graphical structure with co-occurrence data captured in small windows is most helpful for learning, representing, and inferring the syntagmatic relatedness of direct and indirect word pairs. These findings strengthen the claim that the proposed graphical co-occurrence

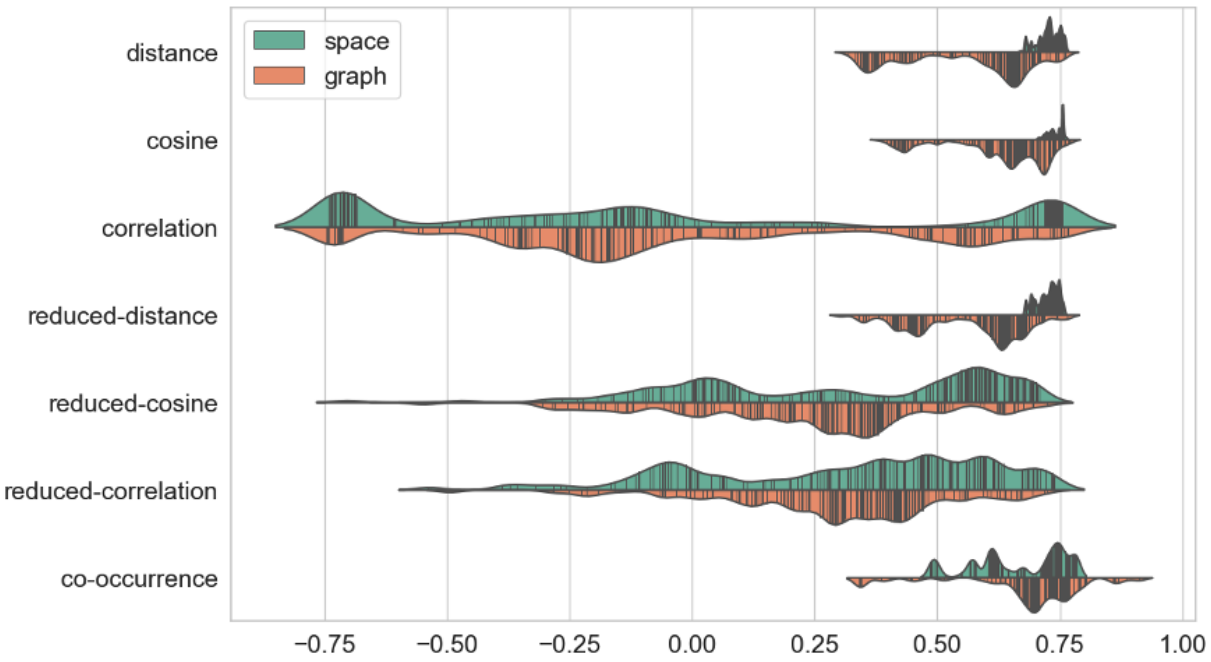


Figure 3.4: Average Model Performance in the Selectional Preference Task for Indirectly Related Words in Experiment 2.

model excels at (i) encoding multiple types of similarities simultaneously (e.g. syntagmatic and paradigmatic relatedness), and (ii) is able to infer the semantic relatedness of words that never co-occurred, by leveraging syntagmatic and paradigmatic relatedness in the same topology.

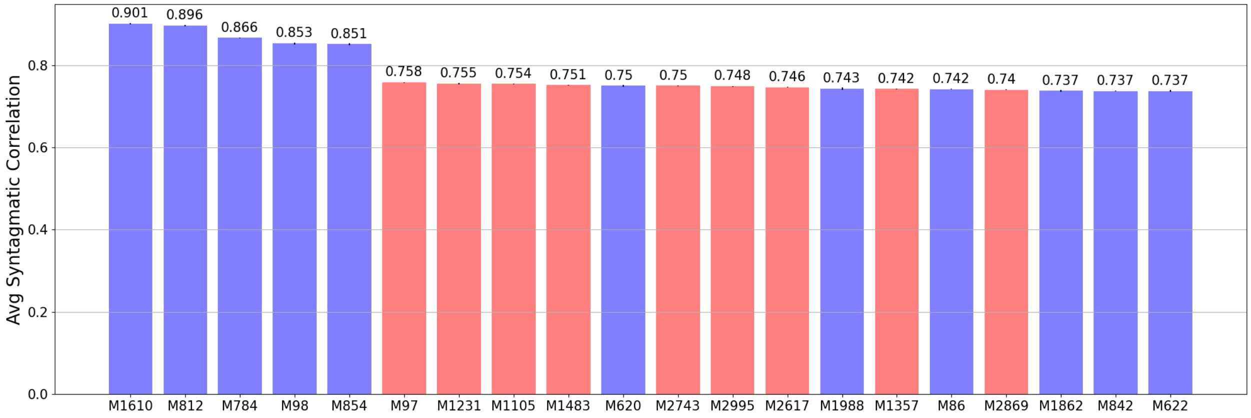


Figure 3.5: Average Performance on the Selectional Preference Task in Experiment 2 for the 20 Best Performing Models.

Lastly, these analyses revealed that, despite variation in training data (10 different runs of the simulated world, each producing a distinct corpus), and variation in minor parameters, the top performers are extremely consistent in terms of model type and performance. Not only are the standard deviations of the Spearman rank correlation miniscule for each of the best performing models, but they also differ little in their parameter setting and overall performance.

Table 3.6: Average performance and parameter values for the top six models.

score rank	mean score	period	bound -ary	window size	window weight	window type	normal-ization	encoding type	data structure
1	0.901	no	yes	1	flat/linear	summed	log	co-occur	graph
2	0.896	yes	no	1	flat/linear	backward	log	co-occur	graph
3	0.866	yes	no	1	flat/linear	backward	ppmi	co-occur	graph
4	0.853	yes	yes	1	flat/linear	summed	log	co-occur	graph
5	0.851	yes	no	1	flat/linear	summed	log/non	co-occur	graph
6	0.758	yes/no	yes	1/2/7	flat/linear	summed	log/non	co-occur	space

Targeted Follow-up model comparisons

What enabled the top models to perform well on the selectional preference task for indirect items? To answer this question, I compared the top co-occurrence graph model to the top co-occurrence space model. There are two reasons why such a comparison is useful. First, these two models are the best graph and space models overall. See Appendix F in the journal article for the best performers within each level of similarity. Second, these two models achieved perfect performance in Experiment 1.

To compare them, I analyzed their ability to infer the plausibility of agents for the verb 'trap'. In terms of overall performance classifying plausible agents of 'trap', the top graphical model achieved a Spearman rank correlation of 0.901 between its predicted semantic relatedness scores and the target semantic relatedness scores. In contrast, the top spatial model scored 0.758. This example is representative of differences in the performance of verbs other than trap.

Looking specifically at which nouns were judged to be plausible agents (shown in Table 3.5, right), both models predicted that the best agents of the verb trap were nouns in the category HUMAN, which co-occurred with the verb in agent position in the corpus. However, the graphical model correctly judged nouns in the categories CARNIVORE, M_HERB, S_HERB, and L_HERB to be decreasingly less plausible as an agent for 'trap'. The spatial model, in contrast, did not differentiate between agent nouns in these categories. Instead, the spatial model assigned all agents that are not in the category HUMAN a relatedness of zero.

This maladaptive behavior of the co-occurrence space model can be explained in terms of how it derives semantic relatedness scores from co-occurrence data: If relatedness is derived directly from co-occurrence frequency, and co-occurrence frequency is zero, then the resulting semantic relatedness must also be zero. This cannot be remedied by tuning minor parameters, such as pre-processing or normalization. The presence of these zeros makes it impossible for co-occurrence space models to directly make fine-grained distinctions between unobserved word pairs (e.g., is bison or wolf a better agent of trap?).

The reason for the relative success of the graphical model is that, given enough time steps, the spreading-activation algorithm produces graded relatedness scores no matter how distantly connected two nodes are. Although the node that corresponds to 'trap' is not directly connected to nodes representing entities that are not of type HUMAN (i.e potential agents of trap), the spreading activation procedure links trap and potential agents via one or more indirect connections. For example, 'trap' and 'wolf$_a$' are connected indirectly via the nodes 'catch' and 'chase', (Figure 3.6a). By leveraging these indirect connections, the spreading activation procedure activates 'wolf$_a$' after three time steps, and the result is a non-zero semantic relatedness between 'trap' and 'wolf$_a$' (Figure 3.6b). This can be verified by inspecting the middle column of Table 3.5. The graphical model correctly ranks members of CARNIVORE above HERBIVORE as agents of 'trap'. The reason is that members of HUMAN are the most frequent (and only observed) agents of 'trap', and the graphical model considers entities of type CARNIVORE to be more semantically similar to entities of type HUMAN than HERBIVORE and HUMAN. This can be further explained in terms of the event semantics of the simulated world: Members of CARNIVORE (e.g. wolf$_a$) perform many of the same actions performed by members of HUMAN, and more so than HERBIVORE.

Correspondingly, members of CARNIVORE co-occur with more verbs (e.g., 'catch' and 'chase') in the corpus that are shared by members of HUMAN relative to HERBIVORE. In turn, the nodes corresponding to members of CARNIVORE have a larger number of shorter — and therefore stronger — paths to nodes referring to members of HUMAN in the network. The same line of reasoning can be used to explain why the graphical model treats members of L_HERB and S_HERB as the least plausible agent of 'trap'; the paths between nodes corresponding to members of L_HERB and S_HERB and nodes corresponding to members of HUMAN are fewer in number and weaker in strength.

To summarize, I have shown that graphical models tend to differentiate the plausibility of unobserved arguments of verbs, while (co-occurrence) spatial models do not. To produce graded semantic preferences, graphical models can compensate for the lack of information about unobserved co-occurrences by leveraging indirect connections via spreading activation to distantly connected nodes.

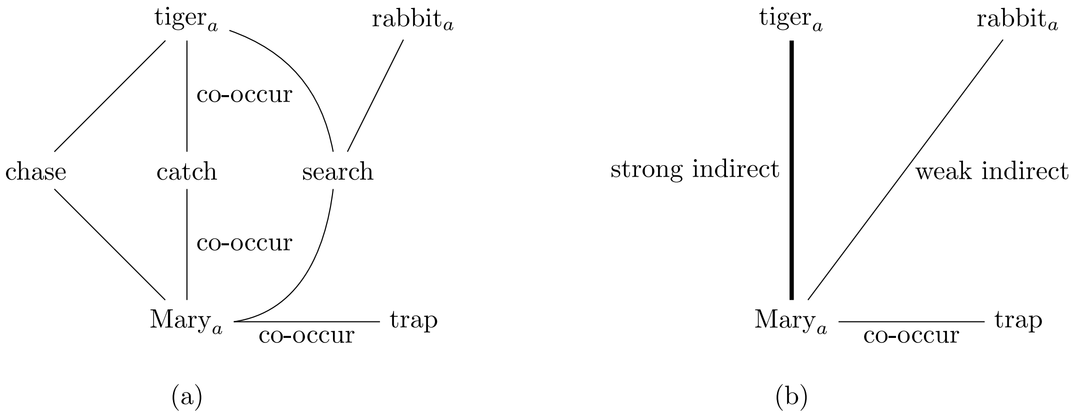


Figure 3.6: An illustration of a Graphical Model Inferring on Unseen Agent-Verb Pair. The graphical model can infer that 'tiger$_a$' is a likely agent of 'trap' despite having never observed the word 'tiger$_a$' as the agent of 'trap' in the corpus it was trained on. (a) The nodes 'tiger$_a$' and 'Mary$_a$' are connected indirectly via three multi-edge paths that involve the nodes 'chase', 'catch' and 'search'. While 'rabbit$_a$' and 'Mary$_a$' are connected only by one path. (b) The multiple paths lead to a strong indirect paradigmatic relationship between 'tiger$_a$' and 'trap', while the single path leads to a weak indirect relation to the same verb.

3.5 Discussion

The primary aim of this chapter is to compare the ability of different distributional semantic models to infer the semantic plausibility of observed and unobserved verb-noun pairs. Models were first trained on artificial corpora grounded in a simulated world with hierarchical event structures and realistic agent-environment contingencies, and then tested on a selectional preference task in two experiments. To succeed in both experiments, a model needed to encode and use fine-grained distinctions in semantic plausibility based on observed co-occurrence (i.e. direct word pairs in Experiment 1) and shared co-occurrence (i.e. indirect word pairs in Experiment 2). During evaluation, I derived semantic relatedness scores for specific verb-noun pairs from each model, and compared them against relatedness scores derived from the corpus a model was trained on. I focused on comparing models that use distances in a vector space as a measure of relatedness, and models that use spreading activation in a graph built from the same co-occurrence data. I was also interested in the relative performance of these models as a function of whether semantic relatedness was defined in terms of word co-occurrence or word similarity. I sampled models systematically from a large space of minor parameters to better understand the contribution of individual modeling choices on downstream model performance.

The findings can be briefly summarized as follows. First, while both graphical and spatial models performed, on average, equally well in both experiments, the best graphical models performed better than the best spatial models on indirect word pairs (Experiment 2, Table 3.5). Second, I observed that models that used co-occurrence frequency to define its dimensions (for spatial models) or edges (for graphical models) generally produced higher and more consistent scores than those that used similarity scores derived from that co-occurrence data.

To better understand the results, I conducted targeted follow-up comparisons of the best performing models. I found that the semantic plausibility judgements produced by the best spatial model were on par with those produced by graphical models in Experiment 1 (for directly related pairs) but not in Experiment 2 (for pairs that did not directly co-occur in sentences in the corpus). Further, I found that encoding co-occurrence rather than similarity was advantageous for generalizing to indirect word pairs. The reason that the performance of spatial models lagged behind was that they assigned a semantic relatedness score of zero to word pairs that did not occur in the corpus they were trained on. In contrast, the best graphical model was able to compensate for the lack of observed co-occurrence by deriving the plausibility of an unobserved verb-noun pair via activation spreading along multiple indirect paths that link the nodes corresponding to the noun and verb in the network. The spreading activation procedure for obtaining semantic relatedness scores proved crucial, as it enabled graphical models to assign non-zero, graded semantic relatedness scores to unobserved word pairs. This means that indirect paths connecting two non-adjacent nodes appear to enable strong inferences about their semantic relatedness. As a summary, I argue that its success critically depends on three components: 1) the graphical data structure, 2) the spreading-activation measure for computing semantic relatedness on the graph, and 3) the encoding of adjacent co-occurrence. It was the combination of these three factors that enabled the co-occurrence graphical model to succeed in the experiments.

To better understand the representational capabilities of the graphical and spatial model, and how the two representational forms are related to each other, I contextualize our findings within a principled framework that formulates the formal similarities and differences between spatial and graphical models. It is suggested that semantic inference in the proposed graphical model, the co-occurrence graph, approximates a traversal of a series of increasingly higher-order similarity spaces. This account is an attempt to pinpoint the fundamental difference between a graphical and a canonical spatial representation of distributional linguistic data. In Chapter 4, I first set up a semi-formal formulation to show the relation between graphical and spatial models, and use it for a more in-depth explanation of he success of the co-occurrence graph relative to its alternatives. Then, I provide a fully formal description on the structural and processing equivalence between spatial and graphical models, and show the potential implications of this equivalence in modeling knowledge. Before delving into the topic, I first mention a critical limitation of the co-occurrence graphical model presented in this chapter. While the co-occurrence graph can handle pairwise semantic relations presented in the corpus, it may struggle in learning multi-way (higher order) lexical dependencies. I briefly discuss the limitation here, and present thorough investigations on the topic in Chapter 5 and Chapter 6.

3.5.1 Higher-Order Dependencies

The semantic content and structure of our artificial language is very simple compared to real world events and languages. Real world languages can have multiple thematic roles (theme, experiencer, instrument, location, time, cause, purpose, etc.) that have higher-order dependencies than the simple pairwise relations in our language (Elman, 2009; Hare, McRae, & Elman, 2003; McRae et al., 2005). As an example, in the following sentences:

1a. Mary carved the stone with the chisel.
1b. Mary carved the turkey with the knife.
1c. Mary roasted the turkey with the oven.

A comparison between sentences 1a and 1b demonstrates that the choice of the instrument is not dependent solely on the verb. The patient matters: What is being carved is important. On the other hand, the patient noun itself is not deciding either: the instrument varies according to the verb. Therefore, the choice on 'knife' as the instrument in sentence 1b depends on the verb and the patient noun collectively. In other words, the dependency is on the phrase 'carve turkey', a complex expression. To study this type of problem, we need to first enrich the sentence structure by including more thematic role types and higher-level dependencies between multiple semantic components. For example, a future direction could incorporate location, such that tigers chase rabbits in the forest, while wolves chase rabbits on the prairie. Additionally, the artificial language may need to include prepositions to support additional nouns in different thematic roles (e.g. 'with' in 'with the chisel').

To account for the tendency of the phrase 'carve turkey' to select the word knife but not chisel and oven, it is likely that some representation of the compositional structure must be available. However, this is not implemented in most semantic models — including the Co-occurrence Graph model. It is important to note that making quantitative inferences about unobserved combinations of words (cities-sleep) likely requires additional innovation beyond that of the co-occurrence graphs. Many researchers in psycholinguistics have proposed that compositionality, the ability to systematically combine small meaning components to form novel meanings, is crucial for understanding human language acquisition and use (Baroni, Bernardi, & Zamparelli, 2014; Fodor & Pylyshyn, 1988; Gershman & Tenenbaum, 2015; Mitchell & Lapata, 2010). For example, inferring the plausibility of 'cars sleep', can be explained in terms of compositional structures and processes that govern how words are combined. However, the co-occurrence graph at most provides a way to construct and measure the semantic relatedness between composable word pairs, but does not have an explicit mechanism for representing composed units, or knowledge about how composed units relate to their parts. These issues lend themselves to the graphical formulation of semantic knowledge representation, and will be discussed in Chapter 5.

While the co-occurrence graph performs well on the artificial sentences, more complex sentences, such as 'Mary drinks juice slowly', are likely to cause difficulty for a network trained with a window size of 1. The adverb 'slowly' modifies 'drinks', which is not adjacent to slowly. The graphical co-occurrence model with window size of 1 cannot represent this long-distance dependency directly, and instead has to represent it using two edges, one that links the adverb to the patient, and another that links the patient to the verb. This example is a symptom of a more general problem, namely that a sequential chain of co-occurrences cannot capture the hierarchical syntactic and semantic dependencies that exist in natural language. For convenience, I will also refer to the Co-occurrence Graph models as Linear Order Network (LON), to emphasize the model's representation of linguistic expressions by a linear chain of the words.

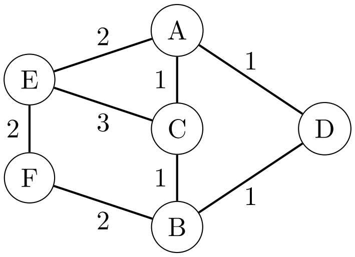
To upgrade the co-occurrence graph model in order to address higher order relations and representation of compositional structure, sentences might need to be translated into more syntactically structured forms, before they are joined into network. The motivation gives rise to the Constituent Tree Network (CTN), in which the graphical structure is built from parsed trees. As a result, compositional structures are explicitly encoded, and therefore it is possible to capture the plausibility of sentences with higher order dependencies (e.g., 'Mary carves a turkey with a chisel'). In Chapter 5, we will see how CTN, alongside with the generalized spreading activation measure that computes phrasal semantic relatedness, manage to represent and generalize higher order (phrasal) selectional preferences.

Chapter 4

Graph and Space

In chapter 3, I empirically tested the capability of graphical and spatial models on learning and generalizing lexical relations observed in an linguistic corpus. The result suggests encoding co-occurrence information work the best and I provide a descriptive explanation for the result. In this chapter, I attempt to formalize the relationship between graphical and vector-space representations. I show that a mathematical equivalence exists between certain evaluation processes on the two types of data structures. In terms of structure, graphs and vector spaces can be linked by the matrices used to define the graph. In a typical semantic network, an edge is formed between node i and j, if the corresponding entry a_{ij} in the adjacency matrix A is non-zero (with its value encoded as edge weight). In this way, the weights of edges attaching to a node in the network is equivalent to the row of the node in the adjacency matrix, and therefore can be considered as the vector representation of the concept node (See Figure 4.1). Based on this structural relationship, I will show that, given a certain computational operationalization, the 'intersecting' spreading-activation on the graph (similar to that proposed by Collins and Loftus (1975)) between two concept nodes is mathematically equivalent to the cosine similarity of the two concepts' original or 'higher-order' vector representations. I show that the inclusion of the 'higher order' information leads to better representation of semantic structure, and propose that this isomorphism between graphical and vector models has implications for theories and models of the nature of semantic representation.

The equivalence we demonstrate between graphical and vector space models is theoretically significant in two ways. First, it shows that - contrary to long-standing belief - graphical spreading activation models and vector space models are not qualitatively incomparable models, and that it may be possible to relate them in a number of ways. Second, the equivalence between graph and vector space representations may inspire us to transfer processes and analysis techniques that are typically used on one data structure to the other, resulting in more powerful modeling capabilities. These results may have profound implications on the nature of meaning representation and machine learning practice, as we will discuss.



(a)

$$\begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline A & 0 & 0 & 1 & 1 & 2 & 0 \\ B & 0 & 0 & 1 & 1 & 0 & 2 \\ C & 1 & 1 & 0 & 0 & 3 & 0 \\ D & 1 & 1 & 0 & 0 & 0 & 0 \\ E & 2 & 0 & 3 & 0 & 0 & 2 \\ F & 0 & 2 & 0 & 0 & 2 & 0 \end{array}$$

(b)

Figure 4.1: A network and the equivalent adjacency matrix: An edge from i to j is formed (with edge weight a_{ij}) if the corresponding entry a_{ij} in the adjacency matrix is non-zero. (a) Weighted graph (b) Adjacency matrix of the network in (a). Each row (or column) in the adjacency matrix can be considered as a vector representation of the concept.

4.1 A semi-formal explanation on the success of co-occurrence graph

4.1.1 Many Higher-Order Embedding Spaces

In order to appreciate how graphical and spatial models differ as representational substrates for semantic relatedness computations, some definitions are needed. Given a row-normalized (illustrated in 2.4 b) word-by-word co-occurrence matrix C, each row corresponds to a vector representation of a target word defined as a set of co-occurrence probabilities. These word vectors reside in a multi-dimensional space, where each dimension is the (normalized) co-occurrence with a word that has appeared in the target word's context. As discussed previously, the pairwise comparison of all word vectors can be used to construct a word-to-word similarity matrix (see Figure 3.1). More formally, the similarity matrix is approximately the product of the co-occurrence matrix C and its transpose C^T. This is the process used to generate the similarity space models in Chapter 3. I refer to the semantic vector space spanned by the co-occurrence vectors as a space with order $(1, 0)$ (will clarify the meaning of the two indices in a bit, and more extensively in the next section), and the vector space spanned by the row vectors of the similarity matrix CC^T as a space with order $(1, 1)$. Vector entries in the latter space are the similarities between the vectors in the order $(1, 0)$ space. In Chapter 3, the co-occurrence and similarity space models are different variants of these two vector spaces.

Note that these two semantic vector spaces represent qualitatively different contents. Whereas entries in order $(1, 0)$ spaces correspond to (normalized) co-occurrence frequencies, entries in order $(1, 1)$ space correspond to similarities between two vectors in the order $(1, 0)$ space. That is, in a similarity space (i.e. an order $(1, 1)$ space), two words are similar in terms of their pattern of co-occurrence with words in their context. While an order $(1, 0)$ space may capture direct syntagmatic relationships like Mary$_a$-trap, an order $(1, 1)$ space captures the paradigmatic relationship between 'Mary$_a$' and 'tiger$_a$', which share many verbs. However, as shown in the results, both types of spatial models struggle when making inferences about syntagmatically related word pairs such as tiger$_a$-trap that do not directly occur in the training corpus. While the verb 'trap' does not directly co-occur with 'tiger$_a$', it does co-occur with 'Mary$_a$', a word that is distributionally similar to 'tiger$_a$'. This relationship cannot be captured by relatedness found in either an order $(1, 0)$ or order $(1, 1)$ space alone. Instead, as mentioned previously, a stepwise procedure to compute this indirect relatedness is needed: One that considers both the paradigmatic relationship tiger$_a$-Mary$_a$, and the syntagmatic relationship Mary$_a$-trap. One way to do this is to input vector representations from both the order $(1, 0)$ and order $(1, 1)$ spaces to the computation of relatedness. First, we can leverage the fact that the order $(1, 1)$ vector that corresponds to 'tiger$_a$' encodes the similarities between it and other words in that same space, e.g., in order to link 'tiger$_a$' and 'Mary$_a$'. Second, we can quantify the strength of the relationship between 'Mary$_a$' and 'trap' by inspecting the order $(1, 0)$ vector representation of 'Mary$_a$'. If the relationship in both steps is found to be strong, this indicates that 'Mary$_a$' and 'trap' are indirectly related.

More formally, if we have not observed word X co-occurring with word A, but we have observed Y and Z co-occurring with A, then it can be inferred that X should co-occur with A to the extent that it is similar to Y and Z. The relatedness of X and A can be estimated as the dot product of the order $(1, 1)$ vector that represents X (the row vector for X in the similarity matrix) and the order $(1, 0)$ vector that represents A (the row vector for A in the co-occurrence matrix). This brings us to the concept of a 'higher order' vector space. This process of creating a higher-order space from a lower order space can be continued to produce increasingly higher order spaces. Returning to the example of inferring the relatedness between X and A, we can take the order 0 vector that represents A in the co-occurrence matrix C and compute its dot product with the vector that represents X in the similarity matrix CC^T. The result can be considered a measure of the indirect relatedness between X and A.

Generalizing from vectors to vector spaces, if we take the dot product of all rows in the order $(1, 0)$ matrix C and the order $(1, 1)$ matrix CC^T, the result is the higher order similarity matrix $C(CC^T)^T$, or simply C^2C^T. In plain English, this operation involves transposing the order $(1, 1)$ matrix and then left multiplying it by the order $(1, 0)$ matrix C. In the resulting matrix, the entry at (i, j) corresponds to the dot product between the order $(1, 1)$ vector for word i and the order $(1, 0)$ vector for word j. Importantly, this process can be repeated to generate increasingly higher order vector spaces. It involves taking an existing matrix of some arbitrary order, and multiplying it by the order $(1, 0)$ space from which it was derived. Starting with a matrix of order $(1, 0)$, namely, CC^T, the process of deriving higher order spaces can be denoted by the sequence CC^T, C^2C^T, $C^2(C^T)^2$, $C^3(C^T)^2$, $\ldots$, (with order $(1, 1), (2, 1), (2, 2), (3, 2)$... in which the two exponents are incremented in an alternating fashion. Replacing the first and second exponent with the variables m and n, respectively, we can denote a space of any order (m, n) using the form $C^m(CT)^n$. An entry at (i, j) in this generalized vector space can be considered the semantic similarity between word i and j at some abstraction level determined by m and n: It represents some kind of similarity between the order $(m, 0)$ vector of word i and the order $(n, 0)$ vector of word j.

4.1.2 Spreading-Activation as Traversal of Increasingly Abstract Embedding Spaces

We are left with the question of how to interpret the entries in a generalized higher order vector space of the form $C^m(C^T)^n$. To interpret these higher order embedding spaces, the formalism of spreading activation in networks can be helpful. Indeed, it can be showed that there is a formal equivalence between the step-wise derivation of higher order spaces, and the process of spreading-activation unfolding across time steps in a graph. As an analogy to the random walk (De Deyne et al., 2016), the matrix C^m describes the activation state of the graph after m steps of activation-spreading, and the entry (i, j) of C^m is the activation arriving at node j from node i via all paths of length m. Correspondingly, the entry (i, j) in the generalized embedding matrix $C^m(C^T)^n$ is the amount of activation that 'intersects' at some intermediate locations in the graph after spreading from node i for m time steps and from node j for n time steps (See Figure 4.2 for an illustration). It is analogous to the probability of two random walks initiated from i and j 'meeting' (intersecting) with each

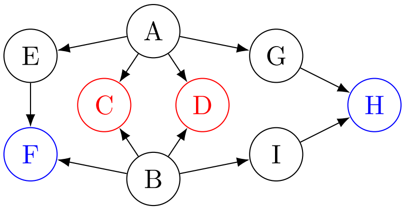


Figure 4.2: Intersecting activation on a network: Activation originated in two source nodes A and B 'meets' on intermediate nodes (in colors). Cosine similarity can be equaled to the intersecting activation on nodes with distance of one to both sources (red). Generalized similarity (cosine similarity between 'higher order' vector representations) can be equaled to intersecting activation traversing longer paths (in blue).

other on the nodes, which is exactly the entry (i, j) in the matrix $C^m(C^T)^n$. In this sense, the order (m, n) can be considered as the steps taken from node i, j. This amount of intersecting activation (random walk meeting probability) is equivalent to the similarities between the higher-order vector spaces discussed above. While this computation has been discussed in previous work (De Deyne et al., 2016) from a methodological perspective, here, I explicitly note that this computation describes an equivalence between representing relatedness in a graph and in higher order vector spaces. A more detailed formulation of this equivalence is provided in section 4.2.

In light of this formal equivalence, I argue that there is a close correspondence between computing relatedness via spreading-activation in a graph and computing relatedness/similarity in vector spaces. In particular, I claim that spreading-activation in the proposed co-occurrence graph can be considered as a step-wise traversal across vector spaces of different levels of abstraction (i.e. order $(1, 0)$, order $(1, 1)$, etc.) in a single topology. The benefit of the co-occurrence graph model, and DG in general, is that the same topology can be used for computing multiple orders of semantic relatedness without needing to determine when to switch to a space at a different level of abstraction. In sum, the spreading-activation procedure for computing semantic relatedness in a graph can be considered as a traversal over successively higher-order vector spaces.

It should be noted that in Chapter 3, I measured activation originating at a single source node and arriving at a single target node only. That is, I set $n = 0$ in all our experiments. In this special case, relatedness is measured as the amount of activation arriving at the target node, instead of at intermediate nodes other than the target node. This method proved sufficient for the co-occurrence graph to perform well in the experiments. This makes sense considering that the two nodes that represent an indirect word pair tested in Experiment 2 were typically no more than 3 edges (i.e. time steps) away from each other in the graph. The traversal of activation across the first two edges corresponds to the computation of an order $(1, 1)$ vector representation of the source word, and the traversal of the third edge corresponds to the computation of the similarity between the order $(1, 1)$ vector representation of the source word and the order $(1, 0)$ representation of the target word.

The equivalence between the co-occurrence graph with a series of increasingly higher order vector spaces has implications for the observed differences in the behaviors of the models examined in Chapter 3. As mentioned earlier, spatial models of the form C and CC^T only encode co-occurrence or similarity but not both. Further, the lexical relatedness derived from these models are restricted to orders $(1, 0)$ and $(1, 1)$, which do not encode information about the indirect syntagmatic relationship between 'tiger$_a$' and 'trap' that needs an order $(2, 1)$ similarity. However, a graphical model equipped with spreading activation can flexibly access lexical relatedness at multiple different levels of abstraction. For instance, the relatedness of the pair Mary$_a$-trap can be quantified by the activation that reaches 'trap' directly from Mary$_a$, and the relatedness of the indirect pair tiger$_a$-trap can be inferred by the amount of activation that reaches 'trap' after multiple time steps. Each time step of spreading-activation, therefore, corresponds to a traversal to a higher order vector space. In contrast to spatial models, the computation of these different kinds of similarities in the graph do not require moving back and forth between different kinds of representational structures, as is the case for spatial models. The ability to flexibly move between levels of abstraction should be especially useful when tasked with inferring the relatedness of words that are related as a result of multiple different kinds of distributional semantic patterns. For instance, I showed that a higher-order space is needed to capture the indirect relationship between tiger$_a$ and 'trap' in the artificial corpus, and that the co-occurrence graph was able to infer the relatedness of such word pairs. The theoretical analysis suggests that the model was able to do so based on its ability to consult multiple orders of similarity as part of the same inference procedure (i.e. spreading-activation).

By turning the attention to the study of the relatedness between distant nodes in a graph, the work here is effectively studying the contribution of higher-order similarity on semantic inference. This is an important step, given that, historically, the use of strongly related words has dominated the study of the organization of semantic memory. An unintended consequence of this focus on lower-order relations has likely contributed to the success and proliferation of vector space models in accounting for psycholinguistic data. However, Chapter 3 demonstrates that a graphical approach can be equally successful, and, further, may have advantages of inference on pairs of words or concepts that are less directly related. This formal analysis is in concordance with previous empirical work which has demonstrated the psychological reality of longer paths in terms of their ability to account for additional variance in human semantic judgments beyond relatedness computed on shorter paths (De Deyne et al., 2016; Rotaru et al., 2018).

4.2 Formal Equivalence Between Spreading-activation and Cosine Similarity

In this section, I sketchily show that with a particular formulation of the spreading-activation, the process is equivalent to the cosine similarity of the vector representation of the concept nodes.

4.2.1 Immediate Activation and Cosine Similarity

I define spreading activation in a way that slightly differs from the canonical random-walk type of definition. The adjustment is made to correspond to cosine similarity of vectors. When a node is activated, it spreads activation to each of its neighbors, with the amount of activation spread determined by the locally normalized weight on the edge linking to the neighbor (the local weight divided by the 'sum' of all outgoing weights). A numeric example is illustrated in Figure 4.3. Here we consider the adjacency matrix W, and define the normalized weight $\widetilde{\omega_{ij}}$:

$$\widetilde{\omega_{ij}} = \frac{\omega_{ij}}{\|\mathbf{r_i}\|_2} \tag{4.1}$$

where ω_{ij} is the (i, j) entry in W as well as the edge weight from i to its neighbor j, and

$$\mathbf{r_i} = (\omega_{i1}, \omega_{i2}, ...\omega_{in}) \tag{4.2}$$

is the i-th row of the n-dimensional adjacency matrix, which is also the vector embedding of i (illustrated

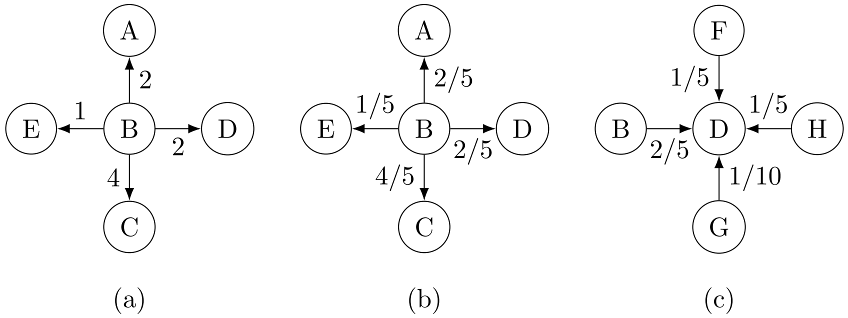


Figure 4.3: Local normalization of edge weights: (a) Original edge weights in the network. (b) Edge weights after normalization based on L_2 norm. See equation (4.3) for the computation. (c) Node activated by locally normalized neighboring nodes.

in Figure 4.1). We should note here that the normalization in Equation 4.1 is by L_2 norm, instead of the common L_1 norm (sum of the weights). Given the current activation level a_i on node i, the activation spread from i to the neighbor j is defined as:

$$a_{ij} = a_i \widetilde{\omega_{ij}} \tag{4.3}$$

In this way, if two arbitrary nodes i, j are activated with a unit activation 1, the activation may intersect immediately on nodes to which both are directly connected (such as nodes C and D in Figure 4.2). I multiply the activation produced by the two sources at each node k where activation intersects, and sum the resulting product. I denote this 'intersecting activation' as $\text{sim}^{1,1}(i,j)$:

$$\text{sim}^{1,1}(i,j) = \sum_k a_{ik} a_{jk} \tag{4.4}$$

The superscript '$1,1$' denotes that the activation intersects on nodes with distance 1 from both activation sources. If activation intersects, the activation is included in the "intersecting activation" sum, otherwise, $a_{ik}a_{jk} = 0$. Plugging in equations (4.1) and (4.3) into (4.4), we have an equation for similarity spanning nodes that have intersecting activation at time step 1:

$$\text{sim}^{1,1}(i,j) = \sum_k \frac{\omega_{ik}\omega_{jk}}{\|\mathbf{r_i}\|_2 \|\mathbf{r_j}\|_2} \tag{4.5}$$

with the source activation $a_i = a_j = 1$. In this case,the right hand side of equation (4.5) is exactly the definition of $\cos(\mathbf{r_i}, \mathbf{r_j})$, and therefore we have $\text{sim}^{1,1}(i,j) = \cos(\mathbf{r_i}, \mathbf{r_j})$. In other words, the cosine similarity between the vector of two concepts is the sum of intersecting activation on the intermediate nodes defined by equation (4.3) and (4.4).

4.2.2 General Activation and Higher Order Similarity

Cosine similarity is one of the most applied measures for evaluating the relationship between vector representations of meaning (Landauer & Dumais, 1997). As I showed, this is equivalent to the 'after 1 time step' intersecting activation in the corresponding network. Now we show this can be generalized to activation spreading over any number of time steps, with the intersecting activation on longer paths being mathematically equivalent to cosine similarities calculated on 'higher-order' vectors derived from the original vector space.

Building of the definition of spreading activation in equation (4.3), I define the activation to a target node j from its neighbors as the following:

$$a_j = \sqrt{\sum_k a_{kj}^2} \tag{4.6}$$

An illustration can be found in Figure 4.3 c. In Figure 4.3 b, node B spread 2/5 of its activation to node D after local normalization on B. Switching to node D, it gets activation from all its neighbors (B,F,G,H) proportional to the respective normalized weights. According to the normal random-walk type definition, the activation received by D should just be the sum of the individual activation. Assume the activation on all neighboring nodes are 1, then the activation at D should be .9. However, we use (4.6), that is, we use L_2-norm of the activation vector $(a_{1j}, a_{2j}, ..., a_{nj})$ instead of simple summation to make the computation compatible with the cosine similarity measure. As in the example, we have .5 for the activation received by D, according to (4.6).

Combining (4.3) and (4.6), I define the general activation-spreading process on a network as following: A set of nodes are activated at the initial moment, and the activation propagates to the neighboring nodes at discrete time steps. In each step, an activated node sends out all its activation to its neighbors following (4.3), and every node receives activation from its neighbors following (4.6). If node i is activated with unit activation 1 at t_0, the activation spread from i to an arbitrary node j after 1 step is $\widetilde{\omega_{ij}}$, and more generally after m steps $(m > 1)$ is denoted as $a^{(m)}(i,j)$, which can be computed by:

$$a^{(m)}(i,j) = \sqrt{\sum_k (a^{(m-1)}(i,k)\widetilde{\omega_{kj}})^2} \tag{4.7}$$

In other words, the activation received at j from i at step m is the L_2-norm summation of the portion of activation diffused to other nodes in the network (from node i) at step $m-1$, that reaches j one step later. This leads to

$$\mathbf{r_i}^{(m)} = (a^{(m)}(i,1), a^{(m)}(i,2), ...a^{(m)}(i,n)) \tag{4.8}$$

as the vector representing activation spread from i to all nodes in the network after m steps. When $m = 1$, it is easy to verify that $\mathbf{r_i}^{(1)}$ is the normalized vector representation of node i. For larger m, $\mathbf{r_i}^{(m)}$ can be considered as a higher-order vector which takes m-step activation traversed in the network as its vector entries. With Equation (4.7), it is easy to show that:

$$\cos(\mathbf{r_i}^{(m_1)}, \mathbf{r_j}^{(m_2)}) = \sum_k a^{(m_1)}(i,k)a^{(m_2)}(j,k) \tag{4.9}$$

in which the left hand side is the cosine similarity between the higher order vector representations of concept i and concept j, and the right hand side is the generalized intersecting activation as of Equation (4.4). Instead of meeting after one step as in (4.4), the intersecting activation has traversed m_1 and m_2 steps from i and j respectively. Therefore, Equation (4.9) shows that the generalized intersecting activation from two concept nodes in the network is mathematically equivalent to the cosine similarity between the 'higher-order' vector that encode information on longer paths. Finally, I denote the general cosine similarity as $\text{sim}^{m_1,m_2}(i,j)$:

$$\text{sim}^{m_1,m_2}(i,j) = \cos(\mathbf{r_i}^{(m_1)}, \mathbf{r_j}^{(m_2)}) \tag{4.10}$$

We need to emphasize that the rigorous equivalence between the cosine similarity and intersecting activation can only be guaranteed with the L_2 normalization in sending (4.3) and receiving (4.6) activation. The adjustment guarantees that $\|\mathbf{r_i}^{(m)}\|_2 = 1$, for all i, m. This leads to:

$$\sum_k a^{(m_1)}(i,k)a^{(m_2)}(j,k) = \frac{\sum_k a^{(m_1)}(i,k)a^{(m_2)}(j,k)}{\|\mathbf{r_i}^{(m_1)}\|_2 \|\mathbf{r_j}^{(m_2)}\|_2} = \cos(\mathbf{r_i}^{(m_1)}, \mathbf{r_j}^{(m_2)}) \tag{4.11}$$

However, without the adjustment, the canonical random-walk type spreading activation normalize the local weight with L_1 (probability) norm, such that $\|\mathbf{r_i}^{(m)}\|_1 = 1$, but not necessarily $\|\mathbf{r_i}^{(m)}\|_2 = 1$. As a result, with the canonical random-walk spreading activation, we have:

$$\sum_k a^{(m_1)}(i,k)a^{(m_2)}(j,k) = \frac{\sum_k a^{(m_1)}(i,k)a^{(m_2)}(j,k)}{\|\mathbf{r_i}^{(m_1)}\|_1 \|\mathbf{r_j}^{(m_2)}\|_1} = \frac{\|\mathbf{r_i}^{(m_1)}\|_2 \|\mathbf{r_j}^{(m_2)}\|_2}{\|\mathbf{r_i}^{(m_1)}\|_1 \|\mathbf{r_j}^{(m_2)}\|_1} \cos(\mathbf{r_i}^{(m_1)}, \mathbf{r_j}^{(m_2)}) \tag{4.12}$$

That is, the sum of activation differs from the cosine similarity of the higher order representations by a multiplier, which is the ratio between L_1 and L_2 norms of the higher order representations. As this multiplier is a function of the higher order representations, in general, we do not have a concise equivalence or correspondence between the graphical spreading activation and the spatial cosine similarity. Empirically, when the order (m_1, m_2) is small, as the case in Chapter 3 and related studies (De Deyne et al., 2016; Rotaru et al., 2018), the multiplier can be bounded by constants, such that the equivalence can be approached approximately. In these cases, the spreading activation on the graph, defined in the canonical random-walk way, still considerably reflects the higher order cosine similarity. However, the rigorous equivalence has to be established at the cost of the canonical definition of spreading activation.

To summarize, I have defined general cosine similarity $\text{sim}^{m_1,m_2}(i,j)$ to capture the equivalence between vector similarity and intersecting activation on network. I have shown the traditionally-defined cosine similarity, i.e. $\text{sim}^{1,1}(i,j)$ - which only utilizes the 'after 1 time step' intersecting activation on the network - is a special case of the general similarity. The general similarity exploits 'higher-order' information embedded in the longer paths in the network. While the rigorous equivalence is established with adjustment to the spreading activation, an approximate equivalence can be guaranteed for the canonical spreading activation when the orders is not too high. In the next section, I relate this finding to existing behavioral and computational works which exploited higher order information. These models have created more structured semantic landscapes, and more accurately captures judgements on weaker semantic relatedness in human studies (De Deyne et al., 2016).

4.3 Spreading-Activation on Distributed Vector Representations

Provided with the formal equivalence between similarity in graphs and vector spaces, I deduce the structural and processing equivalence to a special case. In distributed representations, words or concepts are represented as vectors in a finite dimensional semantic space (Elman, 1990; Landauer & Dumais, 1997). Regardless of the number of dimensions or the content defining the representations in a vector space, this representation is mathematically equivalent to encoding the items in a two-layer network (Figure 4.4):One layer encodes each dimension as a node, and the other encodes each concept as a node. In the graphical representation of the space, each concept node is linked to each dimension node with an edge whose weight is the value of the concept vector on that dimension, and with no edge if the concept's value on that dimension is zero (see Figure 4.4 for further illustration).

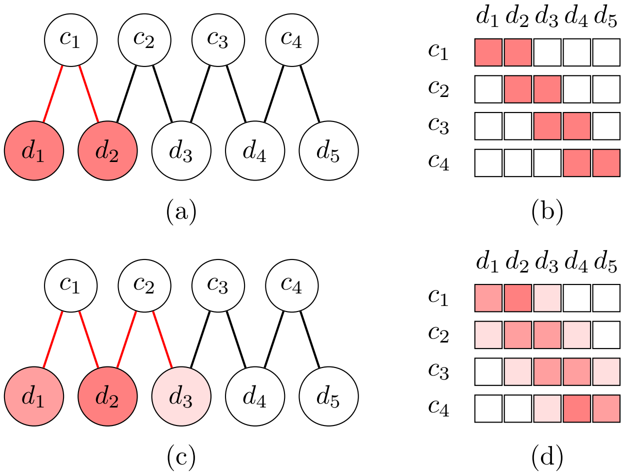


Figure 4.4: Two-layer network representation of vector. (a) Dimensions activated by c_1 with no iteration, mirroring the original vector representation of c_1. (b) Original vector representation of all four concept. (c) Dimensions activated by c_1 after one iteration. (d) Vector representation of all concepts after one iteration.

When formalized as a network, the distributed vector representation of concept can be represented as the pattern of activation in the entire network. At the first time step after concept c_i is activated, activation is spread to all directly-connected dimension nodes. For example, in Figure 4.4b, the concept c_1 has a non-zero value on dimensions d_1 and d_2, so c_1 spreads activation to these dimensions, resulting in d_1 and d_2 being activated proportional to the weight on that edge (which is also the concept's value on that dimension in the vector space). In this case, the translation of a distributed vector into activation on the two-layer graph has only used the direct connections between the concept and dimension nodes (colored red in Figure 4.4a).

More generally, as activation flows back and forth between the layers, the pattern of activation among the nodes can be 'enriched' by these iterations. The spreading activation activates more dimensions, especially dimensions that are more weakly associated with the concept in the original vector embedding, but which share many indirect connections through other nodes. For example, after only spreading along direct connections, the original concept c_1 only activates d_1 and d_2 (Figure 4.4a). After another iteration, c_1 also activates d_3 (Figure 4.4c), reflected by the activation having flowed to node d_3, a dimension not directly connected to c_1. In other words, with more iterations, the representation adds information from longer, indirect paths in the two-layer network (the red path between c_1 and d_3 in Figure 4.4). In this sense, the original vector representation itself encodes only partial information in the concept: the direct connections to other nodes. But any algorithm that allows the representation to take advantage of indirect connections, traversing longer paths between layers, may exploit 'higher-order' information embedded in the indirectly related concepts and dimensions (c_2, d_3 in Figure 4.4d). The result is a 'higher-order' concept representation that encapsulates not only the information in its original vector, but also how the concept relates to other concepts

As I have showed, the most common cosine similarity between two vectors is equivalent to the intersecting activation after one step of spreading from the sources. When generalized to an arbitrary number of time steps, the intersection of activation traverse through 'longer paths'. Specifically for the distributed vector graph, it means more iterations (Figure 4.4c) is equivalent to computing the cosine similarity on 'higher order' vectors of the concepts (Figure 4.4d). In the next section, I will use simulations to show that the representations formed using this process of spreading activation leads to more fine-grained semantic representations.

4.4 Modeling of Semantic Relatedness Using Higher Order Semantic Spaces

Spreading activation models have long been used to explain many semantic effects that do not fall out of direct associations or relationships between words. One such example is the explanation of co-activation of words that belong to the same category, even if not directly associated (such as *elephant* and *zebra*), because they share many features or associations (McRae et al., 1997). Another example is mediated priming - priming between items like *lion* from *stripes* through intermediaries like *tiger* (Balota & Lorch, 1986; Chwilla & Kolk, 2002). A third example is research by De Deyne et al. (2016), who showed spreading activation models that made use of indirect connections better predicted human judgments about word similarities than traditional cosine similarities.

The finding in De Deyne et al. (2016) indicates that including richer information embedded in the longer paths (higher order vector representations) potentially leads to a better account for semantic structure that better predicts human behavior. Thus far, I have shown that the spreading-activation process defined on the network can be 'rephrased' in the language of vector space. Similarly, due to the equivalence, the vector language can be translated into a 'network' narrative. I show that information embedded in the 'higher order' vectors can be extracted through activation spreading. The endeavor in turn could lead to more powerful semantic representations. Below, I provide two demonstrations of this improved semantic processing.

4.4.1 Simulation 1: Simple Artificial Dataset Demonstration

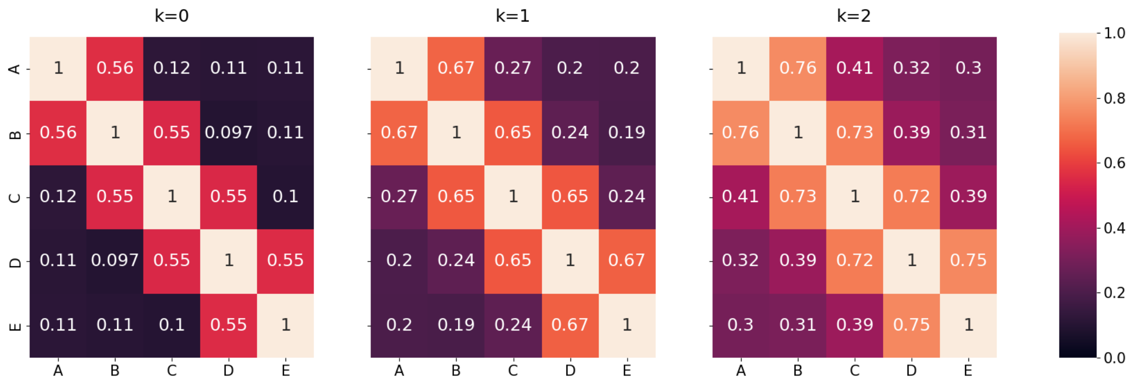


Figure 4.5: Average between-category similarities without and after iterated activation. (a) Similarities obtained without iteration. (b) Similarities after 1 iteration. (c) Similarities after 2 iterations.

To demonstrate the effect of iterating activation on a two-layer 'vector graph' (Figure 4.4), I ran a simulation using a simple dataset in which concepts have indirect and graded relations to each other (like that shown in Figure 4.4). To be more specific, in Figure 4.4b, c_1 does not overlap with c_3, c_4, but c_1 and c_3 can be bridged by c_2, while c_4 is more distant. To be more specific, I generated concepts as vectors with 18 dimensions $d_1, d_2, ...d_{15}$. The concepts' values on those dimensions were assigned such that they belonged to one of five categories (A, B, C, D, E). Concepts in the same category had high values on the shared dimensions (uniformly selected in the interval $(0.8, 1)$) and low values on the other dimensions (selected uniformly from $(0, 0.2)$). The dimensions that define a category, i.e. dimension with high values, partially

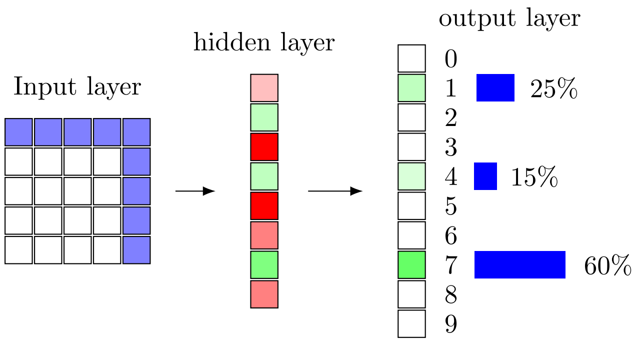


Figure 4.6: Digit classification task: The 10 digits ('7' in the illustration) were displayed in a 5×5 grid and were classified by a three layer neural network.

overlap with the dimensions that define another category. For example, category A took high values in $d_1, ...d_6$, B on $d_4, ..d_9$, C on $d_7, ..d_{12}$, D on $d_{10}, ..d_{15}$ and E on $d_{13}, ..d_{18}$. In this way, categories with adjacent letter labels shared three dimensions and therefore should be more similar to each other compared to other categories. Furthermore, non-adjacent categories do not directly share features, but they still have graded similarity according to how many features they jointly share with other categories. For example, although A and C did not share any feature, they are both similar to B, so A should be more similar to C than to D and E. Such a design simulated the organization of semantic knowledge in humans studied in previous works (Balota & Lorch, 1986; Chwilla & Kolk, 2002; De Deyne et al., 2016).

I generated 100 vectors in each category to create the first-order vector representations for each concept (i.e. vectors equivalent to a graph where only direct connections were activated). Then, I computed vector representations equivalent to letting activation spread for one and two iterations, as described in the previous section. For each of the three time steps (0,1,2), I computed the cosine similarity between all concept vectors, and took the average of the within-category concept similarity and between-category concept similarities to obtain the within and between-category similarities, for each of the five categories. I predicted that the similarity structure should be more differentiated with more activation iterations. Consistent with the prediction, the weakly related categories were not distinguishable with the original vectors (Figure 4.5, $k = 0$). However, with more iterations, the higher-order representations were able to discriminate between distant categories. (Figure 4.5, $k = 1, 2$).

4.4.2 Simulation 2: Between-Layer Iteration While Training Neural Networks

The algorithm described above also has implications for training neural networks. In typical neural networks with at least one hidden layer, the input and output layers are localist representations, and the hidden layers are distributed representations that mediate between input and output layers (Rumelhart, Hinton, & Williams, 1986). Normally, the output layer activation is obtained by multiplying the activation of the hidden layer by the weight matrix connecting the hidden layer to the output layer, and using that product as input into a nonlinear function like sigmoid or hyper tangent.

In the terms I have been discussing, this traditional process is equivalent to only using the first-order information in the graph, without exploitation of higher-order information. Activation in the network can be altered to take advantage of this higher-order information, by recalculating the weight matrix after k iterations, denoted as $\mathbf{W_{ho}^{(k)}}$:

$$\mathbf{W_{ho}^{(k)}} = (\sum_{i=0}^{k} \mathbf{A}^i)\omega_{\mathbf{ho}} \tag{4.13}$$

where $\mathbf{A} = \omega_{\mathbf{ho}}(\omega_{\mathbf{ho}})^t$ is the activation after one round of iteration between the layers. Note: the computation only concerns whether the activation propagated to the output layer is direct or after more rounds or iterations between the layers. It is not involved in the back-propagation, i.e. the formula for computing the derivative stayed the same. Inspired by results in previous sections, I conjecture that more iterations between the hidden and output layer would amplify the effect of training by integrating indirect information not directly connected between specific output and hidden units. As a consequence, the output layer activation calculated after more iterations will converge more quickly on the correct classifications.

To test the conjecture, I trained a three layer neural network (one hidden layer) on a digit classification task like that shown in Figure 4.7. In each training epoch, the network saw ten training trials (one for each digit) in a randomized order, made its predictions, and had its weights adjusted using a standard back-propagation algorithm (with a learning rate of 0.10). I manipulated two variables. The first was the number of hidden units in the network: 8 vs. 4 (4 is the minimum number of hidden units with which the network can solve the task). The second was whether the model was trained in the traditional way: $\mathbf{o} = f(\omega_{\mathbf{ho}}\mathbf{h}) + \mathbf{b}$, versus whether output activation was calculated according to Equation (4.13) using the weight matrix $\mathbf{W_{ho}^{(2)}}$ calculated with two iterations (only used for feedforward activation, the back-propagation was not affected). In each of these four conditions, I trained 10 randomly initialized networks for 1000 epochs.

The results of Simulation 2 supported the conjecture: the training was much faster in the iterated condition compared to the original condition, especially in the challenging case with only 4 hidden units. It was not a small difference: In the 8-hidden unit case, neural networks that computed higher-order correlations at the output layer tended to reach perfect performance after about 200 epochs compared to 500 in traditional models. In the 4-hidden unit models, perfect accuracy was obtained only slightly slower (around 400 epochs) for the higher-order model; whereas the traditional models took many thousands of epochs to reach perfect accuracy. Interestingly, the divergence started after about 100 epochs, consistent with the conjecture that the iterations would help the neural net to converge more rapidly, after the 'right' direction had been found.

4.5 Discussion

In this chapter, I demonstrated an equivalence in both structure and process between graphical spreading-activation models and vector space models of knowledge representation. Specifically, I showed that the intersection of activation traversed through paths with varied length on a graphical model is mathematically equivalent to the cosine similarity of higher order vector representations. This finding bridges the network and vector space representational approaches: (i) It helps to explain the modeling power of network structure De Deyne et al. (2016) from the vector space perspective; (ii) It leads to better understanding of two-layer network representations using distributed vectors, showing how spreading-activation on the two-layer network exploits higher-order semantic information embedded in the vector space; iii) I show that iterating activation between vectors in a neural network results in a graded representation of the semantic information (Figure 4.5); and iv) I show that it significantly accelerates training in neural networks (Figure 4.7). These findings lead to two directions for future work.

First, it raises a series of modeling and theoretical questions concerning the nature of semantic representation and processing. On the modeling side, there are various ways to define spreading-activation on

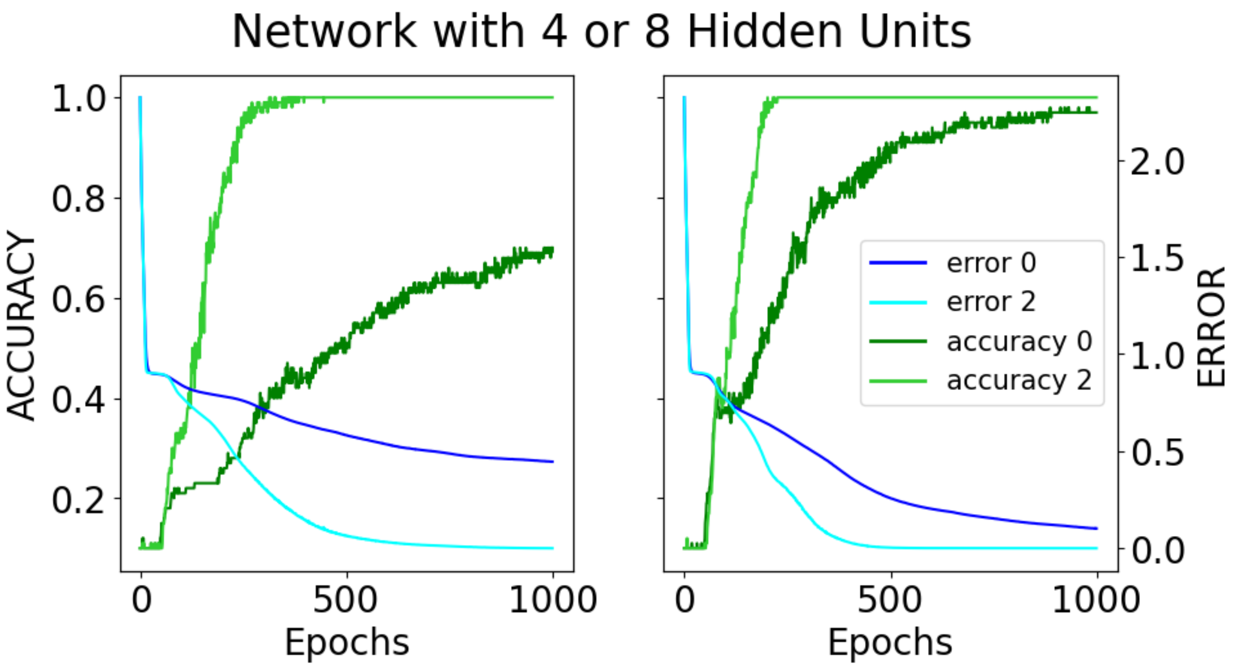


Figure 4.7: Learning in the digit classification task was faster when the neural net had two rounds of iterations between the hidden layer and output layer before feed-forward propagation (lighter lines vs. darker lines), in terms of both accuracy (green lines) and error rates (blue line). The difference was more dramatic in the challenging condition (left).

a network and multiple metrics to evaluate the relationship between vectors (Kumar et al., 2022; Lund & Burgess, 1996). Do processes and measures defined on one data structure have equivalent form in the other data structure (like cosine similarity in the vector space being equivalent to activation intersection on graphs), and could these bridges benefit in semantic models with real-world data? Regarding theories of semantic cognition: what leads to, and under what circumstances are, longer paths along the network (or higher-order vector spaces) beneficial? How does the spreading-activation process on unstructured or layered networks potentially speak to human cognitive processes in semantic tasks? Scrutiny of these questions may deepen our understanding of the nature of semantic representations.

Second, the findings increase our understanding and the capabilities of neural network models. Currently, the simple feedforward models like those used in the simulation are not capable of taking advantages of information not captured by the direct connections between units. The capability *is* present in models making use of other architectures and algorithms, such as recurrent attractor models (Cree, McRae, & McNorgan, 1999) and Hebbian learning models like Boltzman machines. However, these models are not as popular as simple feedforward architectures, mainly due to the amount of data they need and the amount of time they need to train. The approach presented in this Chapter has the potential to add to neural networks the ability to take advantage of the higher-order information, and at the same time decrease the amount of data and training that is needed. Future studies can help us better understand this potential, and test the approach on scaled-up models to examine whether the technique is applicable in more real-world settings and datasets.

Chapter 5

Constituent Tree Network

One of the hallmarks of human language ability is the capacity to judge differences in the truth and meaningfulness (plausibility) of linguistic expressions. The latter topic is investigated in Chapter 3, where I evaluated the direct and indirect lexical relations between word pairs (Mary$_a$- trap, wolf$_a$ - trap), to judge on the plausibility of sentences like *Mary trapped the rabbit* or *The wolf trapped the squirrel*.

Notice that the above-mentioned dependencies are between two lexical items. Nevertheless, some of the lexical dependencies may involve three or more lexical items and may not be reduced to lexical-pair dependencies. While both *spelling* and *brake* can be *checked*, the ease of processing depends on the agent noun (the third variable). It is easier to process congruent sentences like *The journalist checked the spelling* and *The mechanic checked the brake* compared to the incongruent agent-event pairs (Bicknell et al., 2010). Moreover, Bicknell et al. (2010) showed that the effect in their experiments was not due to the agent-patient thematic fit (*mechanic - brake* vs. *journalist - brake*). Their result indicates that some of the lexical dependencies may not be reduced to (two-way) relations between word pairs. They are three-way (multi-way) interactions, i.e. there is the effect from the third word.

In comparison to learning the dependency between multiple words provided in the language input, a more challenging task is to infer on the dependencies involving 'novel' lexical combinations. For example, suppose one has seen *The journalist checked the spelling* and *The mechanic checked the brake*, what would be the thing that a *poet* checks, the spelling or the brake? (Supposed that one has never seen *the poet checked...*) It has been argued that the natural language is generative and thus productive (Fodor & Pylyshyn, 1988), so that the infinite grammatical lexical combinations may never be exhausted in any finite language experience. How could people understand and effectively judge the plausibility of the novel sentences based on their finite input? I argue that a critical capability required for natural language understanding is to first learn the multi-way lexical dependencies from experience and then generalize the learned relations to novel lexical combinations. In other words, a crucial step for a system to be competent in processing meaning is to capture the diverse lexical dependencies (two or multi-way) in the generative natural language.

The question I attempt to address in this chapter is, what representations and computational mechanisms are needed to learn and generalize multi-way lexical dependencies in the language input (to make plausibility judgments on more complex sentences). The remainder of the chapter is structured as follows: In Section 5.1, I formulate the theoretical problem in a series of formal computational task, i.e. the two-way and three-way learning and generalization tasks. I argue that in order to solve the most challenging three-way (multi-way) generalization task, the model needs to form representations of phrases or phrasal relations (lexical relations that involve phrases) in a compositional way. I refer to such generalization based on compositional representations 'compositional generalization' and discuss key factors leading to successful compositional generalization. In Section 5.2, I present a type of distributional graph, i.e. the Constituent Tree Network (CTN), which provides an intuitive blueprint towards solving the task. In Section 5.3, I describe a carefully constructed artificial language that embeds the critical design for the compositional generalization. I test both the LON (Co-occurrence Graph) and the CTN with a series of experiments in section 5.4 and show that the specific representation, i.e. the constituent structure, is the key to the success of CTN. I end the chapter by discussing the significance of the multi-way dependency/compositional generalization problem, and the implication of the CTN model's success on it.

5.1 Introduction

5.1.1 The Tasks

The Learning Task

I formulate the problem of learning and generalizing three-way lexical dependency with an example (the full design will be presented in the method section). Consider a linguistic corpus consisting of the following four sentences *Mary preserves cucumber with vinegar, Mary preserves berry with dehydrator, Mary grows cucumber with fertilizer, Mary sprays strawberry with insecticide.* In these sentences, I focus on the three components that interact with other other: the verb (preserve/grow), the patient noun (cucumber/strawberry) and the instrument noun (vinegar/dehydrator). The lexical choice of the instrument in the 'preserving' event is affected by both the verb and the patient noun, while it may not be determined solely by any of the two items[1]. It is the phrase 'preserve cucumber' that selects on the instrument 'vinegar'. The verb 'preserve' by itself allows both 'vinegar' and 'dehydrator', and the patient noun 'cucumber' associates with both 'vinegar' and 'fertilizer'. In this scenario, the dependency between the verb, the patient and the instrument in 'Mary preserves cucumber with vinegar' is three-way. That is, the exact lexical choice of each one of the three variables (verb, patient, instrument) is conditioned **jointly** on the choices of the rest two.

In addition, I include two more sentences involving the two verbs 'grow' and 'spray': *Mary grows potato/lettuce with fertilizer.* and *Mary spray orange/apple with fertilizer.* Unlike the case in 'preserve' sentences, the instrument in the 'grow' and 'spray' sentences are independent of the patient noun: It is always *fertilizer* for 'grow' and *insecticide* for 'spray', regardless of which noun is in the patient position. In this way, the dependencies between the three components in the 'grow' and 'spray' sentences are no longer three-way: they reduce to the two-way dependencies within the verb-instrument pairs.

Both the three-way and the reduced two-way interactions exist widely in natural language. A competent semantic system should be able to learn these types of dependencies provided in the language input. In this and the next chapter, I refer to mastering lexical relationships in the input as the 'learning task'. For any three-way interactions, models need to specify which instrument is the best fit for a given verb-patient ('VP' is used to denote verb-patient pair in this dissertation) combination. For the two-way interactions, models are expected to select the correct instrument given a verb.

[1] Here I talk about the verb-patient selecting the instrument, while formally the dependency is mutual and symmetric. I could equivalently present the problem in the way that the verb-instrument pair decides on the patient, or the patient-instrument pair selects the verb. I choose selection on the instrument merely due to the ease of description and understanding.

The Generalization Task

Human and powerful semantic models not only learn the patterns in the language input, but also generalize from what they learn to infer on novel lexical combinations. Suppose sentences like *Mary grows pepper, Mary preserves pepper* are also provided in the corpus. This 'novel' patient *pepper* is associated to the verbs in the sentences while the instruments are purposefully left out. This type of patients are referred to as 'instrument-vacant patient' in contrast to the 'instrument-associated patient' like *potato* and *cucumber*, which always co-occur with an instrument. I am interested in how the models judge on the plausible instruments for the novel verb-patient pairs (grow/preserve pepper). For a sentence in which the instrument solely depends on the verb, generalization is not contingent on the patient (in the corpus). So *fertilizer* should be the only choice, no matter what one *grows*. However, for sentences in which the instrument does not solely depend on the verb, more information on the novel patient noun is needed. To direct the generalization, I make these patients 'more similar' to some of the original patients but not others. For example, I make *pepper* more similar to *cucumber/potato/lettuce*, and less similar to the fruits. In this way, it is expected that models find *preserve pepper* more similar to *preserve cucumber* than to *preserve berry* and therefore should select *vinegar* but *dehydrator* as a more plausible instrument for the event.

Critically, I make the VPs similar to each other in a combinatorial way, but not in a 'holistic' way. To be more specific, I make *cucumber* and *pepper* distributionally similar, i.e. the two words have a lot of contexts in common. On the other hand, I make the phrases *preserve pepper* and *preserve cucumber* as a whole distributionally dissimilar to each other. That is, the two-word chunks *preserve pepper* and *preserve cucumber* do not share a lot of contexts. I am interested in the capability of the models to build up the similarity of the whole by combining the similarities between their corresponding parts. This set-up resonates with the pattern observed in linguistic cognition: We can tell that *John grows cucumber with fertilizer* is more similar to *Mary plants squash with manure* than to *Mary preserves berry with dehydrator*, without hearing these sentences before. Building up multi-word relations from parts is crucial to language comprehension, and I will come back to the point in the general discussion of this chapter.

To summarize, the 'novel' patients are associated with certain verbs, but no three-way dependencies involving them are provided in the input. The models need to infer on the plausible instrument for the verb-patient pair based on the type of the verb and also the (combinatorial/compositional) similarity between the instrument-vacant patient and the instrument-associated patients. I refer to these tasks as the generalization tasks in which models generalize on the two-way and three-way dependencies.

Bringing the tasks together, I am primarily interested in how distributional models learn the multi-way lexical dependencies (the two-way and three-way dependencies) provided in a linguistic corpus, and then generalize from the learned dependencies to novel multi-way lexical relations. In particular, I am interested in how the model generalize combinatorially (compositionally) to novel lexical combinations (novel VPs), whose distributional features as a whole has not been identified in the corpus, but whose constituent parts are well characterized by in terms of distributional pattern.

Compositional Generalization

Among the four types of tasks (two-way learning, three-way learning, two-way generalization, three-way generalization), the three-way generalization task is most challenging. As illustrated above, to infer the proper instrument for *preserve pepper*, i.e. *vinegar*, the only cues presented to the model are the word-word lexical dependencies, i.e. *preserve - vinegar* and *pepper - vinegar*. To obtain the three-way dependency *preserve* *pepper* - *vinegar*, a model needs to **compose** the two-way dependencies involving the constituents of the VP, i.e. *preserve* - *vinegar* and *pepper* - *vinegar*. Overall, this type of generalization to novel situations based on knowledge of smaller components, is referred to as compositional generalization.

The compositional generalization entails a series computational abilities. First, the model needs to properly evaluate the two-way (word-word) lexical dependencies that has been presented in the input. These include syntagmatic relations like *preserve* - *vinegar* and *cucumber* - *vinegar*, as well as paradigmatic relations such as *cucumber* - *pepper*. Then, it requires the model to combine the syntagmatic and paradigmatic relationships to form more indirect syntagmatic relations like *pepper* - *vinegar* (by combining *cucumber* - *pepper* and *cucumber* - *vinegar*). These are exactly the capabilities investigated in Chapter 3 and 4. According to the two chapters, LON has no trouble forming these indirect relations. The dilemma is in the next step, i.e. deriving the three-way dependency between the novel VP and the instrument, i.e. *preserve pepper* - *vinegar* from the two-way dependencies. I will clarify why this is difficult in Section 5.3, and show that LON falls prey to this step in Section 5.4.

Alternatively, a model may learn the phrasal syntagmatic relationships, i.e. *preserve cucumber* - *vinegar* from the input, and form 'phrasal similarity', i.e. the paradigmatic relationship between the VPs (*preserve pepper* - *preserve cucumber*) by composing the two-way dependencies. Then it combines the phrasal syntagmatic and paradigmatic relations to derive the more indirect phrasal relation *preserve cucumber* - *vinegar*. This approach is different from the former procedure, but both of them require a similar set of computational capabilities. Regardless of the exactly approach, it should be clear that compositional generalization is not a single computational step, but rather a collection of separate abilities that must be aligned to achieve a common goal. It follows that, if a model is found to reliably perform compositional generalization, it in general should be more likely to achieve success on each of the sub-tasks listed above.

5.1.2 A Novel Approach

In summary, compositional generalization can be a promising approach for making inferences about novel linguistic expressions. However, many existing distributional semantic models in the psycholinguistic literature are not ideally suited to perform compositional generalization. As I will show with more details in Chapter 6, few of existing distributional models possesses all the required capabilities. Most of the models come with their own distinct set of advantages and only possess part of the capabilities. As a result, they tend to perform well in the easier learning and generalization tasks, but not the three-way generalization tasks that requires compositional generalization.

In light of these concerns, I propose the Constituent Tree Network (CTN), which offers a solution to these issues. The CTN is a semantic network created by connecting language input that has been converted into a graphical format. The CTN explicitly encodes 'higher-order' relationships between phrases and lexical items, and stores this information alongside information about distributional structure, while also avoiding entanglement. The studies presented in this chapter will demonstrate that the CTN can be used to infer both the similarity among words and phrases, and the thematic fit between a phrase and a word regardless of whether they are novel or familiar. As such, the CTN has the potential to be an invaluable tool for advancing our understanding of how humans make semantic inferences about novel complex expressions, and the productivity of the human language system more generally (will come back to this point in the discussion). In the following section, I provide a detailed discussion of the CTN's architecture, how it can be trained, and how inference is performed.

5.2 Model

5.2.1 Training: Constructing the Semantic Network Structure

How the CTN model is built have been introduced in section 2.3. Here I briefly summarize the process with specific emphasis. Unlike previous studies, in which words have either been connected manually by the researchers (Collins & Quillian, 1969; Collins & Loftus, 1975) or using free-association norming data (De Deyne et al., 2016; Kenett et al., 2011), words and phrases in the CTN are connected according to both their constituent phrase structure and their co-occurrence frequency in a corpus of natural language. This makes the CTN more straightforward to construct: it only requires a corpus that has been syntactically parsed, and it is therefore less influenced by the intuitions of the researcher or of participants in free-association studies. Training in CTN is analogous to Hebbian learning: Linguistic expressions that co-occur in the language input become connected, and strengthened in proportion to the number of observed co-occurrences.

The first step of training is to transform all sentences in a corpus into graphical form with constituent parsing, as shown in Figure 5.1a. Next, the parse trees are joined at shared nodes to form a network (Figure 5.1b). The frequency of co-occurrence of two connected expressions in the resulting network is used to weight the connection between their nodes. Recall that this process is referred to as 'connecting language'. At the end of training, the result of 'connecting language' is that raw co-occurrence between words and phrases is encoded in a graphical topology. To be more specific, words or phrases occurring close to each other in the corpus (e.g. in the same sentence) are only separated by a few edges in the network (e.g. *cake* and *knife*), whereas other expressions that do not occur close to each other in the corpus are separated by a larger number of edges (e.g. *pie* and *knife*).

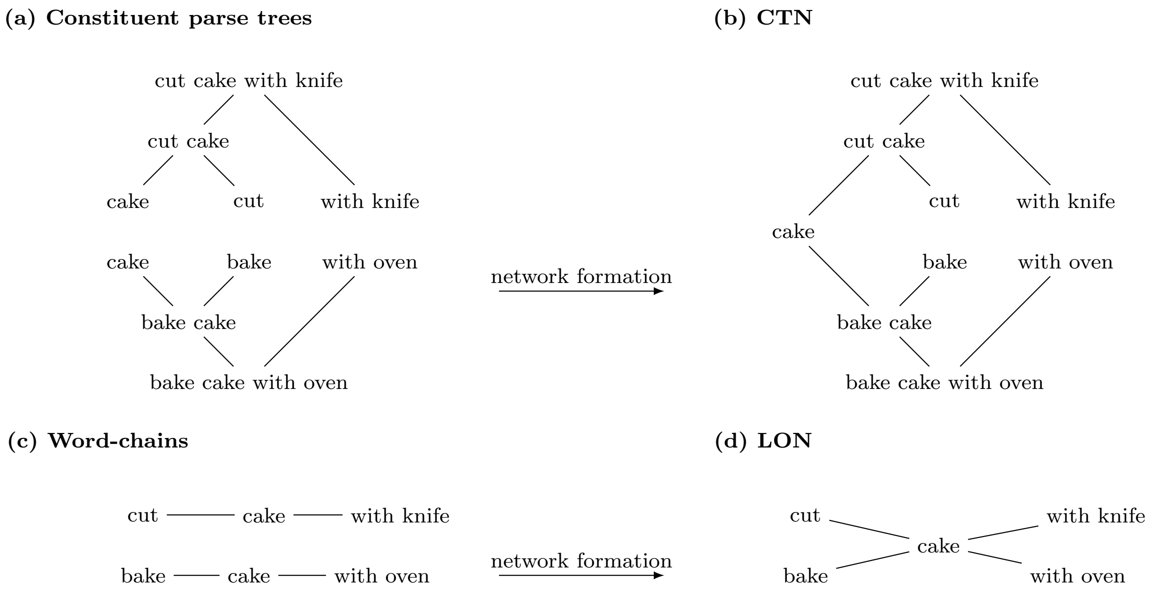
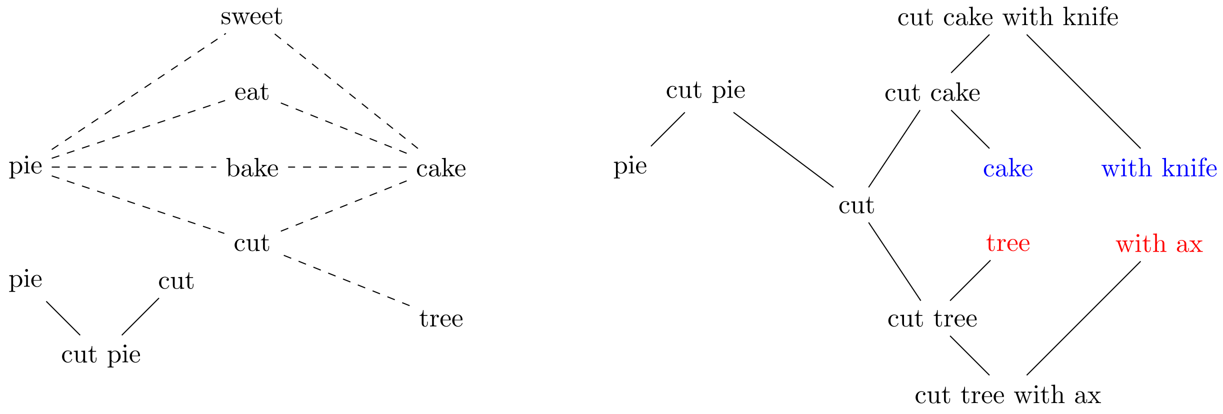


Figure 5.1: Formation of the network structure in the Constituent Tree Network (CTN) and the Linear Order Network (LON) given the mini corpus *'cut cake with knife, bake cake with oven'*. **(a)** The input to the CTN consists of constituency-parsed trees for sequences in the mini corpus. **(b)** The CTN formed is able to reflect dependency between verb phrases and instruments in the input corpus. **(c)** The input to the LON consists of word chains, formed by connecting adjacent words in the mini corpus. **(d)** The LON formed does not specify which instrument is coupled with which verb phrase.

Benefits of connecting language

To better understand the value of 'connecting language' in the current context, consider the complex expression *cut pie* in Figure 5.2b. After the connection of individual parse trees during training, *cut pie* becomes indirectly connected to all observed instruments of *cut* (e.g. *knife* and *ax*) in the resulting network. Although *cut pie* did not co-occur with an instrument in the training corpus (unlike *cut cake* and *cut tree* that are assigned with specific instruments, i.e. *knife* and *ax* in the corpus), the CTN is able to make inferences by following the indirect paths between *cut pie* and instruments observed during training. This makes it possible to evaluate the semantic relatedness of any two nodes in the model's network, especially, those have never occurred together in the training corpus.

While the indirect paths make it possible for the model to consider instruments that never co-occurred with novel phrase *cut pie*, the model still needs to know how similar *cut pie* is to the familiar phrases *cut cake* and *cut tree*, in order to infer on the instrument (Figure 5.2b. Like other distributional graphs, this information is in the number of indirect paths between two nodes. As shown in Figure 5.2b, *pie* and *cake* are connected by an indirect path that traverses *cut* and *cut cake*. Implicit in this path is the information that both patients share the same verb*cut*, more clearly illustrated in Figure 5.2a, where dashed lines represent all paths of length 2 that traverse one intermediate phrasal node. The dashed lines are a simplified representation of the true semantic network structure shown in Figure 5.2b. This diagram illustrates that *pie* and *cake* are highly connected with many short paths leading from one to the other. If *pie* and *cake* share more of these short indirect paths compared to *pie* and *tree*, the network can judge the former pair to be more similar. It is not difficult to imagine that in a large corpus of English, *cake* and *pie* can be found in highly substitutable contexts like *sweet X*, *eat X*, and *bake X*. The number of shared contexts in the corpus determines the number of shared paths in the semantic network. Note that I do not claim that *cake* and *pie* do in fact share more contexts than *cake* and *tree*. Regardless of the actual contextual overlap, I only argue that if such distributional differences exist in English, they will be reflected in the structure of the network.



(a) Distributional Similarity **(b) Semantic Network Structure**

Figure 5.2: The CTN achieves compositional generalization by relying on two key information sources: distributional similarity between lexical items, and constituent structure. The former connects *pie* to *cake*, and the latter indirectly connects *cake* to *knife*. Note that *knife* need not co-occur with *pie* for generalization to be successful. **(a)** The distributional similarity between *pie* and *cake*. Each dashed line represents co-occurrence in the training corpus. **(b)** The relationship between patients of *cut* and the instruments they co-occurred with. Plausible patient-instrument combinations are shown in the same color.

Benefits of the Constituency Structure

However, 'connecting language' itself does not guarantee compositional generalization. With the connections, the model may know either *cut cake* or *cut tree* is closer to *cut pie*, while it still needs to pair up the familiar phrases with their observed instruments (Figure 5.2 b). It turns out that the structure of the connected language matters. Both CTN and LON are trained by 'connecting language', but LON includes less information about linguistic structure. In this chapter, I mostly focus on the CTN, and use the LON as a baseline to test the importance of the constituent structure for compositional generalization.

Constituent structure is encoded in the CTN using special purpose nodes referred to as higher-order or 'phrasal' nodes. An example of such nodes are those labeled *cut cake* and *cut cake with knife* in Figure 5.1a and b. Not only do phrasal nodes help identify which expressions should be grouped together in the same sentence, they are also the building blocks on which compositional generalization relies. An important advantage of phrasal nodes is that their presence in the network enables the use of 'graphical distance' for better quantifying the relatedness between a word and a phrase (and the constituents of the phrase). To illustrate, consider the paths between the nodes labeled *cut*, *bake*, and *knife* in the network shown in Figure 5.1b. Since *cut* and *knife* are components of the same phrase *cut cake*, they are both connected to the same phrasal node *cut cake*. The presence of this node in the network results in a shorter distance between *cut* and *knife* relative to the connection between *bake* and *knife*. In contrast to the LON, where *cut* and *bake* are equally distant to *knife* and *oven* (Figure 5.1d), *cut* is closer to *knife*, and *bake* is closer to *oven* in the CTN. In this way, unlike LON, the CTN is able to capture the fact that *knife* is a thematically better fit to *cut cake*, and that *oven* is a thematically better fit to *bake cake*. In other words, the CTN captures the dependencies between complex expressions (e.g. VPs) and individual lexical items (e.g. instruments). These phrasal dependencies, as I show, are crucial to the success of compositional generalization.

To summarize: the CTN ingests **constituent parse trees** by **connecting** them at shared nodes. As a contrast to the linear chains in LON, the **parse tree** provides the grammatical structures needed to perform generalization and the **connection** enables generalization to novel combinations of familiar items in familiar grammatical structures. While these 'structural' readiness are the foundation of CTN's success in the compositional generalization task, it still needs an effective measures to translate the structure into quantified semantic relatedness, which I explain in the next section.

5.2.2 Inference

With the structure of the CTN in place, two questions remain regarding its ability to perform compositional generalization: First, how is the semantic relatedness between two lexical items (e.g. *pie* and *cake*) measured? Second, how can the CTN infer the relatedness between a lexical item and a phrase (e.g. *cut pie* and *knife*? As a particular type of Distributional Graph, CTN accesses relatedness through the spreading-activation algorithm defined in section 2.1. Readers who have thoroughly read Section 2.1 may skip the following section 5.2.2, as it is largely repetitive to Section 2.1. Nevertheless, the definitions in this section are presented alongside with explanations of the problem in this Chapter, i.e. compositional generalization. Therefore, I also recommend interested readers to go through the defining process with the context problem in mind, which may lead to better understanding of the spreading-activation algorithm.

Quantifying Semantic Relatedness on Graphs

In this section, I lay out the formalism for computing semantic relatedness in the CTN and LON. I begin by defining the relatedness between two neighboring nodes in the network, and work our way up to the semantic relatedness between nodes that are separated by an arbitrary distance.

The relatedness between two neighboring nodes A and B is computed as the normalized weight of the edge connecting the two nodes, denoted as ω'_{AB}:

$$\omega'_{AB} = \frac{\omega_{AB}}{\sum_{X \in \mathcal{N}(A)} \omega_{AX}} \tag{5.1}$$

where ω_{AB} is the weight of the edge linking nodes A and B, and $\mathcal{N}(A)$ is the set of nodes that are neighbors of A in the network. In the CTN and LON, all edge weights are proportional to the co-occurrence frequency of the lexical items denoted by the two nodes in a training corpus. Therefore, the amount of activation that A spreads to a neighbor X is proportional to the co-occurrence frequency between A and X. As a consequence, lexical items that co-occur more frequently with the item denoted by A, receive more activation from A. Normalization reduces the amount of activation arriving at a target node B if source node A has many neighbors relevant to A. This captures the intuition that if a lexical item co-occurs frequently with many other items, the semantic relatedness between any pair should be small.

Next, I define the relatedness between any two nodes separated by a path P. Let P be a path of length k in the network connecting node A and C. The semantic relatedness between them is defined as:

$$\mathrm{SR}^P(A, C) = \prod_{1 \leq i \leq k} \omega'_i \tag{5.2}$$

where ω'_i is the normalized weight of an edge along path P. This definition features how much activation spreads from A to C via path P. Note that the formula is sensitive to graphical distance: Because the weight of each edge along path P is less than 1, the activation must disperse as it flows from A to C. As a result, the larger the graphical distance, the more activation diffuses away along the path. Therefore, the greater the distance between A and C, the smaller the semantic relatedness between the two lexical items they represent. As illustrated in Figure 5.2b, the activation of *cake* resulted in more activation arriving at *knife* than at *ax* (illustrated with color difference). This is in agreement with the strong thematic relationship between *cake* and *knife* relative to *cake* and *ax*.

Finally, I define the general semantic relatedness between a source node A and any node D in the same semantic network as

$$\mathrm{SR}(A, D) = \sum_{P \in \mathcal{P}_{A,D}, L(P) \leq d(A,D)+n} \mathrm{SR}^P(A, D) \tag{5.3}$$

where $\mathcal{P}_{A,D}$ is the set of paths between A and D, $L(P)$ is the length of P, $d(A, D)$ is the graphical distance between A and D, and n is the upper bound on the length of P. In other words, the general semantic relatedness of node A and D is the sum of all activations arriving at D from A through paths that are no longer than n. By excluding paths longer than n, the set of paths that are considered, $P \in \mathcal{P}$, only contains relatively short paths. I filter the longer paths as they contribute less activation at D compared to shorter paths, and are less informative (De Deyne et al., 2016; Rotaru et al., 2018).

Given that this measure of spreading-activation considers only a bounded set of paths, if two nodes share a larger number of paths in that bounded set, then they will activate each other more strongly. For example, under the assumption that there are more paths between nodes labeled *pie* and *cake* compared to *pie* and *tree*, *cake* would be more strongly activated than *tree*. In turn, *knife* would receive more activation from *cake* relative to *ax* (Figure 5.2b).

Operationalizing Phrasal Relatedness

I explained how spreading-activation can be used to compute lexical relatedness, such as between *pie* and *cake*. However, how can this approach be scaled to the relatedness between a phrase and a lexical item? With the availability of phrasal nodes in the CTN, it is possible to use the general lexical relatedness as the building block of a larger formula for computing the relatedness between complex expressions. (Note that this cannot be done with the LON, which lacks phrasal nodes). The relatedness between arbitrary phrase (phrases with arbitrarily complex structure) has been defined by Equation 2.6 in Section 2.3.3, and the phrasal relatedness investigated here is the simplest special case: The phrases only have two words.

Applying Equation 2.6, the phrasal relatedness between a two-word phrase and a word is defined as the product of two word-word relatedness scores. For instance, the relatedness between the verb phrase *cut pie* and an instrument can be written as:

$$\mathrm{SR}(\textit{cut pie}, \textsc{Instrument}) = \mathrm{SR}(\textit{cut}, \textsc{Instrument}) \cdot \mathrm{SR}(\textit{pie}, \textsc{Instrument}) \tag{5.4}$$

As mentioned in Section 2.3.3, another way to compute phrasal relatedness would be to compute the activation originating and/or arriving at phrasal nodes in the CTN; however, such an approach would only work for those phrases that the CTN has encountered during training. This is insufficient, given that the set of expressions that are grammatical under a language is virtually infinite (see the discussion of the productivity of language in Section 5.5). For such reason, I did not follow this approach.[2]

Note that the phrasal relatedness is broken apart into two distinct components, one for each word in the verb phrase. With this formulation, SR(*cut pie*, *knife*) should be scored higher than SR(*cut pie*, *ax*) provided that *pie* and *cake* are distributionally more similar than *pie* and *tree*. This difference is due to the second term in the equation, SR(*pie*, Instrument), which denotes the relatedness between *pie* and a given instrument. If *pie* and *cake* are distributionally very similar, the strong activation of *cake* in the semantic network will spread to *knife* more efficiently than to another instrument.

Compared to other distributional models (e.g. (Mitchell & Lapata, 2010; Baroni, Bernardi, & Zamparelli, 2014)), the compositional function used by the CTN is relatively simple. Consequently, any performance gained by the CTN relative to previous models should not be attributed to the compositional function alone. Instead, I argue that any performance gained by the CTN should be attributed to (1) the graphical data structure that explicitly encodes constituency, and (2) the spreading-activation algorithm which takes care of path length, number of shared paths, and edge weights to provide a graded relatedness measure.

Using a more general formulation, I define the relatedness between a 2-word phrase (w_1, w_2) and a word w_3 as

$$\mathrm{SR}((w_1, w_2), w_3) = \mathrm{SR}^*(w_1, w_3) \cdot \mathrm{SR}^*(w_2, w_3) \tag{5.5}$$

where SR* denotes what I will refer to as the 'adapted' lexical relatedness. The adapted lexical relatedness is similar to the general lexical relatedness, but accounts for a special situation that can arise in the CTN that would hamper its accuracy during compositional generalization.

[2] Although I do not allow phrasal nodes to act as source or target nodes, phrasal nodes are allowed to transmit activation like any other node.

The Adapted Lexical Relatedness

In this chapter, the adapted word-word relatedness score for LON is equal to equation 5.3 with n set to 2. To be precise, the adapted lexical relatedness in the LON is equal to the the sum of the activation arriving at the target node via the first and second shortest paths from the source node.

In the CTN, the adapted lexical relatedness also follows Equation 5.3. However, in addition to considering only the n shortest paths, I also disallowed activation from spreading along the path that traverses the sub-tree that represents the phrase (w_1, w_2). Without this adaptation, the phrasal relatedness can, in some cases, be reduced to a function of the lexical relatedness between the verb and instrument only. This is not ideal, considering that the information provided by the patient would be effectively discarded. When this information is discarded, compositional generalization suffers.

How does the exclusion of that path force the CTN to consider the patient during generalization? Recall that the phrasal relatedness between, say, the verb-patient combination *cut pie* and the instrument *knife* is the result of multiplying two terms, namely $\text{SR}(\textit{cut}, \textit{knife})$ and $\text{SR}(\textit{pie}, \textit{knife})$. During computation of the second term, $\text{SR}(\textit{pie}, \textit{knife})$, activation will spread from *pie* to the phrasal node *cut pie*, and then to *cut*. At this point, the shortest path to the *knife* is along the same path that activation must travel when computing the first term, $\text{SR}(\textit{cut}, \textit{knife})$. When the path from the patient *pie* to the instrument *knife* via the phrasal node *cut pie* is the shortest path in the network (as is the case with networks constructed from the artificial corpus here), the information provided by following this path would be redundant with the information provided by following the path from the verb to the same instrument. If so, the phrasal relatedness could be reduced to a function of only the first term, $\text{SR}(\textit{cut}, \textit{knife})$. More precisely, the phrasal relatedness $\text{SR}(\textit{cut pie}, \textit{knife})$ could be reduced to $c \times \text{SR}^2(\textit{cut}, \textit{knife})$ where c is a constant that denotes $\text{SR}(\textit{pie}, \textit{cut})$.

5.2.3 Summary

In Section 5.2.1, I showed that the constituent trees explicitly encoded in CTN may provide the topological structure benefiting compositional generalization, as a contrast to the LON that lacks such structure. Furthermore, the structural 'advantage' demands an effective relatedness measure on the graph, which I presented in 5.2.2. Notice that both the constituent structure and the effective measure are needed towards the compositional generalization task. As I will show and explain in Section 5.4 and 5.5, lacking the critical structure, LON equipped with the same spreading activation algorithm still fails the task.

5.3 Corpus

I designed a series of experiments to examine the ability of different models to perform compositional generalization. The models learned linguistic structure from a carefully constructed artificial language corpus that was designed to have a specific semantic role structure. Once trained on this corpus, the task of the models was to judge the semantic plausibility of sentences they had seen during training, as well as to judge the the plausibility of sentences they had not seen, but which were consistent with the semantic rules used to generate the corpus.

5.3.1 Corpus Structure and Experimental Design

The corpus was a collection of sentences of the form "AGENT-VERB-PATIENT-INSTRUMENT." (e.g., "Mary preserved cucumber with vinegar"). In the corpus, whether the INSTRUMENT occurred in the sentence or was omitted depended on the verb and patient noun, and the manner is described below. The specific words in each sentence were chosen using rules instantiating a set of 8 isomorphic verb-patient-instrument semantic structures. These semantic structures described which of the six verbs (4 types) in that semantic structure could co-occur with which of the six patient nouns; whether an instrument noun would occur in that sentence; and if an instrument would occur, which of the four instruments it would be. An example of one of the 8 semantic structures is shown in Table 5.1. Each of the 8 semantic structures replicated the same exact structure, but used different sets of words.

Table 5.1: Example sentences from the artificial corpus, for 2 patient categories only. Each category is associated with 4 verb types. Type-3 and 4 verbs always occur with instruments except when patient is *Test* (indicated by bold-face).

Category	type-1	type-2	type-3		type-4	
VEGETABLE	J dice cucumber	J ferment cucumber	J grow cucumber	with fertilizer	J preserve cucumber	with vinegar
	J dice potato	J ferment potato	J grow potato	with fertilizer	J preserve potato	with vinegar
	J dice **pepper**	J ferment **pepper**	J grow **pepper**		J preserve **pepper**	
FRUIT	J dice berry	J pick berry	J spray berry	with insecticide	J preserve berry	w. dehydrator
	J dice apple	J pick apple	J spray apple	with insecticide	J preserve apple	w. dehydrator
	J dice **orange**	J pick **orange**	J spray **orange**		J preserve **orange**	

Patient Nouns

Each structure was organized such that the six patient nouns that specifically associated with it were divided into two semantic categories (e.g., FRUITS and VEGETABLES in this example in Table 5.1) containing three words each. Of the three nouns in each category, two were *Control* nouns that were allowed to co-occur with instruments, and one was a *Test* noun that never co-occurred with an instrument noun in the same sentence in the corpus.

Verbs

Each of the four types of verbs in each structure differed systematically with regard to its pattern of co-occurrence with patient and instrument nouns. Type 1 verbs were *shared category* verbs that could co-occur with patients from both semantic categories. In addition, Type 1 verbs never co-occurred with instruments. Type 2 verbs were *category specific* verbs that only co-occurred with patient nouns from one semantic category, and like Type 1 verbs, never co-occurred with instruments. Type 3 verbs were *category specific* verbs like Type 2 verbs, but did sometimes co-occur with instrument nouns (again, depending on the patient noun). Type 4 verbs were *shared category* verbs that sometimes co-occured with instrument nouns (depending on the patient noun in the sentence).

Instrument Nouns

Two facts are important to note about the instrument nouns. First, as already described, whether an instrument occurred at all in the sentence depended on the verb *and* the patient noun. Type 1 and Type 2 verbs never co-occurred with instruments, whereas Type 3 and Type 4 verbs did, when they also co-occurred *Control* patient nouns.

The second notable fact about the instrument nouns was that the specific instrument that occurred in each sentence was dependent on the specific verb **and** the semantic category of the patient noun. Neither the verb alone nor patient semantic category alone determined the specific instrument noun, only both in combination. For example, “with fertilizer” occurred only with a VEGETABLE and the verb *grow*; “with insecticide” occurred only with a FRUIT and with the verb *spray*; “with vinegar” occurred only with a VEGETABLE and the verb *preserved*; ”with dehydrator” occurred only with a FRUIT and with the verb *preserved.* Of critical importance, while each instrument occurred with one and only one verb-patient combination, the verbs and the patient nouns co-occurred with multiple different instruments. As a result, correctly learning which instrument went in each sentence required learning the joint distribution of verbs, patient nouns, and instrument nouns.

5.3.2 Experimental Design

The 24 sentence types (semantic structures) that can be created from the rules described above define the 2x2 experimental design. The first factor was *Verb Specificity*: was the verb *category specific* (co-occurring with only one patient category) or was it *shared category* (co-occurring with two patient categories)? This factor is used to show that it should be easier to learn the instrument relationships for category specific verbs (Type 3 verbs) than for shared category verbs (Type 4 verbs). In other words, knowing the verb was ”grow” (a category-specific verb) is enough to know that the instrument, if there is one, must be ”fertilizer”. In contrast, for ”preserve” (a shared category verb), you need to also know if the patient was a FRUIT or VEGETABLE to know if the instrument should be ”vinegar” or ”dehydrator”. The second factor was *Patient-Instrument Co-occurrence*: did the patient noun co-occur with instruments during training? *Control* nouns did, and *Test* nouns did not, co-occur with an instrument during training. Thus, *Test* nouns co-occurring with Type 3 and Type 4 verbs have specific required instrument relations that can be learned, with perhaps this being easier to learn for Type 3 (category specific) verbs than Type 4 (shared category) verbs.

The critical test with regard to the ability of the models is how they perform on *Test* patient nouns. These are nouns that never occurred with an instrument in the corpus, with any of the verbs. Thus there is no direct evidence that any instrument is licensed. However, any model capable of learning the compositional structure of the language - the specific verb-patient-instrument pairings, might then be capable of showing generalization to these unseen items. For example, consider if the model learned that “vinegar” was the correct instrument for “preserve” paired with a VEGETABLE (based on seeing ”preserve cucumber with fertilizer” and “preserve potato with fertilizer”. Such a model might then generalize that, if “preserve pepper” was going to co-occur with an instrument, then the best instrument would be ”vinegar”. This would be true, despite that fact that ”preserve pepper” always occurred without any instrument in the corpus.

5.3.3 Corpus Generation

I generated 30 pseudo-random corpora using the semantic structures described above. Each of these 30 corpora had 400 blocks. Each of the 30 corpora were generated by first generating one example of each of the 576 possible legal sentences (all combinations of each agent-patient-instrument with each verb). Next, another 388 blocks of 48 sentences (one sentence for each verb) were generated, where the arguments in each sentence (agent, patient, and instrument) were selected randomly from among the legal options (by the verb). This was done to introduce some variability into each corpus variant, in order to test the robustness of models learning from the corpora. Each resulting corpus had 19200 sentences, with each verb occurring an equal number of times throughout the corpus.

Distributional Structure

The distributional structure needed to predict the correct instrument is best understood from the point of view of individual verbs and their possible arguments. For an illustration of the distributional structure of the instrument *preserve*, see Figure 5.3. In this example, there are four possible instruments for *pepper*, and the choice depends on a number of factors in addition to the choice of verb. For example, when the patient is a member of the FRUIT category, the instrument must be *dehydrator*. In contrast, when the patient is a member of the VEGETABLE category, the instrument must be *vinegar*. Further, the instrument cannot be decided based on knowledge of the patient alone. Each patient can occur with exactly 2 instruments; while members of FRUIT can occur with *insecticide* and *dehydrator*, members of VEGETABLE can occur with *vinegar* and *fertilizer*. This symmetry was intentional: The fact that there are more than one plausible choices when considering the verb or patient alone makes the task difficult for models relying exclusively on one lexical cue.

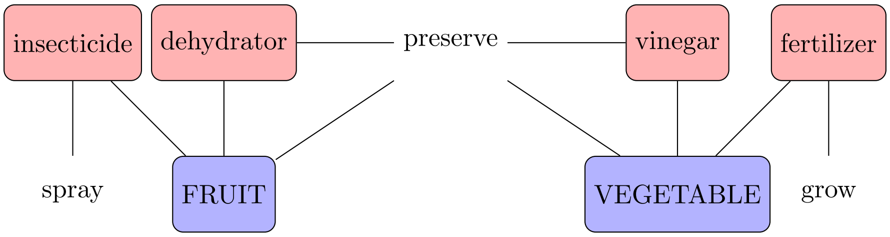


Figure 5.3: A schematic demonstration of the co-occurrence relationships between all instruments that can occur in sentences with the type-4 verb *preserve*. For reference, FRUIT consists of *raspberry*, *strawberry*, and *orange*, and VEGETABLE consists of *cucumber*, *potato*, and*pepper*. Notice the symmetry between the two patient categories: The verb *preserve* is related to both categories in identical fashion.

How to solve the Task

The most difficult version of the task, which requires compositional generalization to solve, is when the patient is an *Test* patient. Recall that an *Test* patient never occurs with an instrument in the same sentence in the training corpus. An example input is the verb-patient combination *preserve pepper*. Because *pepper* is a *Test* patient, there is neither a holistic dependency between the verb-patient combination and any instrument, nor a word-word dependency between the patient and any instrument. As a result, a model cannot infer the most plausible instrument (i.e. *vinegar*) directly from the training data. Instead, it has to identify a proxy phrase from which to generalize. This proxy phrase must be similar to the *Test* verb-patient combination, *preserve pepper*, and must have been observed alongside an instrument. The best candidate proxy phrases given the semantic structure of the corpus are *preserve cucumber* and *preserve potato*. The reason is that both phrases share the same verb as the *Test* verb-patient combination, and their patients are members of the same category, namely VEGETABLE. Once the proxy phrase has been identified, the model may generalize by inferring that the most semantically plausible instrument is the one that was most frequently observed with the proxy phrase in the training corpus.

How might a model identify a proxy phrase? The answer is that a model must have some mechanism for computing the similarity between two phrases. I argue that phrasal similarity must be computed compositionally, when no compound cues (i.e. cues at the level of phrases) are available. In the current example, that is the case. The reason is that *preserve pepper* only occurs with agents in the training corpus, and all agents occur equally likely with all verbs and patients. Thus, there are no distributional cues to allow a model to infer which other phrases the verb-patient combination is related to.[3] Therefore, to obtain the phrasal similarity, a model must break down the verb-patient combination into smaller components, namely a verb and patient. Given that there are a number of VPs that share the same verb, the model can infer phrasal similarity based entirely on the patient. To do so, the model needs only to encode the distributional semantic similarity among words, which is provided in the structure of the artificial corpus.

Once the proxy phrase has been identified, there are still a number of steps for the model to complete in order to arrive at the correct answer: As mentioned in Section 5.1, the model must also have encoded the phrasal dependency between the proxy phrase (e.g. between (*preserve cucumber* and *vinegar*) during training, and, finally, transfer this learned dependency to the *Test* verb-patient combination (i.e. inferring that *vinegar* is the most semantically plausible instrument for *preserve pepper*).

5.4 Experiments

All of the experiments reported in this work follow the same basic sequence of steps: (1) I trained 30 instances of each kind of model, one on each of the 30 randomly-generated instances of the artificial corpus, (2) I used each of these models to generate a set of prediction scores for each instrument in the corpus, given the sentence context, and (3) I used those prediction scores to evaluate each model's average performance across all 30 corpora. Success on this task tested the ability of each model to learn and represent distributional information, and use that information to judge of semantic plausibility of test sentences.

For each trained model, I computed the activation of all 32 instruments for each given sentence context. To compute this score in the CTN and LON, the lexical relatedness scores between the verb and instrument noun and between the patient noun and instrument noun were computed separately, and then combined via multiplication, in accordance with formula 5.4. In this way, both constituents of the verb-patient combination contribute independently to the composite relatedness — in accordance with compositional generalization.

A model's performance on each of the verb-patient combinations was based on whether the highest scoring instrument was the instrument that was structurally most licensed to occur with the given verb-patient combination. For example, given the input *preserve pepper*, the model would only be judged as correct if the model scored the relatedness of the combination *preserve pepper-vinegar* higher than any other verb-patient-instrument combination. The overall performance of a model in a given experiment was the average percent correct over all verb-patient combinations included in that experiment.

5.4.1 Experiment 1: Learning and Generalizing Word-Word Dependencies

In Experiment 1, I evaluated the ability of each model to predict the most structurally-licensed instrument, given contexts in which the instrument can be perfectly determined from a two-way lexical dependency. The cases where this was true was for sentences containing verbs that were "Category Specific", or Type 3 verbs, such as *grow*-VEGETABLE and *spray*-FRUIT (see Figure 5.3. In these cases, inferring the structurally licensed instrument only required identification of the verb, and recalling which instrument occurred with that verb in the training corpus.

[3] In natural language, it is possible to use extra-sentential cues to support phrasal similarity. However, in this artificial corpus, there are no dependencies across sentences that could be used to support phrasal similarity.

Experiment 1a

In Experiment 1a, I tested the models' ability to make correct predictions given the verb and a *Control* patient noun, i.e., a patient nouns that co-occurred with an instruments in the corpus. In Experiment 1b, I examined the extent to which the models were able to generalize learned two-way lexical dependencies to novel expressions, by assessing their ability to predict the most licensed instrument for sentences involving *Test* patient nouns. In the training corpus, *Test* patient nouns were nouns that never, under any circumstances, co-occurred with an instrument. Thus, Experiment 1b tests the ability of the models to generalize.

The purpose of this first experiment was to test whether learning of the observed verb-patient-instrument combinations has taken place. To accomplish this, I computed a score for each instrument that was the semantic relatedness between the verb and that instrument, multiplied by the semantic relatedness of the patient noun with that instrument. The model's prediction was scored as correct if the structurally licensed instrument was scored the highest, and report the average hit rate (i.e. accuracy) across all corpus seeds. Perfect performance is therefore obtained only if a model is able to consistently pick out the correct instrument from all other instruments, for each of the 32 tested verb-patient combinations, and all 30 corpus seeds. Lastly, I computed the accuracy of a baseline model that generated relatedness scores via random guessing.

Experiment 1b

In Experiment 1b, I adopted the same procedures as Experiment 1a, except for using as input verb-patient combinations with type-4 verbs and *Test* patients. Because I only allowed phrases with *Test* patients, there were half as many inputs as compared to Experiment 1a, namely 16. The question I aimed to answer in this portion of Experiment 1 is the following: Given that a model accurately captured a two-way lexical dependency (e.g. *grow cucumber* should be followed by *fertilizer* due to the verb *grow*), can it generalize this knowledge to a verb-patient combination that it has not observed during training? For instance, can the model also infer that *grow pepper* should be followed by *fertilizer*, despite never having seen that verb-patient-instrument combination before? This generalization is based entirely on a two-way lexical dependency, and therefore does not require compositional generalization. Because *grow* is always followed by the instrument *fertilizer* in the corpus regardless of the patient, this task does not require identifying substitutable expressions via similarity-based inference. As before, I recorded a hit only if a model scored the correct pair to be semantically most related (e.g. if the pair *grow pepper-fertilizer* is scored highest among all other tested verb-patient-instrument combinations). The overall hit rate was averaged across corpus seeds, and compared against a random baseline.

Results

The results of Experiment 1a and 1b are shown in the first two columns of Table 5.2. All models performed well above the random baseline (guessing resulted in an accuracy of 0.03). Notably, the CTN achieved perfect performance in both conditions. This means that the CTN was able to learn the lexical dependency between the verb and the instrument, and to successfully generalize the learned dependency to *Test* verb-patient combinations.

In contrast, the LON performed poorly in both conditions. Follow-up analyses of the output of the LON revealed that the model consistently confuses between the two instruments that co-occurred with patients in the same semantic category. For instance, given the type-3 verb *grow*, the LON predicted that *vinegar* and *fertilizer* are equally plausible, despite that only the former (*vinegar*) but not the latter (*fertilizer*) directly co-occured with *grow*. That said, *fertilizer* is not a bad inference, given that this instrument regularly co-occurs with the same set of patients that also co-occur with *grow*, namely *potato*, *cucumber*, and *pepper*. All three patients belong to the category VEGETABLES. This explains why the accuracy of the LON was well above chance, but nonetheless capped around 0.5. The failure to keep track not only of the co-occurrence between patients and instruments (e.g. *cucumber-fertilizer*) but also the patients that co-occur with them (e.g. *grow-cucumber-fertilizer*), is precisely the kind of mistake expected from a model that does not explicitly represent phrasal dependency (i.e. the dependency between larger chunks of language and a lexical item). This limitation can be overcome by leveraging constituent structure, which likely played a crucial role in the success of the CTN in this experiment.

Table 5.2: Accuracy of inferring the structurally-licensed instrument for verb-patient combinations with *Control* (e.g. *cucumber*) and *Test* (e.g. *pepper*) patients. Accuracies are averages across 30 corpus seeds.

	Experiment 1		Experiment 2	
	(a) control patient	(b) exp. patient	(a) control patient	(b) exp. patient
CTN	**1.00**	**1.00**	**1.00**	**1.00**
LON	0.51	0.51	0.49	0.47
Random Guessing	0.03	0.03	0.03	6.72×10^{-5}

5.4.2 Experiment 2: Learning and Generalizing Phrasal Dependencies

In Experiment 1, the dependency between a type-3 verb-patient combination and the correct instrument can be reduced to a two-way lexical dependency. In Experiment 2, however, I used verb-patient-combinations with type-4 verbs instead. An example of a type-4 verb is *preserve*, because it co-occurs with patients from two semantic categories instead of one (Figure 5.3). By evaluating inputs with type-4 verbs only, I effectively evaluated the ability of models to learn and generalize the fact that the correct instrument depends not only on the verb, but also on the patient.

Experiment 2a

In Experiment 2a, I focused only on verb-patient combinations with type-4 verbs and *Control* patients. For these types of verb-patient combinations, the correct instrument can be inferred by recalling which instrument most frequently co-occurred with the tested verb-patient combination in the training corpus. I used the same procedure for evaluation as mentioned above; an inference was considered correct if the relatedness between a tested verb-patient combination and the structurally licensed instrument was greater than for all remaining instruments. For instance, given the verb-patient combination *preserve cucumber*, the model must assign the highest semantic relatedness score to the pair *preserve cucumber-vinegar*.

Experiment 2b

In Experiment 2b, I investigated to what extent a model is able to transfer its knowledge of observed phrasal dependencies to *Text* phrases, which do not co-occur with any instrument in the training corpus. For example, if a model learned that *preserve cucumber* should be followed by the instrument *vinegar*, can it generalize this knowledge to the *Test* verb-patient combination *preserve pepper*? In this situation, *pepper* is a sibling of *cucumber*, because both are considered members of the VEGETABLE category and share contexts. Despite never having observed *preserve pepper* and *vinegar* in the same sentence, can the CTN use this knowledge to infer the correct answer, namely *vinegar*?

The best-fitting (i.e. structurally licensed) instrument is the one that satisfies two structural constraints: (i) The instrument must co-occur with the tested verb in the training corpus (due to a probabilistic lexical dependency between, say, *preserve* and the best-fitting instrument *vinegar*), and (ii) the instrument must co-occur with patients that belong to the same semantic category as the *Test* patient (e.g. *pepper* is the sibling of *cucumber*, and *cucumber* co-occurs with the best-fitting instrument *vinegar*).

Results

The results of Experiment 2a and 2b are reported in the last two columns of Table 5.2. Notably, the CTN achieved ceiling-level accuracy in both conditions. This means that the CTN learned all phrasal dependencies present in the corpus, and correctly generalized these dependencies to the VPs with *Test* patients. It should be noted that achieving perfect score in this task is not a simple feat (see Chapter 6). Similar to Experiment 1, the LON performed substantially worse than the CTN. This is further evidence for the important role that constituency structure plays in compositional generalization to novel phrases. The performance of the LON in Experiment 2 is very similar to that in the previous experiment. The reason is the same as explained in the previous section. The LON is making precisely the same mistake. In fact, the LON did not achieve accuracy significantly higher than 0.5 in any of the experiments. For that reason, I will focus on the CTN in the next two experiments, discussed below.

5.4.3 Experiment 3

Due to the excellent performance of the CTN in Experiment 2b, I asked whether the CTN can be pushed even further. In particular, I asked: Can the CTN use its ability to perform compositional generalization to infer the graded semantic relatedness between instruments and verb phrases? Put differently, can the CTN also infer which instrument is the next best fit after having determined which instrument is the overall best fit? A graded similarity should arise due to the hierarchical semantic structure of patients (see 5.3). To do so, I proceeded as in Experiment 2b, but modified the evaluation to also consider which instrument a model scores in second position (i.e. rank 2).

The determination of the second-best fitting patient is a bit more involved than the best-fitting instrument. Given the distributional structure of the training corpus, it is decided that there are two instruments that are equally plausible in the second position. As long as a model infers that both instruments are less plausible than the best-fitting instrument, and more plausible than all remaining instruments, a hit is recorded. Each of the two instruments highlights a separate, but equally important, aspect of semantic relatedness. Whereas one aspect emphasizes the distributional structure of the verb, the other choice highlights the distributional structure of the patient. For instance, consider the verb-patient combination *preserve pepper*, which was used in Experiment 2b. When considering the verb *preserve*, the second-best fitting instrument is *dehydrator*, given that it frequently co-occurs with *preserve*. On the other hand, when considering the patient *pepper*, the most related instrument is the one that most frequently co-occurs with siblings of *pepper*. The siblings are *cucumber* and *potato*, and each frequently co-occur with the instrument *fertilizer* (due to the verb *grow*). As a result, it did not matter whether a model scored *dehydrator* or *fertilizer* higher than the other; a hit was considered if the two instrument were scored in second and the third place.

Results

The results of Experiment 3 are shown Table 5.3. I separately report the accuracy at inferring the best fitting, and second-best fitting instrument. Accuracy at 'rank 1' and 'rank 2' refers to the accuracy of inferring the best-fitting, and second-best fitting instrument, respectively. Again, the CTN achieved ceiling-level performance under each type of evaluation method, while the LON lagged behind substantially. While the LON is considerably more accurate than a random-guessing baseline, the LON consistently confuses two instruments, as can be inferred by the fact that accuracy is at 0.5.

Table 5.3: Accuracy of inferring the structurally-licensed instrument, separated by rank. Rank 1 accuracy is the accuracy at inferring the best fitting instrument; rank 2 accuracy is the accuracy at inferring second-best fitting instrument. Accuracies are averages across 30 corpus seeds. Note: I report the best accuracy across all parameters for each model separately. In particular, the LON achieved its best performance when the preposition *with* was omitted from sentences that include instruments. The CTN achieved ceiling-level performance regardless of whether the preposition *with* was included.

	Experiment 3		Experiment 4	
	rank 1	rank 2	rank 1	rank 2
CTN	1.00	1.00	1.00	1.00
LON	0.50	0.51	0.53	0.53
Random Guessing	0.03	0.03	6.72×10^{-5}	6.72×10^{-5}

5.4.4 Experiment 4

In Experiment 2b, the tested verb-patient-instrument combinations (e.g. *preserve pepper vinegar*) do not occur in the training corpus. As a result, the combination was considered to be novel. However, it is possible that the mere fact that an *Test* verb-patient combination occurs during training — despite only being followed by the relatively uninformative period symbol — provides some information the model can rely on to achieve a high score. To ensure that this is not the case, and to align our evaluation task to other commonly used task for evaluating compositional generalization (e.g. the SCAN task introduced by B. Lake and Baroni (2018)), I modified the corpus by removing all sentences that contained a verb-patient combination with a type-4 verb and an *Test* patient. For example, *John preserve pepper* was excluded, because *preserve* is a type-4 verb, and *pepper* is an *Test* patient. If the CTN relies on the occurrence of such phrases to perform compositional generalization, their exclusion should reduce its performance in this experiment.

Results

The results are shown in Table 5.3. The perfect accuracy achieved by the CTN suggests that the model does not need to observe a verb-patient pair in the training corpus in order to infer the most plausible instrument for that pair. This finding is important because it presents the model with more naturalistic — and also more challenging — input, and raises our confidence that the model is indeed computing relatedness in a compositional manner as opposed to relying on pre-existing cues in the training data. These results also suggest that the CTN is relatively robust against variations in the training data, and especially the absence of direct cues about which verb is most likely to co-occur with which patient. Other types of models, such as those based on neural networks are often found to be less robust against such variation. In contrast, the LON achieved an accuracy very similar to that in all previous experiments. Again, this finding further strengthens the idea that encoding constituent structure can be relevant to compositional generalization.

5.5 Discussion

In this chapter, I explored CTN's capability of learning and generalizing multi-way lexical dependenceis. The goal was to enable structured inference at both word and phrasal levels without compromising performance in either. To evaluate the approach, I developed an artificial corpus and a task for measuring the capacity to infer the semantic plausibility of novel combinations of familiar items from the corpus. To do well on that task, a model must be able to generalize from known lexical items to new complex expressions — a skill that is often termed compositional generalization.

The results provide strong evidence that a distributional model can learn to solve a difficult semantic inference task that requires compositional generalization. In particular, it is observed that a system that represents both distributional statistics and constituent structure in a graphical data structure, can (i) learn to represent the semantic structure of the corpus it was trained on, and more importantly, (ii) make inferences about novel combinations of familiar lexical items.

The task used to evaluate the ability of CTN requires inferring which of a set of lexical items is a semantically more plausible instrument for a verb phrase that has never previously been observed alongside any instrument in the training corpus. While this task appears simple on the surface, it is actually a complex combination of multiple semantic sub-tasks. To achieve strong performance, a model must (i) learn the sequential dependency between a phrase and a lexical item (e.g. *vinegar* often comes after the verb phrase *preserve cucumber*), (ii) learn the semantic similarity between individual lexical items (e.g. *cucumber* shares distributional features with *pepper*), (iii) infer phrasal similarity from multiple word similarities (e.g. *preserve cucumber* is similar to *preserve pepper*, and (iv) generalize a learned phrasal dependency to a distributionally similar phrase (e.g. *vinegar* is likely to come after *preserve pepper*). Success in this task is a strong indicator that a model can judge the semantic plausibility of novel combinations of familiar items. The development of a system that is able to productively re-combine lexical items and phrases based on limited experience with a finite vocabulary is a critical step towards accounting for the productivity of the human language system.

In the remainder of this section, I (i) review the reasons potentially underlie the success of the CTN in the experiments, (ii) discuss the implications of the results to theories of semantic representation and human cognition, and (iii) briefly motivate comparison between CTN and other distributional models.

5.5.1 Ingredients for Compositional Generalization

The CTN achieved ceiling performance in all the experiments, including those that examined compositional generalization. I argue that the model's success should be attributed to the following three ingredients: First, 'connecting language' made it possible to transform linguistic data into a semantic network. Second, the spreading-activation algorithm provides strong estimates of semantic relatedness by integrating multiple distributional factors encoded in the CTN's graphical data structure. Third, the preservation of the constituency structure in the network proved critical for generalization. I discuss each of these three ingredients in more detail below.

Connecting Language

The merits of the ‘connecting language’ approach are multi-fold. First, it allows words and phrases from different expressions to be connected together in one graph. These connections may be direct or indirect, and the proximity between two expressions in the resulting graph is an important cue to their semantic relatedness. Another benefit of this approach is that all expressions participate during inference in the same way, no matter how distantly connected they are in the graph. The fact that two expressions that did not co-occur in the training corpus become indirectly linked in the semantic network makes it possible for spreading-activation to reach a node representing one expression from the node representing the other. For example, although *pepper* and *vinegar* do not co-occur in our corpus, the CTN is able to establish a connection between these two items on account of their shared co-occurrence with *preserve*.

It should be noted that the CTN connects expressions in natural language instead of concepts or more abstract representations of natural language, such as association norms. The preservation of the raw language input allows the modeler to develop custom procedures for extracting abstract relations on their own terms, and to consider the task and context in doing so. As a result, the CTN can be extended to consider the task and context during inference by adaptively modifying its core procedure for inferring semantic relatedness. For example, we might differentially weight different distributional factors that contribute to the overall semantic relatedness score depending on the task and context. Had the network been built from more abstract representations of language, this would be more difficult to achieve.

Spreading Activation

While ‘connecting language’ prepares the graphical structure for inference, the spreading-activation algorithm is the computational powerhouse for quantifying semantic relatedness. Quantification of semantic relatedness is sensitive to three distributional factors, namely (i) shared co-occurrence, (ii) co-occurrence distance, and (iii) co-occurrence frequency. Each of these factors is encoded in the graph, and by traversing the graph, spreading-activation is implicitly influenced by them. By integrating across these factors, spreading-activation can provide a graded measure of semantic relatedness.

A key factor in the success of spreading-activation for measuring semantic relatedness is the algorithm’s sensitivity to the number of shortest paths between two nodes. For example, in the artificial corpus, *pepper* is distributionally more similar to *cucumber* than to *berry*. This difference in distributional similarity is captured in the graph in terms of the number of paths between the nodes that represent those items (see Figure 5.4). The reason that there are more paths between *pepper* and *cucumber* is because they frequently co-occur with the same three verbs. In contrast, *berry* only co-occurs with one of those three verbs. As a consequence activation can spread more effectively from *pepper* to *cucumber* than from *pepper* to *berry*. In turn, this explains why *pepper* more strongly activates *vinegar* compared to *dehydrator* in Experiment 2b — only nouns in the VEGETABLE category co-occur with *vinegar*, and only nouns in the FRUIT category co-occur with *dehydrator*.

As noted before, ‘connecting language’ and spreading-activation together are insufficient to perform compositional generalization in the setting. The LON, a degenerated version of the CTN that lacks constituent structure, consistently failed to distinguish between the two most plausible instruments in Experiment 2b. I discuss the importance of constituent structure in the section below.

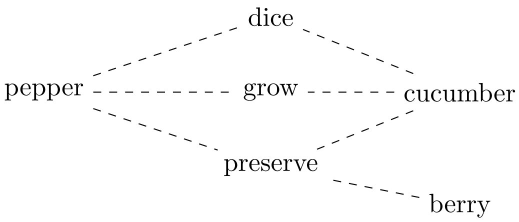


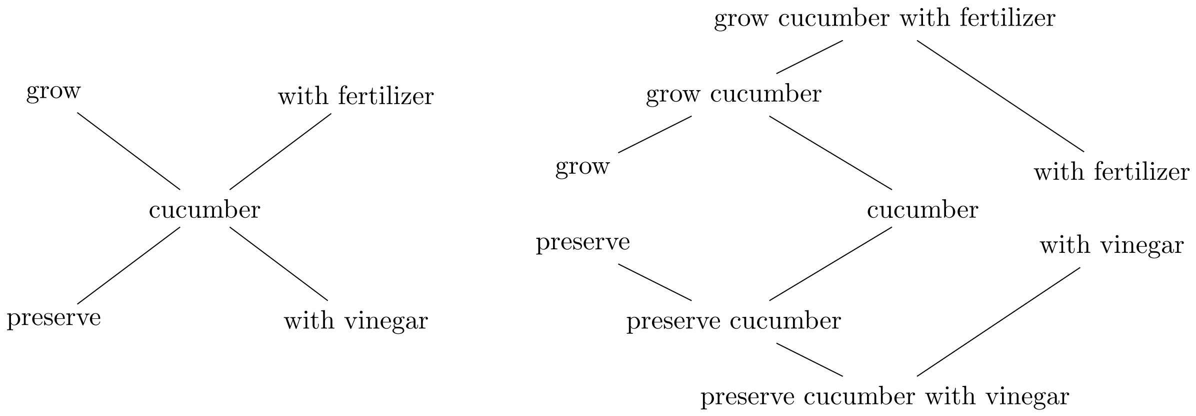
Figure 5.4: Connectivity between *pepper* and *cucumber*, and between *pepper* and *berry* in the CTN trained on the artificial corpus. Each dashed line represents an indirect path with phrasal nodes as intermediaries. The items *pepper* and *cucumber* are both members of the semantic category VEGETABLE, and this shared category membership manifests in the corpus in terms of shared co-occurrence with three verbs related to actions commonly performed on vegetables. In contrast, *berry*, which is a member of FRUIT only co-occurs with one of those verbs.

Constituent Structure

What the failure of the LON in the setting has demonstrated is that merely being equipped with a composition function is not sufficient to actually perform compositional generalization. The topology of the graph on which relatedness is computed must also be considered. Although both the LON and CTN define phrasal relatedness as the product of lexical relatedness scores (Equation 5.5), only the CTN was able to correctly infer the thematically best-fitting instrument for a given verb phrase in all the experiments. The reason is that, in the LON, nodes corresponding to sequentially occurring lexical items are connected linearly, and once those linear chains are joined, the ability to recover phrasal dependency is lost (Figure 5.5a). For instance, the nodes that represent the instruments *fertilizer* and *vinegar* become equidistant to *grow* and *preserve* in the graph. Even after composition by multiplication (i.e. application of Equation 5.5), the phrase *preserve cucumber* is no more related to *vinegar* than it is to *fertilizer*.

In contrast, phrasal dependency in the CTN is supported and preserved by the presence of phrasal nodes in the graph. These phrasal nodes are a result of the constituency parsing applied to all sentences input to the CTN. A phrasal node performs the critical function of linking two or more constituents that belong to the same phrase. After parse trees in a corpus have been joined, verb-instrument pairs are structurally preserved by virtue of being connected to the same phrasal node. The result is a graph where activation from *preserve cucumber* can more easily spread to *vinegar* than to the less plausible instrument *fertilizer*. These results highlight the importance of constituent structure in capturing the semantic plausibility of novel and complex expressions, and provide evidence that the CTN is performing semantic operations in a compositional fashion.

Constituent structure can be encoded in a variety of ways within a model. The key aspect for the CTN lies in its encoding method that promotes compositional generalization. By explicitly representing constituent structure in a graphical form, the CTN captures complex relationships between linguistic elements and effectively leverages these relationships for compositional generalization. Although other distributional models might also encode constituent structure (Padó & Lapata, 2007), they may not do so in a manner that directly supports compositional generalization. The CTN's encoding strategy allows the model to integrate syntactic information explicitly for the purpose of structured inference, facilitating the generalization of learned patterns to novel combinations of lexical items. Consequently, the CTN's ability to encode constituent structure in a manner that fosters compositional generalization is a crucial factor in its success as a distributional model.



(a) LON

(b) CTN

Figure 5.5: Comparison between LON and CTN . (a) The network structure of the LON formed by joining word-chains. (b) The network structure of the CTN formed by joining the constituent trees.

5.5.2 The Relationship between Compositional Generalization and Productivity

The CTN follows in the footsteps of many prior works that have modeled how humans might represent semantic knowledge (Anderson, 1983; Landauer & Dumais, 1997; Lund & Burgess, 1996; Jones & Mewhort, 2007; Kenett et al., 2011; De Deyne et al., 2016; Rotaru et al., 2018; McRae et al., 1997; Baroni, Bernardi, & Zamparelli, 2014; Collins & Quillian, 1969; Collins & Loftus, 1975; Osgood, 1952; Nelson et al., 2004; Miller, 1992). However, few works have attempted to explain how their models might also account for the productivity of the human language system, i.e. the ability to form useful representations of novel combinations of familiar lexical items.

I argue that in order to account for the full range of human semantic behavior, a model must be developed with the productive nature of human language in mind. A language system is considered to be productive when it is able to combine linguistic units from a finite set to form novel meaningful expressions with virtually infinite variety (Fodor & Pylyshyn, 1988). For instance, such a system would be able to generate the phrases *cut cake with knife* and *cut tree with ax*, and to distinguish them from other grammatical phrases such as *cut cake with ax* and *cut ax with cake* that are semantically less plausible. Thus, productivity not only refers to the ability to generate novel grammatical expressions, but also to the ability to infer which of those expressions are semantically more plausible than others (Kutas & Federmeier, 2011; McRae et al., 2005; Resnik, 1996), despite never having previously encountered them.

The CTN model was motivated not only to model aspects of human semantic memory, but also to account for the productive nature of human language. One way in which productivity is harnessed is judging the semantic plausibility of novel expressions. Given any finite corpus, there is an infinite number of expressions that are both grammatical and novel (i.e. do not occur in the corpus). This means that the distributional semantic properties of those novel expressions cannot be retrieved from memory. To address this, semantic properties must be computed on-the-fly, by accessing and integrating representations of smaller components of the novel expression. I consider that modeling semantic plausibility judgment is a promising new avenue for the field of distributional semantic modeling, precisely because it makes contact with the productive nature of the human language system. The CTN is one step in this direction. The CTN makes it possible to judge which novel expressions that are legal under some grammar are actually semantically plausible (relative to other expressions). It is difficult to imagine human cognition without this ability: Should a pie be cut with a knife or an ax? It would be inconvenient to delay a decision until we have heard *knife* and *pie* co-occur in the same sentence.

Interestingly, recently developed large language models (LLMs) appear to be quite proficient at combining parts of language to produce coherent and semantically plausible output. While recent breakthroughs such as GPT-4 (Bubeck et al., 2023) are able to converse on almost any topic without errors in semantic plausibility, it remains to be seen how such models perform on tasks explicitly designed to quantify compositional generalization as how it is done in this chapter. On first glance, we expect the performance to be strong. This raises the following question: What benefit, if any, do more structured graphical models like the CTN offer relative to LLMs? One suggestion is that if both LLMs and models like the CTN are found to similar outputs in standardized tasks, models in the latter class would be useful for researchers studying the inner workings of LLMs. More structured graphical models, therefore, present a valuable tool for opening the black box that powers recent advancements in language technology. I expand on this topic in the next part of the discussion.

5.5.3 Limitations and Future Directions

Due to the novelty of the CTN model, much more work is needed to contextualize them with the broader psycholinguistics literature. This is not only the case for CTN, but also for LON and Distributional Graph in general. Therefore, I leave this whole discussion to Chapter 7. Relatively more important for the CTN model in particular, is to validate the findings (in this chapter) by comparing the model to other distributional models. This will be the topic of the next chapter, immediately following the discussion below.

Comparison to Other Distributional Models

One of the primary motivation for developing and evaluating the CTN is due to shortcoming of existing 'word-encoding' and 'sequence-encoding' models (will be defined in the next chapter) in the distributional semantic modeling framework. Both types of models struggle on difficult semantic tasks that require compositional generalization. While I have shown that the CTN can overcome many of those shortcomings, I am aware that the current work is incomplete due to a lack of comparisons between the CTN and existing models. The gist of the model comparison work aligns with many prior works on this topic, namely that sequence-encoding models such as the RNN and Transformer based models can, in some cases, achieve compositional generalization, but their performance is inconsistent, and might be far from perfect. Further, I found that the Transformer achieved much better results in the compositional generalization portion of our task than the RNN and word-encoding models. In fact, a miniature version of a Transformer based on GPT-2 (Radford et al., 2019) was the only model other than the CTN that achieved strong performance in the current setting.

A thorough analysis of these results are left for future works. Among the many unanswered questions are the contribution of constituent structure to successful compositional generalization. For instance, most distributional models do not encode constituent structure. That said, sequence-encoding models, such as an RNN or Transformer, are often able to infer constituent structure from raw language input (i.e. tokenized but not parsed text). If the rapid progress in the field of language modeling continues to the point where such models are able to succeed on difficult semantic tasks that require compositional generalization, what does this say about the need for explicit representation of constituent structure, such as in the CTN?

Altogether, both the CTN and Transformer[4] have demonstrated success in complex semantic tasks

[4] It should be noted that in our experiments, the Transformer required extensive hyperparameter tuning, and, while well above chance, performance sometimes fell far short of ceiling-level.

requiring compositional generalization. This has significant implications for cognitive theory and machine learning. Compositional generalization tasks evaluate a model's ability to encode phrasal dependencies by constructing compositional representations that facilitate generalization to novel phrases. The Transformer's success in this task suggests that it possesses these computational capabilities, which might explain its strong performance in various large-scale language tasks. However, the underlying mechanisms in LLMs are less transparent due to their complex architecture. In contrast, the CTN model has a more explicit structure and mechanism. Given their similar performance in compositional generalization tasks, the CTN could serve as an interpretive model for LLMs' semantic representations. I hypothesize that similar representations are formed in the Transformer despite differences in architecture and training schemes. Future research should explore the semantic representations within LLMs and compare them to those in the CTN. This line of inquiry could deepen our understanding of the inner workings of LLMs and, more broadly, the representations necessary for natural language understanding in humans. As a start, in the next chapter I examine a few existing distributional models against CTN, with the learning and generalization tasks used in this chapter.

Chapter 6

Multi-way Lexical Dependency

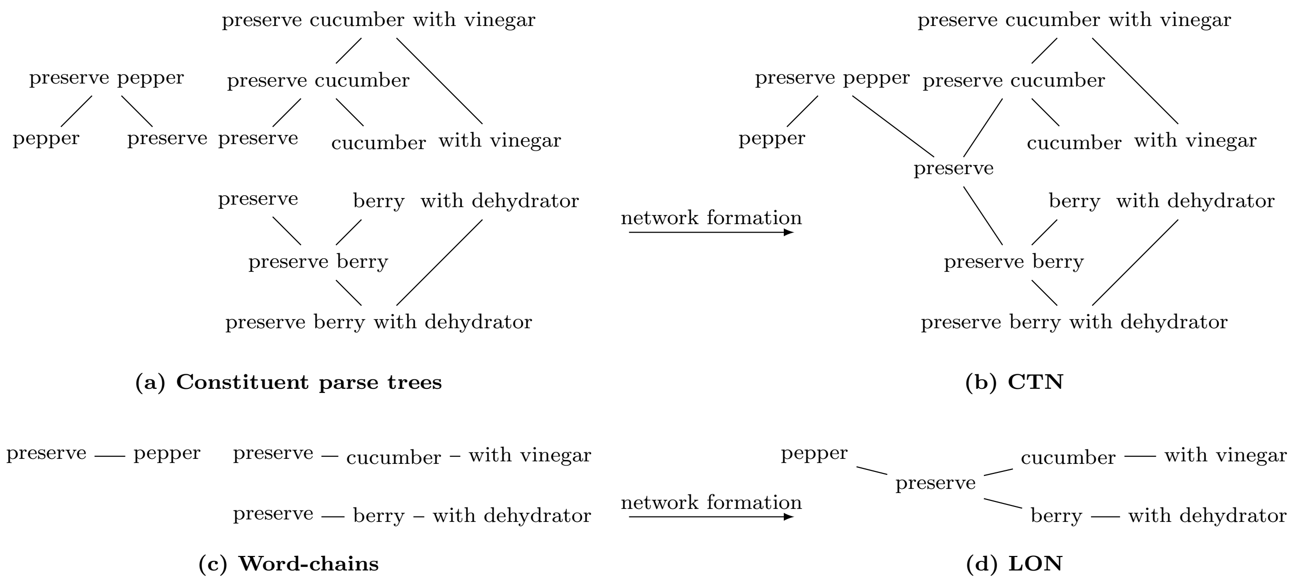
In this chapter, I investigate what mechanism and computational prerequisite can be the cornerstone toward the semantic achievement of learning and generalizing multi-way lexical dependencies. In Chapter 5, I approached the theoretical problem by concentrating on CTN and LON, the two types of distributional graphs focused in this dissertation. In this chapter, I expand the exploration to a broader range of distributional semantic models, by testing them on the learning and generalization tasks in Chapter 5 (with adaptions).

The chapter is structured as follows: First, I emphasize why the Constituent Tree Network (CTN) can perfectly learn and generalize multi-way lexical dependencies: It is based on established syntactic parsing, i.e., The CTN model works on the semantics after the syntax is fully processed. Relating to this advantage of CTN, I review existing distributional models that directly process on raw linguistic input, focusing on the challenges they may have to achieve in the hard semantic tasks. Then, I concisely describe the corpus and tasks to examine the models. They are almost identical to the method in Chapter 5, with minor adjustments for the purpose of model comparison. Next, I compare different sets of models in a series of experiments, to reveal the critical computational mechanism for the compositional generalization (generalizing multi-way dependency) task. By comparing the first two DSMs, I elucidate what contributes to the failure and success in learning multi-way lexical dependencies provided in the input. I then compare two types of neural net DSMs to explore the critical mechanisms toward compositional generalization on the multi-way relationships. Lastly, I compare the models that achieve in the critical generalization tasks and elaborate on the significance of learning and generalizing multi-way lexical dependencies, emphasizing on how the capability may lead to more semantic accomplishments.

6.1 Model Comparison

6.1.1 The Advantage of CTN

In the CTN model, the constituent structure is crucial to the success in representing multi-way lexical dependencies. The topology in the constituent trees specifies the inter-dependency between the verb, the patient and the instrument, which in turn can be accessed by the spreading-activation process once the trees are connected into one network. Alternatively, if the network is not built from trees, but from linear ordered chain of words (Figure 6.1c,d), the structure will fail to capture the multi-way dependencies (as showed in Chapter 5). In other words, the CTN model primarily benefits from the syntactic pre-processing on the raw text.



(a) Constituent parse trees

(b) CTN

(c) Word-chains

(d) LON

Figure 6.1: Formation of the network structure in the Constituent Tree Network (CTN) and the Linear Order Network (LON) given the mini corpus *'preserve pepper', 'preserve cucumber with vinegar', 'preserve berry with dehydrator'.* (a) The input to the CTN consists of constituency-parsed trees for sequences in the mini corpus. (b) The network structure of the CTN is formed by joining the constituent trees at shared nodes. (c) The input to the LON consists of word-chains, formed by connecting adjacent words in the mini corpus. (d) The network structure of the LON is formed by joining word-chains at shared nodes.

However, the syntactic prerequisite can be a major limitation of the CTN model, as constituent parse is not free: Transforming the sequential text input into clean parsed tree can be a sophisticated computational endeavor. From the implementational perspective, since the model does not directly process on raw inputs, it would be difficult to implement the model directly for real world language tasks. Moreover, from the psychological theory perspective, the reliance on constituent parse also constrain the model in terms of its account for semantic development. As a type of Distributional Graph, the CTN model can be considered as a model of long-term semantic memory in a more general sense. However, since the CTN does not operate on raw linguistic input, it may not speak to the semantic development in learners before mastery of syntactic knowledge. These critical issues of the CTN model lead to the exploration on a series of distributional semantic models that directly process raw texts.

When it comes to hard tasks such as generalizing on multi-way lexical dependency, distributional models working on raw texts face a 'formidable' challenge: They need to figure out the semantics and the syntax as the same time. Inspired by CTN's success, it is likely that achievement in this hard semantic problem requires a model to form some quasi semantic-syntactic structures. In the remainder of this section, I briefly review how different existing models of distributional semantics might struggle against the challenge. How the potential lack of one or more computational mechanisms may negatively impact a model's performance in compositional generalization. In particular, I discuss two major classes of distributional semantic models, namely word-encoding models, and sequence-encoding models. While each class of models comes with its own distinct set of advantages, I argue that neither class of models fully possesses sufficient computational capabilities for generalizing multi-way lexical dependencies (compositional generalization).

6.1.2 Word-Encoding Models Struggle to Perform Semantic Composition

Early distributional models primarily focused on representing individual words, in which their meanings are represented as vectors in a high-dimensional space and relatedness is computed using vector-space metrics such as cosine-similarity (Landauer & Dumais, 1997; Lund & Burgess, 1996; Mikolov et al., 2013). These models have been successful in many lexical tasks, and are referred to as word-encoding models in this chapter. However, it is not clear how such representations can be combined to represent meaning of larger expressions, such as phrases, clauses, and sentences (Mitchell & Lapata, 2010). Various functions have been used to combine lexical representations learned by word-encoding models, such as addition, multiplication, and averaging (Mikolov et al., 2013; Mitchell & Lapata, 2010), and more sophisticated functions such as circular convolution (Jones & Mewhort, 2007; Mitchell & Lapata, 2010) and linear map (Baroni, Bernardi, & Zamparelli, 2014). While these functions have enabled some success in modeling tasks which require combining individual representations, there is yet to be a unified agreement on which functions are best and under what circumstances. Additionally, many studies are guided by empirical results, and there is a lack of works attempting to provide a theoretically motivated argument as to why certain functions provide better results than others.

The representation of complex structure presents a challenge for using word-encoding models to generalize compositionally. Specifically, while these models are able to capture word-word similarity (e.g. *cake* and *pie* occur in similar linguistic contexts), they lack clear-cut procedures for constructing representations of complex expressions, such as phrases and clauses. This makes it difficult to compute the similarity between phrases (e.g. the relationship between *cut cake* and *cut pie*) as well as the thematic fit between a phrase and a word (e.g. *cut cake* and *knife*). As a result, researchers aiming to apply the concept of semantic relatedness to complex expressions using word-encoding models are often left without a generalizable solution, and must instead create custom solutions that may work well in certain contexts, but fail to generalize to others (Baroni, Bernardi, & Zamparelli, 2014; Mitchell & Lapata, 2010).

6.1.3 Sequence-Encoding Models May Entangle Co-occurrence Statistics at the Word and Phrasal Level

One way to overcome the limitation of word-encoding models is to train models to *learn* how to combine consecutively occurring lexical items without intervention from the researcher (Elman, 1990; Liu, Vulić, Korhonen, & Collier, 2021; Vulić, Ponti, Korhonen, & Glavaš, 2021). These models are usually neural network models trained to predict next-words given a sequence of words. We call such models ‘sequence-encoding’ models, to distinguish them from the more traditional ‘word-encoding’ models pioneered by Landauer and Dumais (1997) and Lund and Burgess (1996). By being trained to predict which words are likely to come next, such models learn implicit composition functions for how to combine words in their input.[1] These functions are encoded in the network’s weights, and are gradually tuned to the statistics of the training corpus. An advantage of sequence-encoding models over word-encoding models is that they excel at learning which words tend to go together in a sequence. For instance, *cut cake* and *knife* become related on account of the latter often occurring after the former in large corpora. Like word-encoding models, sequence-encoding models have been extremely useful in accounting for a broad range of behavioral phenomena in the psycholinguistic literature.

[1] It should be noted that not all sequence-encoding models predict which word is likely to come next. An alternative training strategy involves predicting words that have been masked by the researcher. Such models often make use of bi-directional context information to make predictions.

Despite their many strengths, many researchers have raised doubts about the ability of sequence-encoding models to generalize to novel, complex expressions. The primary reason for this is their difficulty in differentiating between a representation of a word and that of complex expressions such as phrases. Because the representation of a sequence can include an arbitrary number of words, and since there is no explicit mechanism for keeping track of individual words as the sequence is processed, simple and complex expressions are not clearly differentiated during processing. This hampers the ability of many sequence-encoding models to analyze each word independently from the others in the same sequence. This is especially problematic for recurrent neural networks, and less for the more recent Transformer-based language models. [2] As a result, sequence-encoding models might not be well equipped to identify small, reusable chunks of language. Without an explicit mechanism to do so, most sequence-encoding models cannot reliably perform lexical substitution. Recall that lexical substitution is a critical step for performing compositional generalization. In particular, substitution enables the identification of 'proxy' phrases, such as *preserve pepper* given *preserve cucumber* — by replacing *cucumber* in *preserve cucumber* with *pepper* to form *preserve pepper*.

In a few following sections, I examine word-encoding models (HAL, LON) and sequence-encoding models (RNNs, Transformer) to show what are the exact challenges for the models to achieve in compositinoal generalization. To start, I describe the evaluation and experiments that will be used to test the models.

6.2 Method

All models will be trained on the artificial corpus used in Experiment 1 and 2 of Chapter 5. Variations of the basic corpus will be used in particular experiments and will be described in corresponding sections. For each trained model, pairwise semantic relatedness scores between all verb-patient (VP) pairs and all instruments are computed. Across experiment conditions, the evaluation of a model is always based on how well it captures the 'expected' best VP-instrument fit, i.e. scoring the best fitting instrument for each VP. In other words, all experiments in this chapter will only consider the rank 1 instrument, as in Experiment 1 and 2 of chapter 5. Next, I describe the evaluation and the organization with more details. How each model is trained and how the VP-instrument semantic relatedness is computed in each model will be explained in respective model sections.

6.2.1 Evaluation and Experiments

Once a model is trained, semantic relatedness between the interested VP-instrument pairs are computed. There are 12 VPs and 4 instruments in each cluster (Table 6.1), making it 96 VPs and 32 instruments in the corpus. For each VP, I compute the semantic relatedness between the VP and all 32 instruments, and ranked the instruments by the relatedness scores. The model is considered as 'correct' on the rank concerning certain VP if it select the structurally licensed instrument for the VP (specified below).

For the VPs with instrument-associated patient, e.g. *grow cucumber* and *preserve cucumber*, the structurally licensed instrument is the one that has co-occurred with the VP in the corpus. Among the 96 VPs, 64 of them have instrument-associated patients. They are evaluated in the 'learning' experiment, in which I test whether the model can capture the VP-instrument association provided in the corpus. Half of these VPs

[2] In contrast to RNNs which forcibly compress lexical representations into one or more hidden layers across all time steps, the non-recurrent architecture of Transformers, combined with self-attention, allows for more flexibility concerning how and when lexical representations are combined.

involve in two-way dependencies, e.g. *grow cucumber*, and the other half involve in three-way dependencies, e.g. *preserve cucumber*.

The rest 32 VPs are instrument-vacant, e.g. *grow pepper*, *preserve pepper*. They are evaluated in the 'generalization' experiment. These VPs are not associated with any instrument in the corpus, and the task for the models is to infer the most plausible instrument for the VPs. In this case, the 'closest' instrument-associated VPs to the investigated VP would be the ones with identical verb and same-category patient, e.g. *grow cucumber/potato* for *grow pepper* and *preserve cucumber/potato* for *preserve pepper*. The structurally licensed instrument for the investigated VP would be the instrument associated with the 'closest' VPs, i.e. *fertilizer* for *grow pepper* and thus for *grow cucumber/potato*; *vinegar* for *preserve cucumber/potato* and thus for *preserve pepper*. As in the case of the learning condition, the generalization tasks can be further split into generalizing two-way dependency (*grow pepper - fertilizer*) and generalizing three-way dependency (*preserve pepper - vinegar*).

Table 6.1: Example sentences from the artificial corpus, for 2 patient categories only. Each category is associated with 4 types of verbs/sentences. Two types of verbs/sentences are used in 2-way and 3-way experiments, while the other two types are used to form semantic taxonomy on the patient categories.

Task	2-way		3-way		forming category
learn	J grow cucumber	w. fertilizer	J preserve cucumber	w. vinegar	J dice cucumber/ J ferment cucumber
learn	J grow potato	w. fertilizer	J preserve potato	w. vinegar	J dice potato/ J ferment potato
generalize	J grow **pepper**		J preserve **pepper**		J dice **pepper**/ J ferment **pepper**
learn	J spray berry	w. insecticide	J preserve berry	w. dehydrator	J dice berry/ J pick berry
learn	J spray apple	w. insecticide	J preserve apple	w. dehydrator	J dice apple/ J pick apple
generalize	J spray **orange**		J preserve **orange**		J dice **orange**/ J pick **orange**

In total, there are four experiment conditions: two-way learning (L2), three-way leaning (L3), two-way generalization (G2) and three-way generalization (G3). In each condition, a specific group of VPs are evaluated (32 two-way and three-way instrument-associated VPs in L2 and L3; 16 two-way and three-way instrument-vacant VPs in G2 and G3). A model get a 'hit', if it selects the correct 'expected' instrument for the particular VP. The performance of a model in a particular experiment condition is taken as the hit rate over all (16 or 32) VPs in the condition.

Intuitively, to perform well in the generalization task, the model need to first capture the two-way/three-way dependencies provided in the input, namely, succeeding in the learning conditions. Although I tested all models of interests on the four experiment conditions, I will focus on different conditions for different models. First, I examine two word-count models - Hyperspace Analog to Language, i.e., HAL (Lund & Burgess, 1996), Linear-Ordered Network, i.e. LON (Mao & Willits, 2020), with the focus on the learning task. By comparing the two count models, I explore the necessities for learning multi-way lexical dependency observed in the language input. Then I examine two neural net prediction-based models: LSTM (Hochreiter & Schmidhuber, 1997) and a miniature version of GPT-2 (Radford et al., 2019), primarily focusing on the generalization conditions. By comparing the two prediction models, I try to elucidate the mechanism for successful generalization. I also compare some of the models to CTN, to see how the models approximate the conceptual ideal without the syntactic prerequisite. With the scaffolding examinations of the models, I approach to the computational crux for learning and generalizing multi-way lexical dependencies.

6.3 HAL

The simplest type of distributional semantic models are trained to represent only words. They are typically based on counting raw co-occurrences and matrix-factorization, e.g. LSA (Landauer & Dumais, 1997), HAL (Lund & Burgess, 1996) and BEAGLE (Jones & Mewhort, 2007), and in these models, individual words are represented as continuous vectors in a high-dimensional "word embedding" space. The count based vector representation enables quantitative analyses of similarities and differences between how words pattern in language (Landauer & Dumais, 1997). Some of these models have been studied by the cognitive science community, especially for predicting human behavior in a variety of linguistic tasks (Baroni, Dinu, & Kruszewski, 2014; De Deyne et al., 2016; Evert & Lapesa, 2021), and have been suggested as plausible mechanisms for how human learners acquire the meaning of words (Mandera et al., 2017; Lupyan & Lewis, 2019). Hereafter, I refer to this type of models as count-vector models and specifically investigate the HAL model (in which words are represented by a vector of co-occurrence, in contrast to LON) as a representative of the type.

While HAL and count-vector models are useful in modeling word-word lexical dependencies, they may struggle to learn more complex three-way dependencies. Consider the inter-dependency between *preserve cucumber* and *vinegar* in the example. How could the model correctly select *vinegar* instead of *dehydrator* for this sentence? One way is to compose the vector representations of the verb and the patient noun, to form a vector representation of the VP, and then evaluate the semantic relatedness between the VP and the instrument. I refer to the approach as 'representation-composing'. Extensive work in identifying useful composition functions for word vectors has not yielded promising results so far (Mitchell & Lapata, 2010; Baroni, Bernardi, & Zamparelli, 2014). The test below adds support to this line of work. Alternatively, word-word semantic relatedness (verb-instrument, patient-instrument) can be computed first and the VP-instrument relatedness is then obtained from 'composing' the pairwise relatedness (relation-composing). However, as we will show, this 'relation-composing' approach does not work for this type of model either.

6.3.1 Model Training

In forming the word vector in HAL, multiple minor parameters may affect on the final representation and the performance in the downstream tasks (Bullinaria & Levy, 2007, 2012). I identified the best performing HAL model by tuning the parameters for 'window weight' (flat, **linear**), 'window type' (forward, backward, **sum**, concatenate), and 'number of singular dimensions' (16,22,24,... **30**,32,...,36,64). Bold-face indicates chosen parameters. The window size for tracking co-occurrences was constrained such that all words in a sentence are available for distributional analysis - other window sizes were not considered.

The VP-instrument semantic relatedness was computed in two ways. In the 'representation composing' approach, vector representations of VPs were formed, and the semantic relatedness between VP and instrument were computed as the cosine similarity between the VP and instrument vectors. The VP representations were derived by applying composition functions on the verb and theme noun vectors. In this chapter, I examined addition and point-wise multiplication. In the 'relation-composing' approach, the verb-instrument and patient-instrument relatedness were first calculated using cosine similarity. Then the VP-instrument relatedness were obtained by multiplying the two word-word relatedness. Using a more general formulation, I define the relatedness between the VP (verb,patient) and the instrument as

$$\text{SR}((\text{verb, patient}), \text{instrument}) = \text{SR}(\text{verb, instrument}) \cdot \text{SR}(\text{patient, instrument}) \tag{6.1}$$

in which SR(A, B) denotes the semantic relatedness between linguistic expression A and B. I tuned the parameters separately for the two approaches, and tested the three HAL models (one relation-composing model, HAL-relation, and two representation-composing models, HAL-multiplication for multiplication composition and HAL-addition for addition composition) on the two learning conditions.

6.3.2 Results

Table 6.2: Accuracy of HAL models inferring the structurally-licensed instrument in learning two-way (L2) and three-way (L3) dependencies. Accuracy are averages across 10 seeds.

	Learning	
	L2	L3
HAL-addition	0.27 (0.07)	0.25 (0.10)
HAL-multiplication	0.29 (0.07)	0.23 (0.10)
HAL-relation	0.86 (0.04)	1.00 (0.00)
Random	0.03	0.03

As shown in Table 6.2, all models performed above the random baseline. For the representation-composing approach, the performance were relatively low for both composition functions in either two-way dependency (addition: m=0.27, sd=0.07; multiplication: m=0.29, sd=0.07) or three-way dependency (addition: m=0.25, sd=0.10; multiplication: m=0.23, sd=0.10). In contrast, the relation composing approach performed a lot better in both two-way (m=0.86, sd=0.04), and three-way (m=1.00, sd=0.00) dependency.

It has been shown that SVD dimensionality may significantly affect the performances in HAL like models. Therefore, I further tested HAL-relation models on a broader range of SVD dimensionalities. I selected all even dimensionalities from 22 to 36, which were around 32, the dimensionality performed best in the tuning (only 16, 32 and 64 were involved in the initial tuning). I tested these HAL-relation models on both the two-way and three-way learning tasks. In short, the best model had dimensionality 30 and 32. 32-dimension outperformed 30-dimension in the two-way task, (32-dimension: m=0.95, sd=0.04; 30-dimension: m=0.86, sd=0.04), while 30-dimension outperformed 32-dimension in the three-way task (30-dimension: m=1.00, sd=0.00; 32-dimension: m=0.64, sd=0.10). Models with other dimensionalities performed a lot worse in either the two-way task or the three-way task, or both, compared to the best two models. I also tested the representation-composing HAL models on other SVD dimensionalities, and there were no signicant enhancement.

6.3.3 Discussion

I tested the HAL models on learning multi-way dependencies by computing the dependencies in representation-composing and relation-composing approaches. The representation-composing models failed to learn the two-way and three-way relations. For the relation-composing approach, the best performing model (with 30 SVD dimensions) successfully learned the two-way and three-way dependencies, while models with other dimensionalities failed in at least one type of the learning tasks.

The key to the representation-composing approach is forming a reliable holistic representation of the VP, so that the VP representation relates appropriately to the instrument. The results show that it is in general hard to find a effective compositional function that composes the verb and the patient noun. What I found in general align with earlier findings that deriving effective static embedding of complex expression as a composition of static word embedding remains a considerable challenge (Erk & Padó, 2008; Mitchell & Lapata, 2010). The difficulty lies in guaranteeing the effective word embedding as well as finding the 'right' compositional function that works on the word embedding.

On the other hand, the crux of the relation-composing approach is more fundamental: it is only about the word embedding. The result has shown that, with fine tuning, it is possible to find an HAL encoding (with the proper SVD dimensionality) that produces effective word embeddings: Two-way relations directly derived from the embedding and three-way relations directly composed from the two-way relations may capture the two-way and three-way lexical dependencies manipulated in the corpus. Nevertheless, while one model succeeded in the learning tasks, most SVD dimensionalities still failed. Furthermore, I found that the successful model variation (30-dimension) still failed in the two-way generalization task. In general, I found that the performance of each HAL model (varied by SVD dimensionality) is sensitive to the task. Such sensitivity can be a potential constraint for models like HAL which use static embedding.

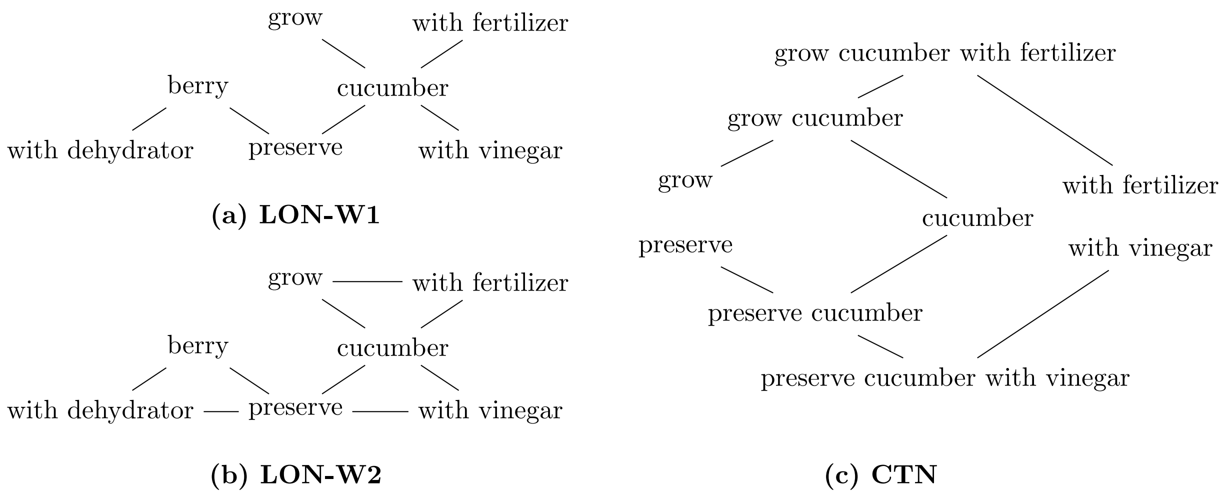
In comparison to the HAL model, I will show that other models may tackle the learning tasks through the two approaches respectively. To be more specific, the representation-composing approach is adopted in the recurrent neural networks (RNNs), and the relation-composing approach is applied more effectively in particular LONs. I leave RNNs for a bit later, and investigate the LON models in the next section. I will mention a critical computational characteristic lacked in HAL but present in the particular LON. This computational advantage helps LON to better catch the pairwise lexical dependencies for the first place. With the comparison between HAL and LON, I explain why HAL failed more in the learning tasks.

6.4 LON

Linear Order Network (LON) can be considered as the graphical version of HAL. Similar to HAL, it encodes word co-occurrence in the raw language input. But instead of taking the co-occurrence frequencies as vector dimensions, LON uses them to form links between adjacent words in a graph (Figure 6.2a,b). While the vectors in HAL can be composed to form phrasal representations, a similar operation does not exist for LON. That is to say, the representation composing approach is not legitimated in LON. Therefore, the three-way dependency can only be computed by composing the word-word relatedness in LON. In this way, the LON faces the same challenge as HAL, that is, to appropriately measure the word-word relatedness. To be more specific, the model has to represent appropriately the verb-instrument and patient-instrument contingencies stipulated in the corpus. As showed in the HAL section, the HAL model had trouble in both verb-instrument and patient-instrument dependency. I am interested in if LON (the graphical correspondence of HAL) can capture the two types of pair-wise dependencies and perform well in the learning task.

I test two LON models: LON-W1, the generic LON model with window size 1 (so that it only links the adjacent words in the corpus, Figure 6.2a); and LON-W2 that links word within a size-2 window (Figure 6.2b). I will show that while the graphical data-structure helps both LON models tackle the patient-instrument dependency, the verb-instrument dependency remains to be a challenge only for LON-W1 but not for LON-W2. This suggests that a larger window size helps in encoding useful information. However, with the a larger window size, the vector HAL model still failed to capture the verb-instrument dependency. I first explain how the two LON models may succeed or fail in learning the pair-wise dependencies and delve deeper into the critical mechanisms towards the computational capability at the end of the section.

First of all, the LONs generically captures the patient-instrument dependency by linking the words within



(a) LON-W1

(b) LON-W2

(c) CTN

Figure 6.2: Comparison between LON with different window sizes and CTN. (a) LON with window size 1 (LON-W1) is formed by linking (with an edge) immediate adjacent words. (b) LON with window size 2 (LON-W2) is formed linking words within a size-2 word window. (c) The network structure of the CTN is formed by joining the constituent trees at shared nodes.

the same sentence, e.g. in both LON-W1 and LON-W2, *cucumber* is closer to *vinegar* (the instrument associated to *cucumber* in the corpus) compared to *dehydrator* (the instrument not associated to *cucumber* in the corpus). However, the LON-W1 does not capture the verb-instrument dependency as it only links adjacent words, e.g. *preserve* does not distinguish between *fertilizer* and *vinegar* (Figure 6.2a). This is due to the linear encoding such that the verbs are connected to the instruments through the patient (as the patient is placed between them). In this way, the linear graphical structure necessarily results in the ambiguity. In contrast, in LON-W2 directly links the verb and the instrument due to the larger window size, so that the verbs may select on the instrument that is associated to it in the input (*grow-fertilizer* and *preserve-vinegar* in Figure 6.2b). With the observation, we can predict that LON-W2 may successfully learn the two-way and three-way dependencies, while LON-W1 confuses between the two instruments associated with the patient, resulting in an around 0.5 performance in the tasks.

6.4.1 Model Training

Training in the LON models involves converting sentences into graphical form and joining the resulting sub-graphs at shared nodes (Figure 6.1c-d). Once the network is formed, the pair-wise lexical relatedness in LON is computed following Equation 5.3. Once the pair-wise lexical relatedness are computed, the VP-instrument relatedness is computed using Equation 5.5.

6.4.2 Results

As predicted, the LON-W2 performed perfectly in both the two-way and the three-way dependencies, while the LON-W1 confused between two competitors (two-way dependency: m=0.51, sd=0.09; three way dependency: m=0.49, sd=0.09). Follow-up analysis shows that the failure was due to the verb's confusion on the two instruments associated to the patient noun, e.g. *fertilizer* and *vinegar* for *cucumber*. LON-W1 was not able to tell which instrument of the two is closer to *grow* or to *preserve*, while LON-W2 had no confusion.

Table 6.3: Accuracy of LON models in inferring the structurally-licensed instrument in learning two-way (L2) and three-way (L3) dependencies. Accuracies are averages across 30 seeds.

	Learning	
	L2	L3
HAL-relation	0.86 (0.04)	1.00 (0.00)
LON-W1	0.51 (0.09)	0.49 (0.09)
LON-W2	1.00 (0.00)	1.00 (0.00)

6.4.3 Discussion

In this section, I tested two LON models on the learning tasks. With a larger window size, the LON-W2 model directly encodes the verb-instrument contingencies which is missed in the LON-W1 model. The result seems to support for larger window when it comes to encoding co-occurrence, as more useful information, e.g. the verb-instrument relation in the current case, can be encoded with a larger window. Nevertheless, larger windows may bring in more noises as well, resulting in more spurious relations encoded in the model, see (Bullinaria & Levy, 2012) for more discussions.

Moreover, the information brought in by the larger window has to be organized and accessed in a proper way. Recall that the HAL-relation model also has a larger window size, but it performed even worse than the LON-W1 network. In LON-W2, the co-occurrence between the verb and the instrument is encoded as a direct link in the network, so that the word node pair has large relatedness in the graph. In contrast, in HAL, the verb-instrument co-occurrences are placed in the word vector entries, which may not be directly accessed through the cosine similarity measure. To be more specific, the HAL model is better at capture the similarity between words with similar context, e.g. *cucumber* and *potato*, but not the relation between words that directly co-occur, e.g. *cucumber* and *vinegar*. It has been shown that the count-vector models in general have trouble in simultaneously capturing both the contextual similarity and the direct co-occurrence (Sahlgren, 2006). This is reflected in the Chapter 3, as well as the current study by the worse performance of the HAL-relation model compared to LON, its graphical correspondence.

Need to note that the non-adjacent verb-instrument dependency can be captured in other ways. For example, the CTN model does not directly link the verb and the instrument. In CTN, they are connected through the phrasal nodes (Figure 6.2c), so that the connection does not traverse the intermediate patient. As a result, the verb-instrument contingencies are naturally encoded, not interfered by the patient. In contrast, the LON is a degenerate version of the CTN in which word chains (words adjacent in the training data) instead of constituency-parse trees are joined. The absence of phrasal nodes in LON makes it difficult to represent the dependency between a VP and associated instruments (Figure 6.2a), and a larger co-occurrence window is needed to directly encode the relation (Figure 6.2b). While the LON-W2 model works for the current corpus, such that the lexical dependencies are uniformly encoded in the language, it probably needs to flexibly adjust the co-occurrence window size when processing on naturalistic data. The comparison between the CTN and LON models shows how syntactic pre-processing might benefit on semantic encoding quality.

The LON cannot form representation of complex expression, and must compose the word-word relation to obtain multi-way dependency. As a result, it relies on accurately capturing the word-word lexical dependencies. On the other hand, the prediction based Recurrent Neural Networks (RNN) form a holistic representation of multi-word sequence and easily build the relation between the sequence and the incoming word. In other words, it captures the multi-way dependency in the representation-composing way. I examine the RNN models in the next section and show that while the RNNs may solve the learning tasks easily, they struggle to generalize from the learned multi-way lexical dependencies to the novel lexical combinations.

6.5 RNN

Earlier on, I showed HAL like models struggled to form effective representation of complex expressions/sequence from the word representations. A way to overcome the limitation is to train models to *learn* to combine consecutively occurring lexical items without intervention from the researcher (Elman, 1990; Hochreiter & Schmidhuber, 1997; Vaswani et al., 2017; Radford et al., 2019). These models are usually neural network models trained to predict next-words given a sequence of words. By being trained to predict which words are likely to come next, such models learn implicit composition functions for how to combine words in their input. These functions are encoded in the network's weights, and are gradually tuned to the statistics of the training corpus. An advantage of these prediction based models over count-vector models is that they excel at learning which words tend to go together in a sequence. Similar to HAL like models, these prediction-based neural-net models have been extremely useful in accounting for a broad range of behavioral phenomena in the psycholinguistic literature.

To start with, I focus on the recurrent neural network (RNN), one of the most well known prediction based neural-net DSM. At its heart, the RNN uses a distributed pattern of activations at its"hidden layer" to represent information about a sequence of items it has been exposed to in the input. It is able to combine information across time steps via recurrent connections from the hidden state at the previous time step to the next. Usually, it is fed a sequences of words from contiguous text and is tasked to predict the probability distribution over next-words. Learning takes place by updating the networks connections (weights, parameters), so that prediction error on the training data is minimized. In this way, the multi-way dependencies in the corpus is gradually built-up. Having seen *John grow cucumber with fertilizer*, the connectionist network form a representation of the sequence *John grow cucumber with* that tends to select *fertilizer* over all other instruments as the next word in the sequence. As a result, identifying the observed VP-instrument dependency becomes a trivial task for the RNNs, it is what the model trained to do.

While success is expected for the learning tasks, it is less clear if the RNNs will generalize to sequence pairs not encountered during training. In the artificial corpus, VPs like *preserve pepper* has not been followed by any instrument in the training data. How can the RNNs infer on the plausible instrument for *preserve pepper*, based on the learned VP-instrument relations? As mentioned earlier, one option is to generalize from the similar VP, e.g. *preserve cucumber*. If the model form similar representations for *preserve cucumber* and *preserve pepper*, namely, similar representations at the hidden layer, it may successfully predict *vinegar* for *preserve pepper*. However, in RNN, the similarity between phrases (a chunk of multiple words) usually turns out not combinatorial. Unable to build up the similarity combinatorially remains a weakness of the RNN models, and I illustrate the issue in detail.

First, the RNNs learn 'holistic' representations of complex expressions at their hidden layer. For example, if the two-word sequences *preserve pepper* and *preserve cucumber* share a lot of contexts in common, e.g., they are always followed by a same set of words, then the RNNs can easily form similar 'holistic' representation in the hidden states for the two phrases. Nevertheless, this is not the case in the artificial corpus here (nor the natural language in general). In the corpus, *preserve pepper* is always followed by the end-of-sequence mark, or a period, while *preserve cucumber* is always followed by the instrument *vinegar*. Therefore, the two phrases as a whole are dramatically dissimilar in terms of their context. Instead, the two phrases are very similar in a combinatorial manner: The two VPs have the identical verb, and their patients are from the same category, thus always follow the same set of verbs. The corpus design mandates the RNN to build up the distributional similarity by combining the pair-wise lexical similarities (*preserve-preserve* and *cucumber-pepper*), i.e. in the dependency-composing way.

However, RNNs were not developed to explicitly keep track of the identity of the individual lexical items that have occurred in the input over the course of processing. Instead, its holistic representation integrates lexical semantic information without explicitly enforcing the preservation of information about which word in the input is responsible for what portion of the activation pattern at the hidden layer. At each time step, rather than updating a portion of the hidden layer state and keeping track of which item is responsible for each portion of the state, the entire hidden layer state is updated all at once, or as a whole. As a result it would be difficult for RNNs to capture the similarity between *cucumber* and *pepper*, and use the lexical similarity to build up the phrasal similarity. I predict that while RNNs can perfectly learn the two-way (*grow cucumber - fertilizer*) and three-way dependencies (*preserve cucumber - vinegar*) from the corpus, they will fail to generalize to the instrument-vacant VPs (*grow/preserve pepper*).

6.5.1 Model Training

I examined two RNN models: The SRN (Elman, 1991) and LSTM (Hochreiter & Schmidhuber, 1997). I identified one highest-performing hyper-parameter configuration for each model after extensive tuning on the 3-way generalization task. Hyper-parameter search was restricted to 1-layer and 2-layer architectures. For 1-layer architecture, 64 hidden units and 32 hidden units performed best for the SRN and LSTM respectively, and 64 hidden units worked best for 2-layer architectures. I examined models on both 1-layer and 2-layer architectures, but will only focus on 2-layer architectures in the result sections. I will come back to the 1-layer architect in discussion. In keeping with the format of the training task, I operationalized VP-Instrument semantic relatedness in terms of prediction error: Given as input *'John preserve pepper with _'*, I computed the prediction error at the last time step, substituting *'_'* with an instrument. The instrument with the least error is selected to be the model prediction. This is in accordance with previous proposals where constraints on predictive processing are considered to reflect knowledge of typical events (McRae et al., 2005).

6.5.2 Results

I examined the SRN and LSTM models on both learning and generalization tasks. As predicted, they both performed well in the learning tasks, in contrast to most HAL models and LON-W1 model. However, in generalization tasks, the two RNNs performed no better than the LON-W1 model, with LSTM's performance better than the SRN (Table 6.4). Especially, generalization on the three-way dependencies ended up as a challenge, (SRN:m=0.28, sd = 0.15; LSTM: m=0.26, sd=0.15). Following-up analysis shows that a great portion of the errors (in average, 30 percent) were from unrelated instrument, i.e. instruments that were assigned neither to the targeting category nor to the sibling category. For now, I do not have a clear interpretation for this specific error in the RNNs. Further diagnoses are needed to better explain the model behaviors.

Table 6.4: Accuracy of RNN models inferring the structurally-licensed instrument in learning two-way (L2) and three-way (L3) dependencies, and generalization on two-way (G2) and three-way (G3) dependencies. Accuracies are averages across 30 seeds.

	Learning		Generalization	
	L2	L3	G2	G3
HAL-relation	0.86 (0.04)	1.00 (0.00)	0.47 (0.05)	1.00 (0.00)
LON-W1	0.51 (0.09)	0.49 (0.09)	0.51 (0.10)	0.47 (0.17)
SRN	0.93 (0.16)	0.91 (0.16)	0.42 (0.19)	0.28 (0.15)
LSTM	0.83 (0.29)	0.84 (0.28)	0.46 (0.22)	0.26 (0.15)
Random	0.03	0.03	0.03	0.03

6.5.3 Reverse Sequence

Since the RNNs perfectly captured the multi-way dependencies in the input, the lower performance on the generalization task of the RNN models could not be attributed to the failure in learning the observed VP-Instrument relation. Therefore, it should concern the semantic similarity between the VPs (*preserve pepper* - *preserve cucumber*). One critical issue could be that the semantic information provided in the corpus was not sufficient for the RNN models. The RNN models investigated in this study were trained to predict the next word in a sequence. These models tend to form semantic representation of words based the right context of the words. The instrument-vacant patients were always followed by the period mark on the right, so that the RNNs were less possible to form the similarity hierarchy for the instrument-vacant patients based on the right context. On the other hand, the critical information for forming the patient category were provided in the verbs, which were the left context of the patients. It could be more difficult for the RNNs in these experiments to learn the semantic structure from the left context, that had led to the lower performance in the generalization task.

To test if the less ideal performance of the RNNs in the generalization tasks was due to the directionality of semantic information encoding, I adjusted the artificial corpus so that the critical categorizing information, i.e. the verbs, appeared on both sides of the patients. I updated the corpus by adding in the reversed copy of the original corpus. To be more precise, for each generated sentence, I added a copy of the sentence but in the reverse word order. For example, for *John preserve pepper.*, I added its reversed version: *. pepper preserve John.* In this way, the critical semantic information, i.e the verbs also appeared on the right side of the patient noun, so that the RNNs may learn the similarity structures through predicting the verbs. I replicated the generalization experiments on the RNNs with the enlarged corpus.

Table 6.5: Accuracy of RNN models on the generalization tasks, trained on the original corpus and the corpus with reversed sentences added.

	Generalization	
	G2	G3
SRN	0.42 (0.19)	0.28 (0.15)
SRN, add_reversed	0.55 (0.31)	0.47 (0.27)
LSTM	0.46 (0.21)	0.26 (0.15)
LSTM, add_reversed	0.67 (0.31)	0.57 (0.29)

If the lower performance in the generalization was indeed due to the lack of left context for deriving the similarity structure of the patients, then including the reversed sentence should eliminate the effect from encoding directionality, and the performance of the RNNs in the G3 condition should approach ceiling. On the other hand, in the G2 condition, the generalization solely depends on the verb (e.g.*grow-fertilizer*). Therefore, adding the reversed sequences should not improve the performances in G2. The results are reported in Table 6.5. The RNNs improved in both G2 (SRN: from m=0.42 to m=0.55; LSTM: from m=0.46 to m=0.67) and G3 (SRN: from m=0.28 to m=0.47; LSTM: from m=0.26 to m=0.57) conditions after the reversed sentences were added. Need to note that while the SRN and LSTM improved, there were still enormous gaps between their performances and the ceiling (CTN). This suggests that while the lower performance on G3 in RNNs was partially attributed to the encoding directionality and the lack of critical semantic information on the appropriate position, other mechanistic problem in RNNs led to their failure in generalization.

6.5.4 Discussion

In this section, I examined one type of neural network models: RNN, that forms a holistic representation of the inputted sequence in the hidden layer, and uses it to predict the incoming words. I showed that while the models easily learned the multi-way lexical dependencies in the training input, they struggled in generalizing from what they learned to novel phrases. This indicates that the failure was due to the difficulty in capturing the similarity between the instrument-associated and instrument-vacant VPs. By adding reversed-order sequences, I confirmed that the less ideal performance was at least partially due to the intrinsic mechanism of the neural model – the inability to build up the VP similarity from the word-word similarity. Such weakness limited the RNNs in making reliable predictions on novel combination of familiar words.

In the next section, I examine a more recent type of prediction-based neural-net models: the Transformer (Vaswani et al., 2017). These models can effectively learn the multi-way lexical dependencies in the input, with a mechanism largely different from that in the RNNs. I show that with the new mechanism, the Transformer type of models successfully generalize from the learned dependencies to novel word combinations. By comparing between the Transformer based models with RNN, and also the CTN, I explore the essential computational mechanism towards compositional generalization.

6.6 GPT

In this section, I investigate the Transformer model (Vaswani et al., 2017). To be more specific, I adopted a miniature version of the GPT-2 Transformer architecture (Radford et al., 2019). While trained using the same next-word prediction task, unlike RNNs, mini GPT-2 does not forcibly compress lexical representations into one holistic representation of across all time steps. Instead, it integrates words with its parallel architecture, using the self-attention mechanism (Vaswani et al., 2017), which is itself governed by weights learned over the course of training. One major advantage of self-attention, and in particular, multi-head self-attention (Vaswani et al., 2017), is that integration of information is not restricted to a single function, as is the case in the RNN (which has only a single recurrent weight matrix), but is unique to a given pair of words and is sensitive to the context in which those words have occurred. In this way, GPT-2 can keep track of the individual identities of all the words in the model's input, and allows for more flexibility concerning how and when lexical representations are combined. To better illustrate the potential advantages of GPT-2, I walk through the mechanism of the Transformer with more details, in the scenario of our generalization tasks. I explain from the modeling theoretic perspective, that how the GPT-2 model may overcome the limitations of the RNNs and has a greater chance at success in the generalization tasks.

6.6.1 Transformer Architecture and Mechanism

The Transformer architecture consists of stacked Transformer blocks, in each of which an attention layer is located. The input sequence is propagated up through the blocks before the final output layer vocabulary selection. To start with, each token in the input sequence is the vector embedding of its type and all tokens in the sequence are simultaneously propagated into the Transformer blocks (Figure 6.3a). Within the i'th block, three matrices, i.e. the query matrix $\mathbf{W_Q^{(i)}}$, the key matrix $\mathbf{W_K^{(i)}}$ and the value matrix $\mathbf{W_V^{(i)}}$ respectively operate on all token vectors simultaneously, ending up with a query vector, a key vector and a value vector for each of the input token. Namely, for the j'th token in the input, we have

$$\mathbf{q_j^{(i)}} = \mathbf{W_Q^{(i)}} \mathbf{x_j^{(i)}}, \qquad \mathbf{k_j^{(i)}} = \mathbf{W_K^{(i)}} \mathbf{x_j^{(i)}}, \qquad \mathbf{v_j^{(i)}} = \mathbf{W_V^{(i)}} \mathbf{x_{(j)}^{(i)}} \tag{6.2}$$

in which $\mathbf{q_j^{(i)}}, \mathbf{k_j^{(i)}}, \mathbf{v_j^{(i)}}$ are the query, key and value vector for the j'th token in the i'th Tansformer block. The value vector can be considered as a 'lexical' representation of the token within the attention layer, the query vector of a token represents 'how' the token is going to 'attend' on other tokens, and the key vector represents 'how' the token may be attended by other tokens within the layer. To put it formally, for the j'th token in the input, the model computes its attention score to the l'th token, denoted as a_{jl}:

$$a_{jl} = \frac{\mathbf{q_j} \cdot \mathbf{k_l}}{\sqrt{d_k}} \tag{6.3}$$

in which d_k is the dimensionality of the key vectors. Once the attention scores are computed, the value vector of token j is updated by its 'self-attention' $\mathbf{u_j^{(i)}}$:

$$\mathbf{u_j^{(i)}} = \sum_l a_{jl} \mathbf{v_l^{(i)}} \tag{6.4}$$

which is a recombination of all token vector inputs, weighted by its attention (a_{jl}) to these tokens. The updated values $\mathbf{u_j^{(i)}}$s go through a fully-connected feedforward layer with residuals and normalization, before they are propagated, as the inputs, into the next Transformer block. Once the input goes through all the blocks, each word token predicts its next word (Figure 6.3a) simultaneously. The feedback is back-propagated, and learning happens on all connections, including the query, key and value matrices of the attention layers in each Transformer blocks. Gradually, these attention layers learn how to attend to the tokens in the block through its value matrix, and how it should attend to, and obtain attention from other tokens through its query and key matrices.

Several important aspects concerning the mechanism of Transformer are vital to the success of the generalization task. First of all, it keeps a separate representation (the value vector) of each positioned token in the sequence throughout the forward propagation, and the tokens are processed simultaneously in parallel. In addition to the separate representations, the attention mechanism makes it possible to flexibly combine (compose) the token representations. Figure 6.3 illustrates the difference between the Transformer model and the RNN models when processing a sequence. In an RNN model, the input tokens in *Mary preserve pepper with* are entered one by one. When the four words have been seen by the model, a holistic representation of the whole sequence is formed in the network, and it is difficult to tell how each word token have contributed to this holistic representation. However, in the Transformer architecture, in each Transformer block, the

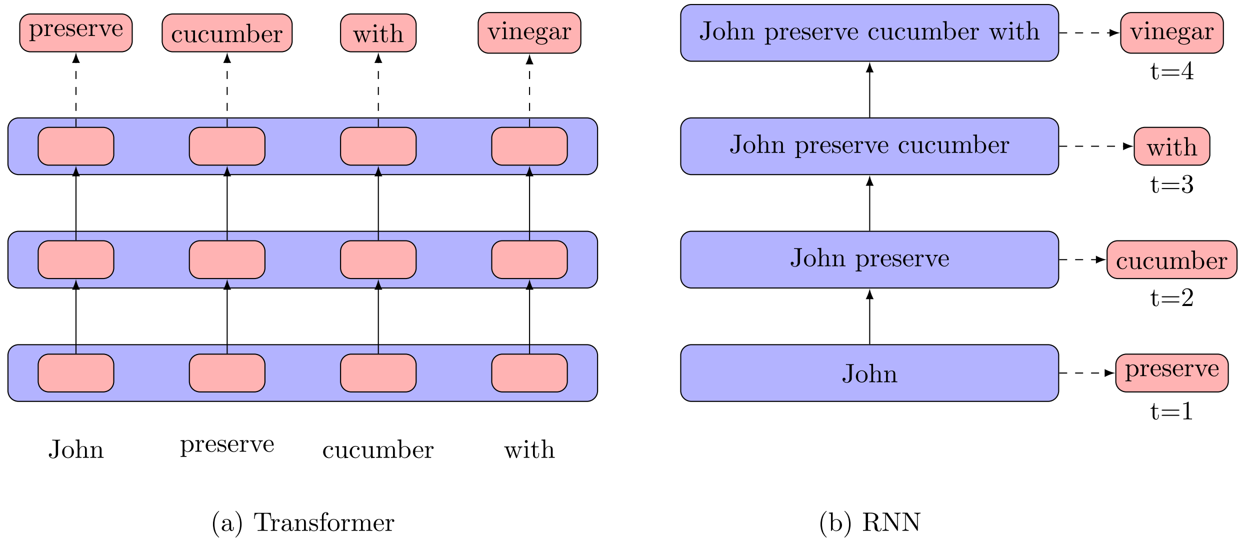


Figure 6.3: Architecture and processing mechanism in Transformer. (a) Transformer process and predict the next word for each word in the sequence in parallel. (b) RNN forms holistic representation at each moment for predicting the next word.

representations of the positioned tokens are processed in parallel. In other words, they do not **necessarily** mingle. These representations can be context-free, if the token has never paid any attention to other tokens in the previous blocks. Alternatively, they can become contextualized, if the prediction task requires it to pay attention to other tokens. Recall that 'paying attention' to other tokens means updating the token representation by recombining with representations of other tokens (see equation 6.4). How each token in each block attends to other tokens, i.e. the actual attention attribution, is decided by the query and key matrices, which are the results of learning. Taking these together, what the model ends-up learning, is a flexible combination (composition) of the token representations towards successful prediction, which may potentially capture lexical dependencies in the input in an effective way.

I walk through different approaches that the Transformer model may adopt for the generalization tasks. The critical task is to predict the upcoming instrument (word) given the incomplete sequence such as *Mary preserve pepper with*. In the Transformer model, it is the token *with* (the value vector of *with*) that leads to the prediction at the final output layer (Figure 6.3). However, the representation of *with* in the last Transformer block is very likely to be a combination of *with* and other tokens in the input. There are multiple ways that such combination can be made. The first option is that *with* does not attend to any other words in the first several blocks, while *pepper* attends to *preserve* and a representation of *preserve pepper* is formed (placed in the position of *pepper*). Then *with* can attend to *pepper*, to be more specific, a combined representation of *preserve pepper*. In this way, *with* ends up forming a combined representation in the last few blocks which leads to the prediction. This approach resembles the process in the RNN models, in which the holistic representation of VP is formed as each token is inputted sequentially, with the goal to predict the instrument. As discussed before, the approach leads to effective learning of the VP-instrument dependency observed in the input, but it may fail in the generalization case: *preserve pepper with*. Since *preserve pepper* is always followed by the end-of-sentence mark in the input, its representation in the Transformer might be different from the representation of *preserve cucumber*. Therefore, while *with* attending to *preserve cucumber* may lead to the prediction of *vinegar*, attending to *preserve pepper* may totally distract the model onto another direction. If the mini GPT-2 succeeds in the task, it will be unlikely that the model adopts such an approach.

Alternatively, *with* can attend to *pepper* and *preserve* individually. Ideally, it can first attend to *pepper*, to form a representation that narrows down the scope of the instruments to *vinegar* and *fertilizer*, i.e. the two instruments that other vegetables like *potato* and *cucumber* co-occur with. Need to note, the model can recognize *pepper* as *cucumber* since the words occur in very similar contexts, so that their value vectors and the effects of these vectors on *with* are likely to be similar. This first selection can be done during the first several Transformer blocks. In the later blocks, *with* may attend to the 'clean' representation of *preserve* (the token has not attended to and combined with other tokens before). In this way, it may further exclude *fertilizer* from the plausible instrument list, as neither *preserve*, nor the current representation of *with* attend to *fertilizer*. If the model succeeds in the task, then it is likely that the model adopts this second approach. Need to emphasize, this second approach is only possible with the parallel architecture and the attention mechanism in Transformer, and it is not available for the vanilla RNNs. If the results show the success of the Transformer, it will indicate that it is the flexible representation combination founded by the Transformer architecture and attention mechanism that leads to the success in generalization.

6.6.2 Model Training

The training of mini GPT-2 was similar to that of the RNNs. I tuned both 1-layer and 2-layer Transformer architectures, and only report the results of 2-layer models, leaving the 1-layer models for discussion. I identified one highest-performing hyper-parameter configuration for each model after extensive tuning on the three-way generalization task. I found that 32 hidden units performed best for the Transformer. In particular, the Transformer required a significant amount of hyper-parameter tuning to reduce variance over different seeds. While I observed strong performance of the Transformer during tuning (near 100% accuracy in all conditions), there was a noticeable drop in performance in the three-way generalization condition (see Results) after re-training on 30 novel random seeds. The VP-Instrument semantic relatedness was operationalized in the same way as the RNN training. Since the mini GPT-2's training also followed the predicting-the-next-word regime, it might as well suffer from the directionality issue. Therefore, I trained the model on both the original corpus and the corpus with reversed sentences added. The model should perform better when the corpus is bi-directional.

6.6.3 Results

The mini GPT-2 model, whether trained on the original corpus or the augmented corpus performed well in all tasks. First, similar to the RNNs, the model trained on both corpora had ceiling performance on the learning tasks. In addition, mini GPT-2 almost perfectly generalized in the G2 condition (*grow pepper* - *fertilizer*), which the LSTM had failed. Most importantly, the mini GPT-2 model had a relatively high performance in the three-way generalization task (m=0.87, sd=0.15), even without adding the reversed sentences. The results suggest that mini GPT-2 had not only learned the multi-way dependencies provided in the input, but also successfully generalized to novel VPs that had not been associated with any instruments in the corpus. In addition, the result suggests that in mini GPT-2, the representation of *with* probably combined with the representation of *pepper* and *preserve* in a step-wise way. When it attended to *pepper*, it narrowed down the selection on *vinegar* and *fertilizer*, and it pinpointed on *vinegar* after it had attended and combined its representation with that of *preserve*. Need to note that these detailed mechanisms are still speculative, and systematic investigation on the resultant network activation is needed to test the hypothesis in future studies.

Table 6.6: Accuracy of mini GPT-2 inferring the structurally-licensed instrument in learning two-way (L2) and three-way (L3) dependencies, and generalization on two-way (G2) and three-way (G3) dependencies. Accuracies are averages across 30 seeds.

	Learning		Generalization	
	L2	L3	G2	G3
LSTM	0.83 (0.29)	1.00 (0.00)	0.46 (0.22)	0.26 (0.15)
mini GPT-2	1.00 (0.00)	1.00 (0.00)	0.98 (0.04)	0.87 (0.15)
mini GPT-2, add_reversed	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.91 (0.20)

6.6.4 Complete Novel Combination

In all generalization tasks above, the instrument-vacant VPs were ‘novel’ not in the sense that they were not observed by the model, but in the sense that they did not collocate with any instrument in the training corpus, so that there was minimal distributional information for relating the VPs to other VPs a whole. Nevertheless, the occurrence of the phrase may still be informative to some of the models, which had made the current task different from other compositional generalization tasks, such as the SCAN tasks (B. Lake & Baroni, 2018) in which the tested stimuli never occur in the input. I am curious about how the model would react to a totally novel verb-patient combination. How would the models infer on the plausible instrument for *preserve pepper*, if this phrase has never been given to the model? To make the instrument-vacant VPs totally novel to the model in the test, I eliminated the sentences with those VPs in the corpus. For example, *John preserve pepper.* was excluded, so was the corresponding reversed sentence. If a model relies on the occurrence of the phrase, excluding them might result in lower performance in the three-way generalization task. Notice that the update was based on the corpus with reversed sentences.

Table 6.7: Accuracy of mini GPT-2 and CTN on the generalization tasks, trained on (1) the corpus with reversed sentences added, or trained on the corpus with reversed sentences but without either (2) the three-way instrument-vacant VP (e.g. *preserve pepper*), or (3) the two-way instrument-vacant VP (e.g. *grow pepper*) or (4) both two-way and three-way instrument-vacant VPs.

	Generalization	
	G2	G3
GPT-2, add_reversed	1.00 (0.00)	0.91 (0.20)
GPT-2, add_reversed, *preserve pepper* removed	1.00 (0.00)	0.70 (0.40)
GPT-2, add_reversed, *grow pepper* removed	1.00 (0.00)	0.97 (0.07)
GPT-2, add_reversed, both phrases removed	1.00 (0.00)	1.00 (0.00)
CTN, all conditions	1.00 (0.00)	1.00 (0.00)

I trained all models on the upated corpus and retested the models on the generalization tasks. After excluding the instrument-vacant VPs, the GPT-2 model’s performances in the three-way generalization dropped considerably (G3: m=0.70, sd=0.40, see row 1 and 2 in Table 6.7), This indicates that the model mistakenly confused some instrument with the top candidate. Interestingly, the performance on the two-way generalization remained at the ceiling for mini GPT-2. In comparison, the CTN stayed at the ceiling even with the phrases removed from the training corpus.

The decreased performance of Transformer in the generalization task was intriguing, as it might provide more insight about how the model had achieved. With further inspections, I found that the model selected *fertilizer* instead of *vinegar* for the most of the times. I speculate that this was due to the absence of *preserve pepper* in the training set. As a result, the model took *preserve pepper* as *grow pepper*, and ended with representation resembling those for *grow pepper* or *grow cucumber*. Consequently, it predicted *fertilizer*, the designated instrument for *growing* things. To test this speculation, I further manipulated the corpus. I deleted the 'interfering' two-way instrument-vacant VPs, e.g. *grow pepper*, while kept or deleted the three-way instrument-vacant VP, e.g. *preserve pepper*. I trained the mini GPT-2 and CTN on these two new corpora and tested them on G2 and G3 tasks. If the mini GPT-2 was indeed distracted by the two-way competitor, then deleting them would eliminate the effect, leading to better performances. This turned out to be the case. The Transformer model performed at ceiling when both phrases were removed, and performed almost at ceiling when only the competitor (*grow pepper*) was removed (see row 3 and 4 in Table 6.7). Interestingly, when the competitor was removed, the model performed significantly higher when the competitor stayed. This suggests that the mini GPT-2's three-way generalization was somehow sensitive to the input. Need to note, the CTN model had perfect performances across these different corpora.

Here I provide a speculative interpretation of Transformer's behavior, which might further hints on how the model function in the language task. I hypothesize that mini GPT-2 succeeded by attending to different tokens in the VP separately alongside the propagation upwards the Transformer blocks. However, while the embedding fed to the first block were 'clean' (in the sense that they had not been combined with the representation of other tokens), the value vectors in the upper blocks were very likely to have been contextualized. That is, when *with* attended to *pepper* on the second block, it was possible that the value vector for *pepper* on the layers had been a combination of *pepper* and *preserve*. In this case, the model might not pay attention to the representation of the tokens separately, and it might be confused on *preserve pepper* in the test phase, if the phrase had not been given in the training set. When only *preserve pepper* was removed, in the test phase, GPT-2 may form a representation of *preserve pepper* which is similar to that of *grow pepper*, on the position of *pepper* in the upper attention blocks. When the model processes the novel phrase *preserve pepper with*, *with* could be confused by the competing clues from the value vector of *grow pepper* on the *pepper* position and the value vector for *preserve* on the *preserve* position. While the first representation would direct *with* towards predicting *fertilizer*, the second would bias it towards *vinegar*. In this way, it may lead to indecisiveness between the two instruments, ending up with the presented result.

However, when *preserve pepper* was included in the training corpus, it would prevent the model from recognizing *preserve pepper* as *grow pepper*. Since the model had seen the phrase, it should be able to form a distinctive representation of *preserve pepper* (as to that of *grow pepper*), and the representation of *pepper* should not be confused with that of *grow pepper*. As a result, in the test phase, *with* would only attend to less confusing representation of *preserve* and *pepper*, and narrow down the selection to *vinegar*. Lastly, when the interfering *grow pepper* was removed, the model would not conflate *pepper* with *grow pepper* from the beginning, so that it should be able to make the right prediction.

6.6.5 Discussion

In this section, I investigated the Transformer based mini GPT-2 model. The model succeeded in all tasks, especially the challenging three-way generalization tasks, even in the corpus without reversed sentences. Basing on the result, I speculate that the model's achievement was due to its flexible combination of the instrument (predicted by the word *with*) with the patient and the verb in a step-wise way. Such a behavior is allowed by the parallel network architecture and the attention mechanism in Transformer. However, with following-up tests, I showed that the success of mini GPT-2 model was sensitive to the input: It did not ace the three-way generalization task when the critical phrase (*preserve pepper*) was absent and the competitor (*grow pepper*) was present. While this might be an issue with the specific corpus here, it remains an open question how the contextualized representation would affect the model when it is trained on the more complex naturalistic language input. I get back to this topic in the general discussion.

6.7 Discussion on Multi-way Lexical Dependency

In this chapter, I examined a series of distributional semantic models on their capability of learning and generalizing multi-way lexical dependencies. I found that the HAL model failed in the learning tasks due to its less effective lexical representations, and this was remedied by the LON model with appropriate window size. The connectionist RNN models addressed the learning problem in another way: It directly formed a holistic representation of the VP (sequence) and captured the relation between the chunk and the instrument through the prediction-based training. However, the holistic representation made it difficult for the RNNs to generalize to novel VPs. Finally, I showed that the Transformer-based mini GPT-2 model tackled the compositional generalization problem with the attention mechanism operating on a parallel processing architecture. These computational characteristics functionally approximate the CTN model, which performed perfectly in the learning and generalization tasks.

In this section, I elaborate on the models' success and failures in order, and propose two computational capabilities towards success in learning and generalizing on multi-way lexical relations. I argue that these learning and generalization tasks are essential components of semantic cognition, and the capabilities on the task reflects a system's competence in a critical aspect of semantic processing. Then, by aligning the gold-standard CTN model and the Transformer model, I propose that some form of 'syntactic' processing might be essential in the process of 'becoming semantic', and the syntax-semantics interaction may play an important role in language development. I talk about two lines of work inspired of the current work: (1) modeling endeavor concerning semantic representation in Transformer-based language models; and (2) behavioral experiments on the development of complex semantic dependencies in human.

6.7.1 The Hard Semantic Problem

In Chapter 5 and 6, I stress and focus on the hard semantic problem: representing and processing the multi-way lexical dependency in a generative language. The starting point is to effectively represent two-way or word-word lexical relations, which have been studied for more than half a century (Collins & Quillian, 1969; Collins & Loftus, 1975; Deese, 1962; Smith et al., 1974; Resnik, 1996). The more recent distributional models (Jones & Mewhort, 2007; Landauer & Dumais, 1997; Lund & Burgess, 1996; Mikolov et al., 2013) seem to have found effective ways to represent the two-way lexical relations, e.g. the LSA model reached a decent score in the TOFEL test by merely calculating the word-vector similarity. Nevertheless, there are evidences showing that the lexical representations in these models are not effective enough to address harder semantic problems, namely, scaling up to the dependency between three or more lexical items (Chen, Peterson, & Griffiths, 2017; Mitchell & Lapata, 2010). I followed on this line of works and designed the VP-instrument (multi-way lexical) dependency learning task. If the lexical representations in the HAL type model are effective enough, they should to some extent learn the multi-way lexical dependency. The model failed. Neither the pair-wise relations formed from the lexical representation could be composed into effective VP-instrument relations, nor the VP vector representation composed from the word representations could select the right instrument with the canonical vector space measure. The failure was not specific to the HAL model. I examined more recent models like Word2Vec (Mikolov et al., 2013) which also form static lexical representations, and the performance was similar.

Multiple factors potentially attributes to the failure. First, it has long been argued that, due to an unavoidable encoding deficit, these models struggle to entertain different types of semantic relations simultaneously. They either well capture co-occurrence relations such as *bread-knife*, or similarities such as *bread-toast*, but never both of them (Sahlgren, 2006; Mao & Willits, 2020). Here, the HAL-relation model failed to capture the verb-instrument co-occurrence. Another problem is that it is difficult to find an explicit analytical function to compose the lexical vectors to form effective phrasal representations (Mitchell & Lapata, 2010), and this was also reflected in the two HAL-composition models (see Table 6.2). Based on these evidences, I argue that the static lexical representations formed in these 'word-encoding' models are not sufficiently effective for more involved semantic tasks.

I presented two alternatives to address these issues of HAL models in the learning tasks. One of them is the RNN, which forms a holistic representation of the VP (the sequence) to predict the instrument. While the RNNs have no trouble in capturing the observed VP-instrument dependencies, they failed to generalize on novel verb-patient pairs. The critical deficit of the recurrent model is that, they aggregate the parts for the representation of the whole, at the cost of the identity of the individual parts. As a result, they would take *preserve pepper* to be more similar to *spray berry* in contrast to *preserve cucumber* as the first two VPs as a whole have the identical distribution (followed by the period), which is different from that of the third (followed by *with* and an instrument). In other words, the model may not represent the VPs in a combinatorial way. As a result, it is a challenge for the RNNs to generalize in a combinatorial pattern. It would generalize to *store pepper* and infer the plausible instrument *vinegar* only if when the phrase as a whole had similar distribution to *preserve cucumber*, but not in the case that *store* and *pepper* are similar to *preserve* and *cucumber* respectively, while the phrases as a whole does not share any similarity.

What can be learned from the failure of the RNN models is: It is useful to keep the stand-alone representations of the lexical building blocks. Moreover, there has to be a way to flexibly and effectively combine these building blocks to form multi-way lexical dependencies. This gives rise to the graphical models like LON and CTN, and also the Transformer-based neural network models. Although differing in many aspects, these models share the two above-mentioned computational capabilities. The LON model can be considered as a graphical equivalent of HAL models. In contrast to HAL models which represent words as vectors in a space, these graphical models directly encode the lexical relations through the graphical structure, and they can assess different types of lexical relations (a fatal challenge to the HAL models) through the spreading activation measure. However, I have shown that the LON needs the proper window size to capture the necessary lexical relations for the first place (see Figure 6.2). While the LON with window size 2 can capture the lexical dependencies in the current artificial corpus, it requires a window with more flexible sizes when operating on naturalistic input (in which the dependencies may vary by distances). This is a non-trivial computational capability on itself. In contrast, the CTN model capture the lexical relations in a more effective way. With the constituent parsing, it directly transforms the syntactic structure into lexical dependency reflected in the conceptual structure of the sentence. Once the word-word lexical relations are effectively represented in the graphs, the multi-way dependencies can be computed from the word-word relations from a simple composition function (see Equation 5.5). On the other hand, the Transformer-based mini GPT-2 model tackles the task in a totally different way. It sticks to the vector lexical representations just like RNNs and HAL. However, unlike the recurrent nets, it represents lexical tokens separately and process them in a parallel way. Meanwhile, unlike the HAL type model which stores all lexical information in a static vector, the Transformer encode the lexical dependencies through the attention mechanism, so that multi-way dependencies can be formed by propagating through the attention layers.

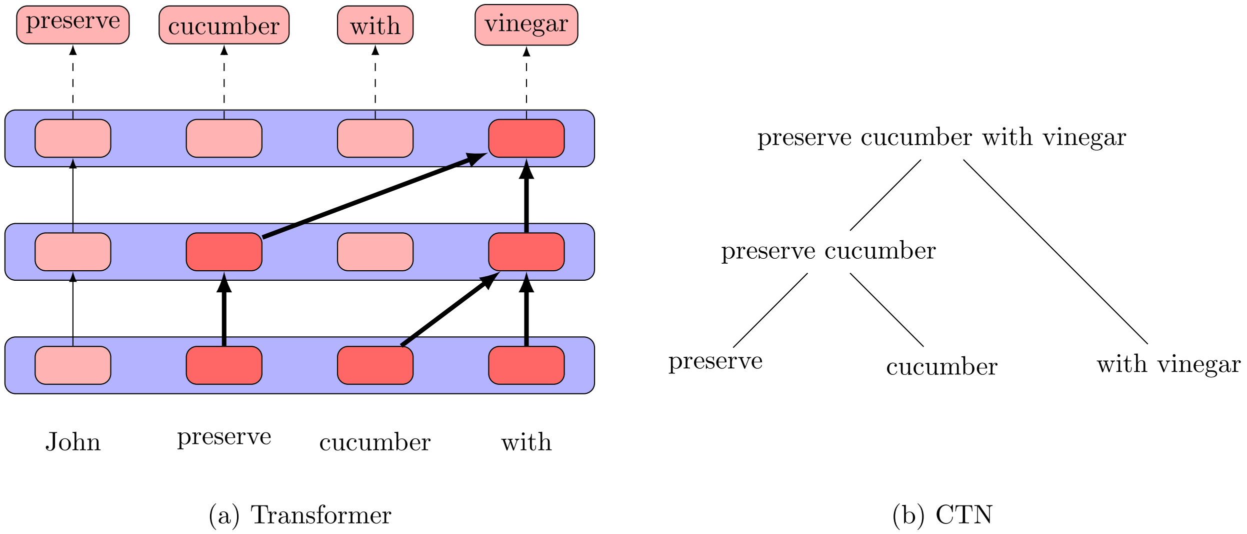
To summarize, while realized differently, the Transformer-based model and the CTN model can (1) effectively form lexical representations and (2) combine them flexibly. With these two computational capabilities, the knowledge of the models are not restricted to the particular lexical combinations that are given in the language input: they can effectively form representations with novel lexical combination, basing on the combinatorial patterns they have learned. This leads to the success in the challenging generalization tasks involving multi-way lexical dependencies.

6.7.2 Becoming Semantic

Lexical semantics and sentence meaning are the two cornerstones of natural language semantics. My concentration on multi-way lexical dependencies connects the two facets of meaning: I investigated the lexical relationships in a sentential frame. To be more specific, while the word-word lexical relations such as *preserve-vinegar* and *cucumber-pepper* consists a great portion of semantic knowledge, the scaled-up multi-way lexical dependencies matter to the grammaticality and meaningfulness of sentences, (e.g. Is *Mary preserve cucumber with pepper.* an anomalous sentence?). Regarding this, the multi-way dependency task is qualitatively different from the two-way dependency tasks. In order to learn and generalize on the multi-way dependencies, it requires the models to capture some structure of the linguistic expressions, i.e. chunking the words or roughly parsing the sentence. This is reflected in the difference between the behaviors in HAL type model (that specifically focus on word representation), and the more capable Transformer and CTN models.

With a closer look at the CTN and Transformer, it is observed that some structural representation might be formed on their way towards the more full-fledged semantic capabilities. In CTN, the syntactic parse gives rise to the constituent structure, which further dictates the lexical dependencies. In this way, the lexical semantic representation benefits from the top-down 'guidance' from an established syntax module. In contrast, the Transformer model works in a bottom-up manner. The initial drive is to predict the next word. When there is complex lexical dependency in the input, the model has to capture such dependency to achieve in the prediction task. On the way, lexical representations are combined and some constituent-like structures might emerge (Figure 6.4). Regardless of the process being top-down (CTN) or bottom-up (Transformer), the formation of structural representation potentially separates mini GPT-2 and CTN from HAL-like static models and vanilla RNNs. These structural representations not only facilitate the models to capture the observed multi-way relations, but also help generalization in a regular way (*preserve pepper-vinegar* suggested by *preserve cucumber-vinegar*). As a result, these capabilities contribute to processing sentence meaning.

Unlike the explicit encoding in CTN, the emerging structure in mini GPT-2 was speculated from the result. However, there are further evidences suggesting the emergence in mini GPT-2. As mentioned earlier, I tuned RNNs and mini GPT-2 with both 1-layer and 2-layer architectures. When comparing 1-layer RNNs and mini GPT-2, I did not find a significant difference between their performances on the critical three-way generalization task. More importantly, 1-layer mini GPT-2's performance was around .75, which was a bit far from the ceiling. However, when the models were trained on 2-layer architectures, there was a significant enhancement in mini GPT-2's performance, while the RNNs performed even worse. Note that while all mini GPT-2 models had attention, no hierarchical representation could have been formed with the 1-layer architecture. This suggests that the second layer augmented the representational capability of mini GPT-2, with the speculative attention hierarchical structure formed in the 2-layer architecture.



(a) Transformer

(b) CTN

Figure 6.4: Structural representations in Transformer and CTN. **(a)** Speculative learning happened in the Transformer architecture: the word token *with* gradually learned to attend tokens across layers which end up forming a hierarchical structure through forward propagation. **(b)** The corresponding constituent tree used to build CTN.

Now I come to a temporary conclusion mark on what makes a full-fledged semantic system: The system should be able to learn idiosyncratic multi-way lexical dependencies and generalize the relations sensibly to novel lexical combinations. Such an ability may help in (1) establishing a full-fledged lexical representation of individual word meaning, (2) evaluating sentences with lexical combinations that have never been met. To make that happen, a learner or a model is likely to need separate representations of the lexical items, and flexibly combine the lexical representations and relations to handle more complex dependencies.

Meanwhile, some structural representations is likely to be formed alongside the system becoming semantically competent. While such a process might be lexically driven, the structural representations emerged through the process might give rise to some primitive syntactic structures. Taking the CTN and Transformer models together, I hypothesize an intriguing interaction between syntax and semantics: (a part of) syntactic capabilities might be driven by capturing the complex lexical dependencies in the language input (Transformer), while an established syntactic system may in turn have great impacts on forming sophisticated semantic representation (CTN). The interaction might play a critical role in language development in general, and I propose two lines of future endeavors relating to it.

6.7.3 Future Directions

First, the current study motivates diagnostic research on Transformer-based models, to investigate what representation is formed for the model to learn the multi-way lexical dependencies and generalize combinatorially. Researchers may train the Transformer models on a similar corpus that encodes multi-way lexical dependencies and extract the activation states in the attention layer during and after the training. To be more specific, these diagnoses may respond to the speculation on how the Transformer based model capture the lexical dependencies. Does the model really pay attention to word tokens in a 'hierarchical manner' while propagating up the input through Transformer blocks? Does the hierarchical attention just specify the idiosyncratic lexical dependencies, or it leads to some syntactic structure-alike representation and primitive grammatical categories (Noun vs. Verb, NP vs. VP). If some form of syntactic representation can be found in the Transformer architecture, how does it contrasts to the syntax/constituent parse trees proposed in classic syntactic theory? As we see in this chapter, the contextualized representation might restrict the models in generalizing for totally novel lexical combinations, when the models were trained on the highly constraining artificial corpus. It is an open question what might happen if the corpus is more like the naturalistic language input, in which the semantic constraints are more irregular and diverse.

To summarize and elaborate, the diagnostic studies may help 'open the black box', elucidating what kind of process is happening and what structure and representation might be emerging in the neural-net architecture, in response to the lexical dependency-rich inputs. Since most of the state-of-the-art scaled-up language models are based on the Transformer architecture, the current study and the line of follow-up diagnostic research may further our understanding in the representational mechanisms leading to the phenomenal linguistic capabilities in the language models.

The second line of works are behavioral experiments concerning the interactive dynamics between syntax and semantics and its role in language development. Basing on the current results, I hypothesize that tracking lexical dependencies in the language input might trigger emergence of early syntactic ability (Transformer), while explicit syntactic representation may in turn benefit in development of semantic structure. To start with, a range of learning and generalization tasks concerning simple (pair-wise) and complex (multi-way) lexical dependencies can be designed and tested on children in different age groups. These studies should be simplification of the lexical-dependency experiments that have been run in adults and children (McRae, Spivey-Knowlton, & Tanenhaus, 1998; Rayner et al., 2004; Bicknell et al., 2010), or adoption of the similar experiments on children (Abu-Zhaya, Arnon, & Borovsky, 2022; Borovsky, 2022; Kueser, Peters, & Borovsky, 2022). The experiments may help carve out the age of children capturing multi-way lexical dependency in natural language, and correlational studies can align the semantic achievement against their syntactic development. While in its early stage, the behavioral attempts taken together with the modeling works may contribute to future exploration on the interactive dynamics between the syntax and semantics towards a full-fledged representation of meaning.

Chapter 7

General Discussion

In Chapter 3-6, I presented two groups of studies investigating the capabilities of the two distributional graphs (and other distributional model) in learning, representing and generalizing lexical relationships. These modeling endeavors are motivated by both evidences of lexical (knowledge) effect in language comprehension (Bicknell et al., 2010; Ferretti et al., 2001; McRae et al., 2005; Rayner et al., 2004; Trueswell et al., 1994) and theoretical interests in modeling semantic memory with language. In Section 7.1, I step back and relate the presented studies with a broader range of researches on knowledge representation and psycholinguistics. I show how the works of Distributional Graph modeling potentially bridge the field of knowledge representation/semantic memory with the field of language comprehension, towards a better understandings in the theory of meaning.

Similar to other distributional models, the Distributional Graph approach does not only form a knowledge representation, but also tracks how the representation have come into being, by encoding what type of inputs. In this sense, comparisons between distributional graphs with other distributional models may have profound and intriguing implications on theory of language acquisition and theory of semantic (knowledge) development. While the implications have been briefly discussed at the end of Chapter 6, I dig deeper into this topic in Section 7.2.

I will end this chapter by summarizing and listing a few future directions suggested by the dissertation studies. This dissertation focuses on two themes: (i) Representation of lexical relationship as the critical computational problem, and (ii) the Distributional Graph approach as the representational/algorithmic modeling 'solution'. The works in this dissertation encourage computational and behavioral investigations on lexical dependency, as well as empirical and theoretical research on developing Distributional Graph models. These prospective directions will be proposed in order in Section 7.3.

7.1 Bridging Knowledge and Language by Distributional Graph

7.1.1 Semantic Dependency as Knowledge in the World

Human categorizes. We categorize things and form concepts based on perceptual and more abstract cues (Abernethy, Campbell, & Faigman, 2005). We not only categorize things, but also form categories of actions in a flexible manner (Malt, Gennari, & Imai, 2010). Categorization helps us as well as other species parse the world, and master its regularity. Such regularity concerns the relationship between instances, e.g. two individual tigers (entities), two attempts of leap (action). Once categories are built and concepts are formed, it is possible to reveal the associations between the concepts (entities, feature and actions). Birds *fly*, mammals *walk*, and fishes *swim*. While exceptions happen from time to time, as birds do walk (in a clumsy manner), mammals can swim, and some fishes fly (e.g., flying fish), the world shows regularity in terms of how concepts are associated with other concepts, features, etc. A possible landscape (regularity) of such associations is illustrated in Figure 7.1a. Each row and column represents a lexicalized concept, and the color in the cell represents the graded association between the two concepts. As illustrated in the figure, some concepts may form categories based on the shared associations to other concepts. Such association between lexicalized concepts is what I refer to as 'lexical relationship' or 'lexical dependency' (the quantitative aspect of relationship) in this dissertation. As implemented in the world simulation in Chapter 3, as well as in reality, the lexical relationships reflect the association of concepts that interact with each other in the world.

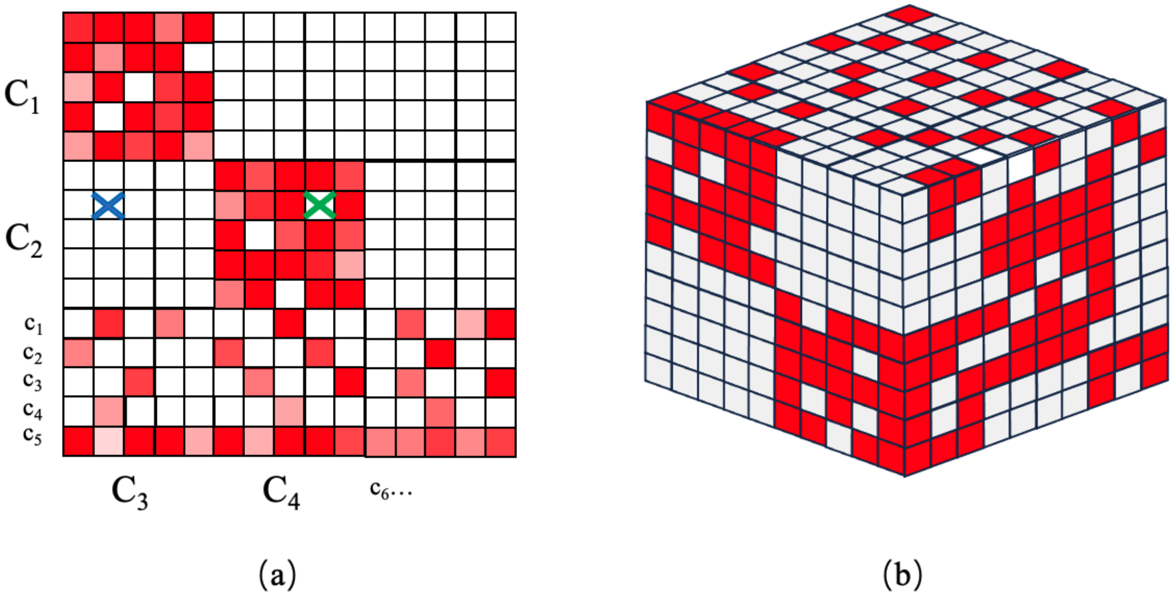


Figure 7.1: How concepts in the world may be associated. (a) Two-way lexical dependency/association: each row and column is a single lexicalized concept, and colored cell reflect the association between the corresponding concepts. (b) Three-way lexical dependency: each cell represent the association between three concepts

By attending to and representing these associations, we can organize categories/concepts in a knowledge structure (Collins & Quillian, 1969; Deese, 1962; Osgood, 1952; McRae et al., 1997; Smith et al., 1974). Such a knowledge structure may benefit us in two ways that potentially leads to new knowledge. First, by associating the two concepts bearing the lexical relation, it may associate a group of concepts and features that leads to new findings. For example, considering associating BIRD and FLY, it simultaneously brings in the semantics of BIRD and FLY, including the features of the animal and the manner of the action. These details in the features and manners may inform on the question such as what features may lead to the capability of flying, e.g. having a beak or having feathers. Answers to such questions may elicit a second type of novel knowledge, e.g. generalization to new plausible lexical relations. For example, suppose the green cross in Figure 7.1 refers to the unknown relationship between a certain type of bird and the action FLY, while the blue cross refers to the unknown relationship between a the same type of bird and the action SWIM. If a system may attend to other features of this type of bird, as well as the features of other types of birds, then it is likely to infer that this bird might fly but not swim. As readers might have noticed, this landscape is the type of semantic constraint that I have used to test models in Chapter 3, 5 and 6.

To summarize, the world seems to embed regularity. We handle the regularity among instances by categorization and forming concepts. Once concepts are formed, it is essential to notice the systematic associations among the concepts. An effective representation of such association may not only help us interpret what we have observed, but also generate novel knowledge to predict plausible, but unseen events. Throughout this section, what I emphasize is the structure and regularity of the world and the potential benefits of forming knowledge reflecting such regularity. But how can we master these complex world knowledge? There is a powerful tool: Language.

7.1.2 Lexical Dependency as Knowledge in Language

Language helps categorization. The linguistic labels play an essential role in the early phase of word leaning (Perszyk & Waxman, 2018), and they help grown-up form categorizes (Lupyan, Rakison, & McClelland, 2007). This is probably the most straightforward Language's contribution to constructing knowledge. What I would like to emphasize though, are the crucial roles of Language on top of labeling categories: (i) Phrases and sentences serve as pointer to the lexical associations, (ii) Linguistic structures help us to capture complex lexical relations and generate novel concepts. In this sense, language and knowledge deeply intertwine with each other leading to co-evolution of the two systems. Historically, more attention (in the field of psycholinguistics) has been paid to the direction that world (lexical) knowledge matters to language processing. Here, I emphasize the other direction: The **structure and content** of language gives rise to more complex and established knowledge representation, and as a result, the linguistically informed knowledge representation may reciprocate language processing.

Crucial Roles of Language in Forming Knowledge

As argued above, rich world knowledge may reside in the 'lexicalized' associations between concepts. With language, we do not just produce a single concept label or think of one concept. We can think of, and talk about events, which bind multiple concepts, including entities, actions and features. Language may help us attend to and effectively represent and use those associations. This is reflected in early semantic development (Arias-Trejo & Plunkett, 2009, 2013) and early language development benefiting from distributional linguistic cues (Lany & Saffran, 2011; Willits, Seidenberg, & Saffran, 2014; Scott & Fisher, 2009). In other words, by producing, hearing, and thinking of multiple words (concepts) organized in an event structure, we may accumulate lexical association (co-occurrence) statistics, and this process potentially leads to mental representations illustrated in Figure 7.1.

Aside from the association brought by the mere (linguistic) co-occurrence, Language provides the structure that explicates how multiple (more than two) lexical concepts are associated within complex linguistic expressions (phrase and sentence). For example, in the sentence *A mechanic checks engines, while a journalist checks spelling*, the grammatical structure helps us specify relationships such as *mechanic - check engine* and *journalist - check spelling.* In other words, the structures in these complex expressions specifies how lexical concepts could be **chunked** and then organized to reflect their (multi-way) relationships. As I have shown in Chapter 5 and 6, the chunking (forming some structure) plays a huge role in effective representation of complex lexical dependencies and success in the challenging compositional generalization task. I will discuss with more details on the nature of the structure formed in this process in the following sections. The point here is that such structural information is embedded in language, and may contribute to knowledge representation if they can be effectively encoded (as in the CTN and Transformer model, and also the unknown computational algorithm of human language processing).

Finally, the linguistic structure and the generative nature of Language may give rise to novel concepts. An **employer** is a person or an institute that hires labors working for the sake of profits. **Marathon** is a sport in which athletes run 42195 meters in memory of a Greek messenger. These concepts are described and defined with language, and I argue that Language is the most efficient, and probably the only effective way to generate such semantically involved concepts. From another perspective, it is based on the generative nature of Language that we can create novel lexical concepts as a linguistic formulation of known concepts.

Knowledge and Language Need to be Brought Together

Given the significance of Language in forming knowledge, the field of cognitive psychology though has focused more on the other side. Psycholinguists have long been noticed that lexical knowledge (world knowledge) is crucial to language processing (McClelland et al., 1989; Trueswell et al., 1993, 1994; Garnsey et al., 1997; Kamide et al., 2003; Bicknell et al., 2010; Willits et al., 2015). Nevertheless, the role of Language, especially the linguistic structure of language has been overlooked by earlier semantic memory/knowledge representation researchers. The distributional semantic model approach, started from early 1990s, have shown the value of language in representing semantic memory (Landauer & Dumais, 1997; Lund & Burgess, 1996; Jones & Mewhort, 2007; Griffiths et al., 2007; Mikolov et al., 2013; Pennington, Socher, & Manning, 2014). However, as I have argued and showed with studies in previous chapters, these modeling attempts have troubles effectively preserving the structural (grammatical) information in language, and as a result they struggled in challenging language tasks, such as learning and generalizing multi-way lexical dependencies.

In contrast, while LLMs trained exclusively on linguistic input have shown extraordinary capabilities in various language and semantic tasks, the end-to-end architecture have challenged investigation of their internal representation leading to the success. In the next section, based on the findings in the previous chapters, I argue why Distributional Graph can be considered as a promising approach of linking Knowledge and Language towards better semantic representations.

7.1.3 Distributional Graph as a Proposal for Bridging Knowledge and Language

All distributional models process on linguistic input. A crucial capability in distributional models is to effectively extract and represent the structural information embedded in language that reflects (simple and complex) lexical relationships. Two types of relationships are relatively challenging: the indirect relations (leading to generalization), and the multi-way relations (leading to compositional generalization). In contrast to other distributional models, distributional graphs structure concepts in a network topology, and access relatedness with network metrics. As I have shown in the previous chapters, the graphical topology provides a structural prerequisite for representing lexical dependency, and the spreading-activation algorithm proves to be an effective functional process to capture multiple distributional features that affect semantic relatedness. These structural and processing characteristics makes Distributional Graph a relatively successful model for representing indirect and multi-way lexical dependencies, as well as semantic relationships in general.

Indirect relations

Despite modeling details, every distributional model primarily encodes co-occurrence of words. A critical demand on distributional model is to capture the indirect relations, i.e. the relation between words/expressions that do not occur together. The most common indirect relation is paradigmatic relationship (similarity), which earlier distributional models, e.g. LSA and HAL, can easily model (Landauer & Dumais, 1997). However, in these vector space models, encoding and evaluation on paradigmatic relations usually come as a cost of the representation of the primitive syntagmatic relations. As a result, it is not easy for these models to evaluate more indirect syntagmatic relations which binds the primitive syntagmatic relations and the paradigmatic relations.

As shown in Chapter 3, what distinguishes Distributional Graph and the spatial models is the graphical topology that preserves all orders of relations in the same structure. With the spreading-activation algorithm, the LON model can simultaneously evaluate direct and indirect relations of various orders. Importantly, the evaluations on the indirect relations are not at the cost of those on more direct relations. Furthermore, I showed in Chapter 4 that there is a quasi equivalence between the spreading-activation process on a graph and different 'orders' of cosine similarities in the corresponding vector space representation. Such equivalence implies that the graphical topology may provide a representation that potentially better preserves lexical association embedded in language input. The representation of indirect relation may in turn provides support for generalization on word-word relations.

Multi-way Dependency

While 'connecting language' provides the infrastructure for evaluating indirect relations, the graphical topology also benefits in representation of multi-way lexical dependency. As shown in Chapter 5, the constituent structure effectively helps chunking the relevant lexical items (*preserve, cucumber, vinegar*). By explicit connecting different constituent structures into one graph, lexical dependencies are reflected in graphical metrics such as graphical distances, etc., which can be effectively evaluated by the spreading-activation algorithm.

Through further investigations in Chapter 6, I find that the constituent structure is not the only form leading to effective representation of complex dependencies. As long as the critical relations are encoded with edge links, LON (with window size 2) can capture the multi-way lexical dependency. Furthermore, such representation are not constrained to explicit graphical representations. I showed that Transformer models can achieve in the critical compositional generalization task. Following-up analysis suggests that some quasi semantic-syntactic structure might have been formed in the Transformer model, based on its self-attention mechanism. From there, I argue that the exact form of the representation might be less important. What essential in a model or any computational system is a mechanism that ensures chunking words and forming some structure that helps specifying lexical dependencies. No matter it is by self-attention in a Transformer block, or an explicit syntactic parse tree, *preserve* and *cucumber* needed to be bound as a whole to associate with *vinegar*. Furthermore, such binding needs to be flexible so that the system can form representation compositionally (combinatorially), and generalize from learned knowledge to novel experience (*preserve pepper with vinegar*).

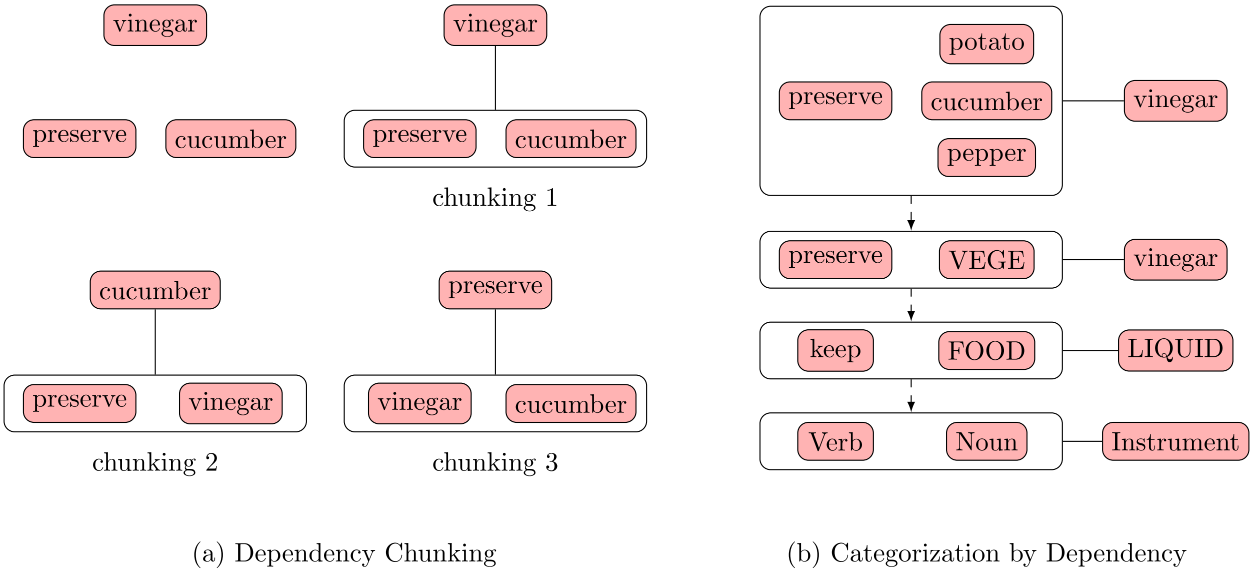
In this sense, what is the value of Distributional Graph? While both the mini GPT-2 model and the CTN model achieved in the compositional generalization tasks, the result showed that mini GPT-2's performance was more sensitive to the input. More importantly, the representation in Distributional Graph is much more transparent compared to that in the Transformer architecture. Therefore, despite the similar performance, Distributioanl Graph models may provide a clearer 'prototype' that better illustrates the 'computational problem' (Marr, 1982). What is the nature of the hard semantic problem concerning representing complex lexical relationships? What kind of representational data structure may lead to the success of the challenging compositional generalization task? The Distributional Graph approach is promising in the way that it provides an explicit high level semantic representation and an effective process on the structure, that potentially underlines the solutions to the hard semantic tasks.

As a summary, the Distributional Graph approach encodes lexical relations from linguistic input in a network topology with fewer losses of the structural information embedded in the language. Consequently, language tasks that demand more on linguistically structured knowledge (e.g., compositional generalization) can be informed by the DG representation that preserves more linguistic structures. In this way, Distributional Graph can be considered as a potentially valuable attempt to bridge knowledge representation and language comprehension.

7.2 Capturing Lexical Dependencies Towards Semantic and Language Development

In Chapter 5, I showed that the constituent structure contributed to the success in the compositional (multi-way) generalization task. In Chapter 6, I demonstrated the critical factor for such generalization is not necessarily the constituent structure *per se*, but any structure that effectively captures the lexical dependency pattern in the language input. Based on the results in Chapter 6, I speculate such effective structure may emerge in the Transformer-based mini GPT-2 model, leading to its success in the critical tasks. Since the mini GPT-2 model directly processes on raw linguistic data, its success (as a contrast to the success in CTN) has an intriguing implication on language learning. In this section, I first delve into possible emerging structures in the process of learning lexical dependencies, and then discuss how the emerging structure might inform on language development.

7.2.1 Emerging Structure



(a) Dependency Chunking

(b) Categorization by Dependency

Figure 7.2: Structures emerge by lexical dependencies. (a) Possible chunks formed to reflect dependencies among three words. (b) Categorization with common lexical dependency as a source.

In the artificial corpus used in Chapter 5 and 6, there is a three-way dependency in the trio *preserve, cucumber, vinegar*. To succeed in the critical compositional generalization task, a model needs to chunk the three words in some way, e.g., it needs to bind *preserve* and *cucumber* to decide on *vinegar*. Such chunking is illustrated in chunking 1 of Figure 7.2a, top right. Notice that there are potentially multiple ways to chunk the three words. For the corpus used in the dissertation, chunking 1 and 3 might be more effective compared to chunking 2, as the pairs in chunking 1 and 3 can perfectly predict the third item in the trio, while the prediction by chunking 2 is less certain. More formally, the conditional probability of *vinegar* or *preserve* given *preserve cucumber* or *vinegar, cucumber* is 1, while the probability of *cucumber* given *preserve, vinegar* is 0.5. In general, how a distributional model chunks the items may depend on the lexical dependencies in the input, and the training task. While there are different chunking options, powerful models should be able to chunk words in a way that better reflects the lexical dependency pattern in the input language.

What exactly are these structures formed by chunking? From the first sight, it is straightforward that the structures in 7.2a are semantic structures reflecting multi-way lexical dependency. However, if we zoom out and consider all structures formed in a given corpus, these 'local' dependencies might be further clustered into a hierarchy of structural dependencies featuring different level of abstraction. This is illustrated in 7.2b. On the top, we see that *preserve* is bound to *cucumber, potato*, and also indirectly to *pepper*. Collectively, the larger chunk has dependency with *vinegar*. This cluster of dependencies can be abstracted as the dependency between *preserve* VEGETABLE and *vinegar*. Furthermore, imagine in a larger corpus, the dependency can be further abstracted, so that *preserve* is extended to the 'light' verb *keep* with a general sense of *preserving*, VEGETABLE and *vinegar* extended to their super-ordinate category FOOD and LIQUID. At the end, the dependency is abstracted to the level of grammatical categories like Verb, Noun and Instrument.

Different levels of abstraction, once formed may benefit the representational system realizing different goals. The lexical level representation specify the exactly dependencies between lexical items. The dependency between verb and semantic category may facilitate generalization on the respective abstracted level. For example, with the representation of *preserve* - VEGETABLE, a model is able to generalize from preserving observed vegetables (*cucumber*) to preserving unobserved vegetables (*pepper*). Finally, the grammatical category level abstraction may benefit syntactic judgement of the sentences.

From a more traditional perspective, the different levels of abstraction might have completely different origins. A critical point in this dissertation, though, is that multiple levels of dependency structures can be formed based on the process of distributional learning. Huebner and Willits (2018) showed that distributional models can capture linguistic structures, from the higher level syntactic category, e.g. Noun vs. Verb, down to the semantic taxonomies. Furthermore, in Chapter 5, I showed that the high-level syntactic structure (syntax tree), once lexically specified (constituent tree), becomes an effective representation for challenging semantic tasks. This indicates that there are overlaps between the syntactic structure, and semantic structures tailored for lexical dependencies. The structure speculated in mini GPT-2 in Chapter 6 is between the lexical level the syntactic level. These modeling works imply that, there are rich dependency patterns in the linguistic data, spanning the abstraction spectrum from lexical semantics to syntactic categories. These dependencies, once picked up by effective models, may enable achievements in hard semantic tasks. Taking a step further, if similar computational mechanism exists in human mind, it is possible that the lexical dependency patterns, especially those between multiple words, can be an essential source for human to learn semantics and syntax.

7.2.2 Co-evolving Semantics and Syntax out of Lexical Dependencies

The Transformer investigated in Chapter 6 and the CTN model proposed in this dissertation suggest two stages of semantic development in human learner. For a Transformer model in its early stage of training, it is not endowed with syntactic parsing capability and it processes on raw context. In this case, it mirrors human learners not sufficiently proficient in syntax. It is possible that similar to Transformer, young learners form quasi semantic-syntactic representations of their language input. These quasi representations might not faithfully reflect the meanings of the input, so that the learners may find it difficult to interpret sentences with complex semantic dependencies. In terms of development of semantic memory and knowledge representation, the concepts learned in this stage should have simpler lexical/semantic dependencies. In this stage, development of syntax and semantics may reciprocate each other. Over time, the Transformer gradually learns the lexical dependencies in the language input, so that they can better grasp the structures in novel input (e.g. *preserve pepper with vinegar*). In this stage, the behavior of the Transformer model might become similar to that of the CTN, despite that they are implemented differently.

The CTN model is trained on parsed sentences, taking advantage of the structured input. In this sense, the model reflects human learners who have achieved (near) adult-level syntactic competence. In this second stage, they can form correct structures of their language input, and effectively judge on sentences with complex semantic dependencies. As a result, in this stage, learners should be able to integrate more complex lexical/semantic dependencies into their knowledge/semantic structure. Since their syntax proficiency is almost at ceiling, the development becomes one-sided. Learners should benefit from their syntax maturity and they are prepared to gain knowledge through reading. To be more specific, they are able to take in knowledge that involve sophisticated semantic relationships, written in long and hard sentences. The suggested correspondences between the two models and two development stages encourages a few threads of future directions, which I propose with details in the next section. Here, I would like to emphasize more on the first stage development.

From a developmental perspective, the CTN by itself may not account for the emergence of early language skills, especially syntax acquisition in young children. While the focus on CTN concerns semantic capability after rudimentary parsing abilities are in place, it is important to ask where syntax comes from. To firmly place the CTN within the field of developmental psychology will take time, and cross-domain collaboration between cognitive modelers and language acquisition researchers working at the interface between early syntax and semantics. While there is a large literature that has examined the emergence of syntax in young learners, much more work is needed to relate such findings to semantic representation. That is, how the less mature semantics and syntax systems co-evolve based on the various inputs (linguistic and non-linguistic). To be more specific, how might the incremental emergence of syntax, e.g. the ability to parse linguistic expressions into constituents, influence how semantic knowledge is represented in a child and in distributional models?

One approach is to study the emergence of syntax from a connectionist perspective, which starts with relatively little to no linguistic knowledge, and is aided by gradual emergence of the ability to parse language input over the course of language experience. As suggested by the performance of mini GPT-2 in Chapter 6 and previous studies (Huebner & Willits, 2018, 2021b), the process might involve categorization on linguistic inputs in a scaffolding manner, illustrated in Figure 7.3. In the early stage, learners or models may learn from more globally effective lexical dependencies, e.g. those between articles and nouns, and form coarse grammatical categories on the dependency landscape (Figure 7.3a). After the basic grammatical categories are acquired, learners may further specify more local idiosyncratic dependencies and form finer (semantic) categories (Figure 7.3b). Alternatively, the most local lexical-level dependency might be captured before global pattern (Goldberg, 1995), or the two processes happen at the same time. While how the learning unfolds await more behavioral and computational endeavors, I emphasize that the interactive semantic and syntactic development can be informed by the process of learning lexical dependency in language input.

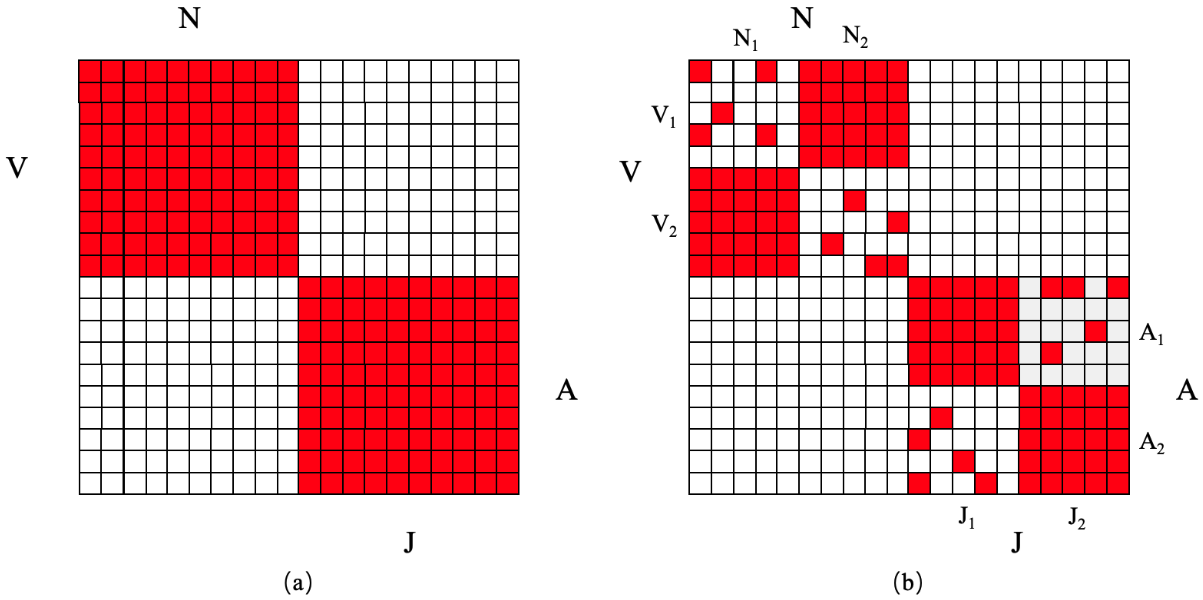


Figure 7.3: Categorization based on lexical dependency in a scaffolding manner. (a) Grammatical categories, e.g. Noun, Verb, Adjective, Adverb, are formed based on more straightforward global distributional patterns. (b) Finer local categories, e.g., N_1, N_2 are formed.

7.3 Future Directions

In this dissertation, I developed a type of distributional semantic models: Distributional Graph, which forms knowledge structures by encoding linguistic distribution pattern in a network data structure. While this dissertation has focused on formal investigations in the computational mechanisms and capabilities of Distributional Graphs and related models, the results encourage a wide range of theoretical and empirical works. First, more behavioral works testing the effect of indirect semantic relations, as well as compositional generalization in language processing are needed to ground the modeling results in Chapter 3 and 5. Furthermore, as mentioned in the last section, the success of the CTN and Transformer model suggests a two-stage semantic development modulated by syntactic proficiency, which leads to behavioral experiments for testing the hypothesis. In turn, the behavioral findings need to be compared with model simulations. While existing distributional models are readily implemented for simulating human behaviors, substantial works are needed for Distributional Graph. This leads to multiple lines of theoretical and empirical modeling studies on the graphical model. Finally, I propose an idea of generalizing the modeling gist in Distributional Graph for a more comprehensive representation of knowledge and semantic structure.

7.3.1 Behavioral Works

Language Processing in Adults

The results in Chapter 3 showed that Distributional Graphs is better at differentiated relatedness varied by abstraction orders. If Distributional Graph is a plausible representation of human semantic memory, then people should be able to differentiate direct and indirect lexical pairs predicted by Distributional Graph relatedness measure. Previous works have shown that people are sensitive to indirect semantic relations in priming (Balota & Lorch, 1986), and that network distance (on semantic network built from normative association) between word pairs predicts human semantic (similarity) judgement (De Deyne et al., 2016; Kenett et al., 2017; Kumar et al., 2019; Rotaru et al., 2018). With the proposal of Distributional Graph, more behavioral works can be added to the literature. Distributional graphs do not conflate syntagmatic and paradigmatic relationships. Therefore, it is possible to create syntagmatic and paradigmatic stimuli sets (consisting of pairs that are all syntagmatically related, or paradigmatically related) based on how the word pairs are connected in the distributional graph. Then, the syntagmatic and paradigmatic pairs can be used to correlate human's semantic judgement and model (by Distributional graph and other semantic models) predictions. Parallel to the results in Chapter 3, if human's relatedness judgements on syntagmatic (paradigmatic) pairs can be predicted by the corresponding relatedness in a distributional graph, then the proposed DG structure could be a plausible representation of human semantic memory.

In terms of multi-way lexical dependency and compositional generalization, the modeling works was motivated by a series precedent behavioral experiments (Bicknell et al., 2010; Ferretti et al., 2001; Kamide et al., 2003; McRae et al., 2005; Rayner et al., 2004). These works have shown that people do represent multi-way lexical dependencies and use them when comprehending language. The modeling work in this dissertation suggests that distributional cues can be useful for generalizing multi-way dependencies. If the mechanism the CTN employs to succeed in the evaluation of compositional generalization is informative about the processes that takes place in the human language system, it should be possible to confirm the connection with behavioral experiments. A procedure for doing so might involve eye-tracking or EEG methods to quantify the semantic facilitation of a verb phrase on an expected instrument at the end of the sentence. For the expected instrument *knife*, a comparison between the facilitation effect of *cut cake* (canonical), *cut cookie* (similar), *cut blicket* (pseudo) would be informative about whether participants can encode a phrasal dependency (e.g. between*cut cookie* and *knife*) and use it to interpret a novel phrase (e.g. *cut cookie with knife*). In this proposed experiment, the familiarity of a verb phrase, e.g. the extent to which people are familiar with *cut cake* and *cut cookie* can be quantified with corpus data.

Two-stage Semantic Development

Results in Chapter 5 and 6 suggest that semantic development might be modulated by syntactic proficiency. Before the full-fledged syntax, the semantic structure should be dominated with simpler relations (between two words, or two concepts), and the proportion of multi-way lexical dependencies should be relatively low. One way to approach to the structural complexity of developing semantic representations is to examine the relationship between words from different grammatical classes in young children. For example, Kueser, Horvath, and Borovsky (2023) constructed semantic network of nouns and verbs exposed to 16-30 months old to study the effect of interactions between the two classes of words in early word learning. Similar approach can be extended to more word classes, e.g. adjectives and adverbs, and a broader range of ages. Network analysis can be used to reveal how lexical dependency within and between different word classes develops over time. The two-stage hypothesis predicts that relationships between different word classes should develop relatively later in comparison to within classes relations. Emergence of between classes relationships should correlates with the process of syntactic maturation.

Another prediction by the two-stage hypothesis is that the sensitivity to the multi-way lexical dependencies presented in the language input alongside with their syntactic proficiency. Existing works have shown that young children (5 to 10 years old) can learn multi-way dependencies, and generalize the learned dependencies to novel lexical combinations (Abu-Zhaya et al., 2022; Borovsky, 2022). Future works can replicate these studies, while correlating the performance in learning and generalizing multi-way lexical dependencies to syntactic readiness and complexity of semantic structure.

7.3.2 Modeling Works Concerning Distributional Graph

Some of the behavioral works proposed in the last section require comparing human data with model predicted relatedness. In this case, it requires constructing distributional graphs from naturalistic linguistic data. To proceed for that goal, it is necessary to first thoroughly investigate the capability of distributional graphs on more semantic tasks. Then, a series of theoretical and empirical research are needed as preparation for constructing distributional graphs from naturalistic input.

Computational Capability and Mechanism in Various Distributional Models

In Chapter 3, 5 and 6, the critical findings were limited to a single type of semantic relation (i.e. indirect syntagmatic relations). A more complete understanding of the distributional graphs, as well as other distributional models (for comparison) requires considerable amount of works on additional tasks. For instance, while I examined the ability of the CTN to infer the relatedness between a phrase and a word, I did not systematically examine its ability to infer the relatedness between individual lexical items. Nor did I investigate how performance might differ depending on the type of relation, namely syntagmatic relation or paradigmatic relation. Especially, while I claim that successful representation of the indirect syntagmatic relationships requires successful representation of paradigmatic relations, no task in this dissertation directly evaluated paradigmatic relationships, e.g. similarity between structurally isomorphic phrases. The performance of distributional graphs and other models can differ substantially by the type of relation being examined. In future studies, more distributional models should be tested against distributional graphs on both syntagmatic and paradigmatic relationships. Having separate experiments on different types of relations may explicate the advantage and disadvantage of different distributional models in semantic representation.

Scale up Distributional Graph with Naturalistic Language Input

Throughout this dissertation, all models are trained on toy artificial corpora. While the artificial corpora are necessary concerning the goal of formal investigation in the current studies, the proposed distributional graphs need to be trained on naturalistic corpus to account for data in behavioral experiments, and to compare with established distributional models. Therefore, an important future direction is to train CTN on the real-world language data, and examine whether the model maintain its capability in various semantic tasks.

Nevertheless, there are considerable challenges for scaling up the ‘connecting language’ approach. First, unlike artificial corpus, naturalistic data involves enormous noises. To apply the model with real-world input, it is necessary to (i) develop effective pre-processing algorithms to clean up the data and (ii) understand the possible effects of the noises on the topology of the graphical structure. These noises includes punctuation, inflection, and function words, to name a few. While it would be straightforward to swipe out these unwanted components in the naturalistic input, more caution is needed as it has been shown that functional components such as tense and aspect may affect behaviors in semantic tasks (Willits et al., 2015). Additionally, the CTN model implemented in the current studies works on parsed binary constituent trees, which are manually generated as the artificial corpora are simple. However, naturalistic corpora are often noisy and complex, so that a primary concern is to first find an effective automatic parser to parse the sentences. Nevertheless, even the state-of-the-art parsing algorithms are not perfectly accurate. Therefore it is important to know the effect of parse error on the resultant semantic network. In addition, accessing to semantic relatedness in the network structure is computationally complex, and the complexity scales up alongside with the size of the corpus. As a result, another preparation before applying naturalistic input is to design a computationally plausible algorithm to implement the spreading-activation measure on large-scale networks. In summary, while scaling up the Distributional Graph models to naturalistic corpus is necessary, more theoretical and empirical works are needed to show that the movement is plausible and practical.

7.3.3 Generalize Distributional Graph for Broader Knowledge Representation

Lastly, I present some very high-level and long-term visions that expands on the Distributional Graph approach. Rather than being inspired by the studies in this dissertation, these high-level goals are the initiatives of the Distributional Graph endeavor. While the current distributional graphs encode mere word co-occurrence, a broader range of semantic relations can be encoded in the edges of the network structures. These relations may explicitly encode useful syntactic and semantic knowledge and can be accessed through the spreading-activation algorithm. The enriched encoding may lead to a mechanism which potentially produces ‘generative concept’. In turn, the ‘generative concept’ mechanism may inform on the developing knowledge structure in both human individuals and humanity as a whole. I will present the visions primarily from a modeling perspective and then stress on the significance of the modeling endeavors to scientific research in cognitive psychology.

Beyond Word Co-occurrence

Although the initial modeling efforts in Distributional Graph have focused on lexical relationship, the nature of the relations encoded in the semantic networks is essentially co-occurrence frequency in linguistic corpus. As non-linguistic information can be essential building blocks of semantic memory, the current distributional graphs are profoundly limited by its ignorance of non-linguistic modalities. Recent studies have shown that, similar to linguistic corpus, there are robust co-occurrence structure among real world visual objects (Greene, 2013; Sadeghi, McClelland, & Hoffman, 2015), and human form association between frequently co-occurring visual objects and scenes (Bonner & Epstein, 2020; Zettersten, Wojcik, Benitez, & Saffran, 2018). Furthermore, computational modeling of adult semantic structure suggests significant contribution from multi-modal information (Deyne et al., 2021), and developmental works also indicate that in addition to conceptual or linguistic features, perceptual cues play an independent and presumably more important role in early word learning and semantic development (Hills, Maouene, Maouene, Sheya, & Smith, 2009; Peters, Kueser, & Borovsky, 2021; Seidl, Indarjit, & Borovsky, 2023). These computational and behavioral works motivate incorporating non-linguistic distributional input alongside with linguistic corpus. It has long been noticed that there is a systematic difference between what people know and what they produce in language. By integrating multi-modal inputs, the generalized distributional graphs may better account for behavioral data in adult semantic processing, and brings more insight to the theory of language development, e.g. verb learning.

More importantly, unlike the classic semantic networks (Collins & Quillian, 1969; Collins & Loftus, 1975), edges in the presented distributional graphs do not encode any content information. They are only structural constructs to layout the topological relations between concept nodes. One may argue that the constituent trees in the CTN model bear syntactic information and significantly contribute to the representational capability of the model. However, that structural information is in the topological form of the trees, and there is no specific information on each individual edge that links the node pair. In this way, while the spreading-activation algorithm can be used to evaluate quantitative relatedness between any two nodes in the network, no qualitative or formal relations can be accessed in the current distributional graphs.

Similar to the classic semantic networks, information can be directly encoded in the edges of the distributional graphs. For example, after integrating non-linguistic input, it might be necessary to create multi-edges between each single node pairs, with each edge labeling the type of relation, e.g. visual co-occurrence, linguistic co-occurrence, etc. In this way, it is possible to access and integrate multiple types of relations through the activation process.

There is one type of linguistic information that can be directly labeled in the CTN, which may lead to more powerful semantic representation. For each complex expression, the sub-tree containing the expression and its constituents specifies a lexical composition. For example, the sub-tree containing *preserve cucumber*, *preserve* and *cucumber* in a constituent tree network specifies that the lexical item *cucumber* can be combined (composed) with the lexical item *preserve* to form the phrase *preserve cucumber*. Such constituting relations can be encoded by a type algebra used in formal semantics (Montague, 1970). To be more specific, types are assigned to the constituting nodes that specifies who is the function (predicate) and who is the argument. In the case of *preserve cucumber*, it can be specified that *preserve* is the function (predicate) and *cucumber* is the argument. Then, the spreading-activation accessing to the sub-tree can elicit the semantic memory for the lexical argumentation of *preserve cucumber*. Notice that the argumentation here is different from the three-way relationship aimed at in the previous chapters. In those chapters, the evaluated relation only concerns how closely the targeting word pairs/triplets are related. After encoding the argumentation information on the edges, the spreading activation can not only evaluate how far *preserve* is from *cucumber*, but also whether they can be composed to form a grammatically valid chunk (a phrase). Recall a feature of distributional graph is that it may evaluate the relatedness between observed lexical combinations as well as novel lexical combinations. In this sense, the encoding of argumentation information results in the evaluation of novel expressions such as *preserve pepper* and *soundly sleeping colorless green idea* in terms of both their plausibility and grammaticality. This leads to the potential of generating **novel** concepts in the CTN model. I explain it with more details below.

Generative Concept

In Chapter 5 and 6, the models were tested on relatedness between novel lexical combination, e.g. *preserve pepper - vinegar*. The relation was 'novel' as *preserve pepper* and *vinegar* were not associated in the input. From another perspective, the evaluation was also about the plausibility of the novel expression, i.e. *preserve pepper with vinegar*, a phrase that had never occurred in the input. Indeed, all studies in this dissertation essentially target at the plausibility of novel expressions, i.e. novel combinations of represented words. However, a complex expression is not merely a set of words: *preserve pepper with vinegar* is roughly ok, but *vinegar pepper preserve with* is not. To make sense of an expression, we have to confirm its 'grammaticality' in the first place, before judging its 'plausibility'. With the encoding of the argumentation information on the edges of CTN, it might be possible to judge on both the grammaticality and semanticality of a given expression. These judgements can be considered as parts of the comprehension process. For the other direction, the CTN model might help construct grammatical and plausible expressions/concepts from scratch. For example, given an arbitrary set of words, the spreading activation process may access their grammatical information to form candidate structures. Then the model may judge the semantic plausibility of the structures based on the activation based relatedness, to select relatively meaningful constructs out of the list.

While a substantial amount of theoretical and technical problems are needed to work out, the CTN model (and potentially other distributional graphs that effectively encodes grammatical information) provides a possibility to generate meaningful novel concepts. For example, given *baby sleep*, it might generate a sentence like *A cute baby sleeps soundly* or *Soundly sleeping baby.* The approach distinguishes itself from generative grammar (Chomsky, 1965) that generates grammatical yet meaningless expressions (*Colorless green ideas sleep furiously.*) It also contrasts with state-of-the-art LLMs in the way that the generative process in Distributioanl Graph can be maximally transparent and interpretable. The grammatical information are explicitly encoded in the network and the lexical relatedness are evaluated by the spreading-activation.

However, concept generation in distributional graphs are more likely to be unconstrained. In other words, it would be a great challenge for distributional graphs to function like the state-of-the-art language models, performing tasks in response to prompts. Essentially, Distributional Graph is a model for semantic memory representation. The model provides knowledge that may facilitate understanding, but it does not comprehend language and perform tasks by itself. Regardless of these limitations, the unconstrained concept and expression generation in distributional graph would still be valuable. The meaningful concepts generated from the unconstrained process can be added to the distributional graph itself. In this way, the distributional graph may not be a static knowledge representation. It is possible to become a developing and updating knowledge structure. This feature of distributional graph may shed light on knowledge development in human.

Towards a Developing Knowledge Structure in Human

For both human individuals and the human society as a whole, we do not just learn a fix amount of facts. We make new findings, generate new ideas, and create new concepts all the time. While the new knowledge may come from multiple sources, a great portion of them are linguistic constructs, e.g., **Marathon, Structuralism, the second Newton's law**, and they need to be formulated in language. From time to time, we create new words to lexicalize concepts expressed by complex phrases, and we use complex expressions to describe and explain the new ideas and concepts. Despite the exact content of new knowledge or concepts, they are likely to be described or defined with novel combination of familiar words or phrases. An interesting question arises based on the observation above: what knowledge structure may facilitate the generation of new concepts and knowledge? To be more specific, what features in semantic structures may lead to creation of new thoughts and ideas? There have been works showing that lexical semantic structure in people correlates with their creativity (Kenett, Anaki, & Faust, 2014; Kenett et al., 2016). With more established information and linguistic structures encoded in distributional graphs, and the concept generation mechanism of the model, it is possible to delve into more detailed mechanisms that potentially gives rise to creativity. To be more specific, we may manipulate the the structure or content of the input, and then test the resultant distributional graphs on the concepts/expressions they generate. The generated concepts can be evaluated internally by the models themselves, as well as by humans with meaningfulness and creativity norming. Importantly, since the distributional graphs are mechanistically transparent, it is possible to know what features of the network model or that in the input might have led to the different performances. In this way, such simulation studies might inform on creativity and knowledge development in human.

As a final remark for the generalizing Distributional Graph proposal, while I have described most of the works from the computational modeling perspective, the research can be fundamentally related to cognitive psychology. First of all, the elements and procedure in model construction are grounded by psychological realities: multi-modal inputs and grammatical information are both essential for constructing semantic/knowledge structure. Second, the goal of the modeling effort is also psychologically oriented. By enriching the model encoding with input critical to human knowledge development, the proposal aims at a semantic structure that can productively generate novel meaningful and grammatical expressions. Most importantly, the model is supposed to specify how do each part of the model architecture, the processing mechanism, and the input statistics give rise to the performances. In other words, the model provides a representational account for the (prospective) behaviors. The emphasis on the representational account for behaviors resonates with the general merit of cognitive psychology.

The proposal here humbly relates to the innovation of digital computer seven decades ago. Initially, the 'modeling' practices were totally practical: it targeted at automatizing sophisticated scientific computations. However, the creation of the computer in turn affected attitude of related scholars toward psychology. After the 'mental representation' that systematically maps to behaviors in the digital machine had been explicitly realized, curiosity on mental processes in human mind started to grow. Similarly, it will be a considerable computational challenge to build a transparent representation that productively generates novel grammatical and meaningful concepts. However, a model achieving the goal may elicit more psychological interests on the mental processes and representations in human mind that gives rise to the cognitive feat. In turn, these understanding in knowledge and meaning representation may help us create more semantically competent machines, as well as develop a more knowledgeable human scoiety.

Chapter 8

Conclusion

Explaining human's sophisticated semantic behaviors has been one of the most challenging, and at the meantime most intriguing topic in cognitive psychology. To be more specific, the behaviors include capturing indirect semantic relations, as well as learning multi-way lexical dependencies and generalize the dependencies to novel lexical combinations. These semantic capacities potentially give arise to the productively generated novel expressions and concepts in human. While the field has long recognized the role of world (lexical) knowledge in sophisticated language processing, less attention had been paid to incorporating linguistic input for constructing knowledge representation. In this dissertation, I brought **Language** and **Knowledge** together by developing a linguistically informed semantic/knowledge representation, i.e., **Distributional Graph**, that may better address sophisticated semantic/language tasks.

The distributional graph models transform linguistic input into graphical forms, and then join the linguistic graphlets into a semantic network. In this way, the model encodes distributional (semantic) relationship between the lexical units (words and phrases) in a graphical topology, which can be accessed through network metrics. Inspired by previous works (Collins & Loftus, 1975; Anderson, 1983; De Deyne et al., 2016), I developed a spreading-activation algorithm to evaluate the relatedness between nodes in the graph. The measure in theory integrates multiple distributional features and is able to evaluate semantic relatedness between linguistic concepts in a graded manner. It is shown that the spreading-activation approach indeed effectively captures the semantic constraints embedded in linguistic distributional patterns.

In two groups of studies, I showed the formal computational capabilities of two specific types of distributional graphs, the LON and the CTN models. In the first group of studies, I showed that graphical encoding of linear co-occurrence (LON) may successfully bind direct syntagmatic relationships with paradigmatic relationships to form a representation of indirect syntagmatic relationships. Such representational capability enables the model to generalize on word-word lexical relationships. By systematically manipulating two relevant modeling parameters, i.e. the encoded information and the representational data structure, I showed that the graphical structure and the co-occurrence encoding collectively contributed to the high performance. With further formal investigations, I formulated a mathematical equivalence between the general spreading-activation processes on LON and multi-order cosine similarity in the corresponding vector space representation. The equivalence suggests that the graphical approach is advantageous in terms of a simultaneous access to similarity evaluations on vector spaces with various abstraction orders. In this way, integration of different orders of semantic relations, e.g. syntagmatic and paradigmatic relations, is allowed in the graphical LON, leading to the effective representation of indirect relations.

In the second group of studies, I showed that while the canonical LON is able to generalize on word-word lexical dependencies, it struggles to learn and generalize lexical dependencies involving three words. As a contrast, by explicitly encoding the constituent structure in the semantic network, the CTN model may successfully learn the multi-way lexical dependencies embedded in the linguistic input, as well as generalize, in a compositional manner, the learned dependencies to novel lexical combinations. With a series of comparison experiments to other distributional models, I showed that LON models encoding more distant co-occurrence and a Transformer-based model may succeed in the critical compositional generalization task, alongside with the CTN model. Basing on following-up analyses and further experiments on the models, I argue that a quasi semantic-syntactic structure might be necessary to guarantee effective representation of multi-way lexical dependencies. While the structure is granted in the CTN model, it might have emerged in the Transformer-based model due to its self-attention mechanism.

While the studies in this dissertation have primarily focused on testing formal computational capabilities of the distributional graphs with toy artificial corpora, the results of the modeling work may have profound implications in theory of knowledge representation and semantic development. First, it has demonstrated the unique value of encoding linguistic input in graphical topological structure and the effectiveness of the spreading activation in relatedness evaluation. In this way, the work has shown the potential of Distributional Graph as an approach to form effective semantic representation based on linguistic input. Second, as a distributional modeling approach, the performance of CTN against Transformer-based models suggests a possible two-stage semantic development in human. A great amount of behavioral experiments are needed to test the hypothesis and validate distributional graph as a model for human semantic memory, and these studies require prerequisite works to scale up Distributional Graph with naturalistic linguistic input. Finally, by integrating multi-modal distributional data and detailed grammatical information on the edges, it is possible to construct distributional graphs that allow generation of grammatical and meaningful novel concepts and expressions. Along the way of this challenging modeling march, it is likely that the modeling outcomes constantly inspire cognitive psychology studies that explore corresponding behaviors in human, as well as the underlying cognitive mechanism. The representational account in the generalized Distributional Graph may deepen our understanding of the developing nature of human knowledge, and in the long run contribute to creating more interpretable artificial intelligence and nurturing more knowledgeable humans.

References

Abernethy, A. P., Campbell, L. C., & Faigman, D. L. (2005). Human category learning. *Annual review of psychology*, *56*, 149-78.

Abu-Zhaya, R., Arnon, I., & Borovsky, A. (2022). Do children use multi-word information in real-time sentence comprehension? *Cognitive science*, *46 3*, e13111.

Anderson, J. R. (1983). A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*, *22*, 261-295.

Anderson, J. R., & Bower, G. H. (1974). A propositional theory of recognition memory. *Memory & Cognition*, *2*, 406-412.

Arias-Trejo, N., & Plunkett, K. (2009). Lexical–semantic priming effects during infancy. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *364*, 3633 - 3647.

Arias-Trejo, N., & Plunkett, K. (2013). What's in a link: Associative and taxonomic priming effects in the infant lexicon. *Cognition*, *128*, 214-227.

Asr, F. T., & Jones, M. N. (2017). An artificial language evaluation of distributional semantic models. In *Conference on computational natural language learning.*

Balota, D. A., & Lorch, R. (1986). Depth of automatic spreading activation: Mediated priming effects in pronunciation but not in lexical decision. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *12*, 336-345.

Baroni, M., Bernardi, R., & Zamparelli, R. (2014). Frege in space: A program of compositional distributional semantics. *Linguistic Issues in Language Technology*, *9*.

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 238–247).

Bejar, I., Chaffin, R., & Embretson, S. E. (1990). Cognitive and psychometric analysis of analogical problem solving..

Bicknell, K., Elman, J. L., Hare, M., McRae, K., & Kutas, M. (2010). Effects of event knowledge in processing verbal arguments. *Journal of memory and language*, *63 4*, 489-505.

Blei, D. M., Ng, A., & Jordan, M. I. (2001). Latent dirichlet allocation. *J. Mach. Learn. Res.*, *3*, 993-1022.

Bonner, M. F., & Epstein, R. A. (2020). Object representations in the human brain reflect the co-occurrence statistics of vision and language. *Nature Communications*, *12*. Retrieved from https://api.semanticscholar.org/CorpusID:214727124

Borovsky, A. (2022). Developmental changes in how children generalize from their experience to support predictive linguistic processing. *Journal of experimental child psychology*, *219*, 105349.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodei, D. (2020). Language models are few-shot learners. *ArXiv*, *abs/2005.14165*.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J. A., Horvitz, E., Kamar, E., ... Zhang, Y. (2023). Sparks of artificial general intelligence: Early experiments with gpt-4. *ArXiv*, *abs/2303.12712*.

Bullinaria, J. A., & Levy, J. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, *39*, 510-526.

Bullinaria, J. A., & Levy, J. (2012). Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and svd. *Behavior Research Methods*, *44*, 890-907.

Chen, D., Peterson, J. C., & Griffiths, T. L. (2017). Evaluating vector-space models of analogy. *ArXiv*, *abs/1705.04416*.

Chomsky, N. (1965). aspects of the theory of syntax / by noam chomsky..

Chwilla, D. J., & Kolk, H. H. J. (2002). Three-step priming in lexical decision. *Memory & Cognition*, *30*, 217-225.

Clair, M. C. S., Monaghan, P., & Ramscar, M. (2009). Relationships between language structure and language learning: The suffixing preference and grammatical categorization. *Cognitive science*, *33 7*, 1317-29.

Collins, A. M., & Loftus, E. F. (1975). A spreading-activation theory of semantic processing. *Psychological Review*, *82*, 407-428.

Collins, A. M., & Quillian, M. R. (1969). Retrieval time from semantic memory..

Cree, G. S., McRae, K., & McNorgan, C. (1999). An attractor model of lexical conceptual processing: Simulating semantic priming. *Cognitive Science*, *23*(3), 371–414.

De Deyne, S., Navarro, D. J., Perfors, A., & Storms, G. (2016). Structure at every scale: A semantic network account of the similarities between unrelated concepts. *Journal of Experimental Psychology: General*, *145*(9), 1228.

Deese, J. (1962). On the structure of associative meaning. *Psychological review*, *69*, 161-75.

de Saussure, F., Bally, C., Sechehaye, A., Reidlinger, A., & Baskin, W. (1960). Course in general linguistics. *Journal of American Folklore*, *73*, 274.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, *abs/1810.04805*.

Deyne, S. D., Navarro, D. J., Collell, G., & Perfors, A. (2021). Visual and affective multimodal models of word meaning in language and mind. *Cognitive Science*, *45*.

Deyne, S. D., Navarro, D. J., Perfors, A., Brysbaert, M., & Storms, G. (2019). The "small world of words" english word association norms for over 12,000 cue words. *Behavior Research Methods*, *51*, 987-1006.

Deyne, S. D., Navarro, D. J., Perfors, A., & Storms, G. (2016). Structure at every scale: A semantic network account of the similarities between unrelated concepts. *Journal of experimental psychology. General*, *145 9*, 1228-54.

Elman, J. L. (1990). Finding structure in time. *Cogn. Sci.*, *14*, 179-211.

Elman, J. L. (1991). Distributed representations, recurrent nets, and grammatical structure. *Mach. Learn.*, *7*, 195-225.

Elman, J. L. (1993). Learning and development in neural networks: the importance of starting small. *Cognition*, *48*, 71-99.

Elman, J. L. (2009). On the meaning of words and dinosaur bones: Lexical knowledge without a lexicon. *Cognitive science*, *33 4*, 547-582.

Erk, K., & Padó, S. (2008). A structured vector space model for word meaning in context. In *Proceedings of the 2008 conference on empirical methods in natural language processing* (pp. 897–906).

Erk, K., Padó, S., & Padó, U. (2010). A flexible, corpus-driven model of regular and inverse selectional preferences. *Computational Linguistics*, *36*, 723-763.

Evert, S., & Lapesa, G. (2021). FAST: A carefully sampled and cognitively motivated dataset for distributional semantic evaluation. In *Proceedings of the 25th conference on computational natural language learning* (pp. 588–595). Online: Association for Computational Linguistics.

Ferretti, T. R., McRae, K., & Hatherell, A. (2001). Integrating verbs, situation schemas, and thematic role concepts. *Journal of Memory and Language*, *44*, 516-547.

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

Garnsey, S. M., Pearlmutter, N. J., Myers, E. A., & Lotocky, M. A. (1997). The contributions of verb bias and plausibility to the comprehension of temporarily ambiguous sentences. *Journal of Memory and Language*, *37*, 58-93.

Gentner, D. (1975). Evidence for the psychological reality of semantic components: The verbs of possession..

Gershman, S. J., & Tenenbaum, J. B. (2015). Phrase similarity in humans and machines. *Cognitive Science*.

Goldberg, A. E. (1995). Constructions: A construction grammar approach to argument structure..

Greene, M. R. (2013). Statistics of high-level scene context. *Frontiers in Psychology*, *4*.

Griffiths, T. L., Steyvers, M., & Tenenbaum, J. B. (2007). Topics in semantic representation. *Psychological review*, *114 2*, 211-44.

Hare, M., McRae, K., & Elman, J. L. (2003). Sense and structure: Meaning as a determinant of verb subcategorization preferences. *Journal of Memory and Language*, *48*, 281-303.

Hills, T. T., Maouene, J., Riordan, B., & Smith, L. B. (2010). The associative structure of language: Contextual diversity in early word learning. *Journal of memory and language*, *63*(3), 259–273.

Hills, T. T., Maouene, M., Maouene, J., Sheya, A., & Smith, L. B. (2009). Categorical structure among shared features in networks of early-learned nouns. *Cognition*, *112*, 381-396. Retrieved from https://api.semanticscholar.org/CorpusID:14148634

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735-1780.

Huebner, P. A., & Willits, J. A. (2018). Structured semantic knowledge can emerge automatically from predicting word sequences in child-directed speech. *Frontiers in Psychology*, *9*.

Huebner, P. A., & Willits, J. A. (2021a). Scaffolded input promotes atomic organization in the recurrent neural network language model. In *Conference on computational natural language learning.*

Huebner, P. A., & Willits, J. A. (2021b). Using lexical context to discover the noun category: Younger children have it easier. *Psychology of Learning and Motivation*.

Jackendoff, R. (2002). Foundations of language: Brain, meaning, grammar, evolution..

Jones, M. N., & Mewhort, D. J. K. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological review*, *114 1*, 1-37.

Kamide, Y., Altmann, G. T. M., & Haywood, S. L. (2003). The time-course of prediction in incremental sentence processing: Evidence from anticipatory eye-movements. *Journal of Memory and Language*, *49*, 133-156.

Kenett, Y. N., Anaki, D., & Faust, M. (2014). Investigating the structure of semantic networks in low and high creative persons. *Frontiers in Human Neuroscience*, *8*.

Kenett, Y. N., Beaty, R. E., Silvia, P. J., Anaki, D., & Faust, M. (2016). Structure and flexibility: Investigating the relation between the structure of the mental lexicon, fluid intelligence, and creative achievement. *Psychology of Aesthetics, Creativity, and the Arts*, *10*, 377-388.

Kenett, Y. N., Kenett, D. Y., Ben-Jacob, E., & Faust, M. (2011). Global and local features of semantic networks: Evidence from the hebrew mental lexicon. *PLoS ONE*, *6*.

Kenett, Y. N., Levi, E., Anaki, D., & Faust, M. (2017). The semantic distance task: Quantifying semantic distance with semantic network path length. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *43*, 1470–1489.

Kueser, J. B., Horvath, S., & Borovsky, A. (2023). Two pathways in vocabulary development: Large-scale differences in noun and verb semantic structure. *Cognitive Psychology*, *143*.

Kueser, J. B., Peters, R., & Borovsky, A. (2022). The role of semantic similarity in verb learning events: Vocabulary-related changes across early development. *Journal of experimental child psychology*, *226*, 105565.

Kumar, A. A., Balota, D. A., & Steyvers, M. (2019). Distant connectivity and multiple-step priming in large-scale semantic networks. *Journal of experimental psychology. Learning, memory, and cognition*.

Kumar, A. A., Steyvers, M., & Balota, D. A. (2021). Semantic memory search and retrieval in a novel cooperative word game: A comparison of associative and distributional semantic models. *Cognitive science*, *45 10*, e13053.

Kumar, A. A., Steyvers, M., & Balota, D. A. (2022). A critical review of network-based and distributional approaches to semantic memory structure and processes. *Topics in cognitive science*.

Kutas, M., & Federmeier, K. D. (2011). Thirty years and counting: finding meaning in the n400 component of the event-related brain potential (erp). *Annual review of psychology*, *62*, 621-47.

Lake, B., & Baroni, M. (2018). Generalization without systematicity: On the compositional skills of seq2seq recurrent networks. In *Int. conf. on mach. learn.* (pp. 2873–2882).

Lake, B. M., & Murphy, G. L. (2020). Word meaning in minds and machines. *Psychological review*. Retrieved from https://api.semanticscholar.org/CorpusID:221043556

Landauer, T. K., & Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, *104*(2), 211.

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*, 259-284.

Langacker, R. W. (2008). Cognitive grammar: A basic introduction..

Lany, J., & Saffran, J. R. (2011). Interactions between statistical and semantic information in infant language development. *Developmental science*, *14 5*, 1207-19.

Liu, F., Vulić, I., Korhonen, A., & Collier, N. (2021, November). Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 1442–1459). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Lund, K., & Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, *28*, 203-208.

Lupyan, G., & Lewis, M. (2019). From words-as-mappings to words-as-cues: the role of language in semantic knowledge. *Language, Cognition and Neuroscience*, *34*(10), 1319–1337.

Lupyan, G., Rakison, D. H., & McClelland, J. L. (2007). Language is not just for talking. *Psychological Science*, *18*, 1077 - 1083. Retrieved from https://api.semanticscholar.org/CorpusID:13455410

MacDonald, M. C., Pearlmutter, N. J., & Seidenberg, M. S. (1994). The lexical nature of syntactic ambiguity resolution. *Psychological Review*, *101*, 676-703.

Malt, B. C., Gennari, S. P., & Imai, M. (2010). Lexicalization patterns and the world-to-words mapping..

Retrieved from https://api.semanticscholar.org/CorpusID:53064973

Mandera, P., Keuleers, E., & Brysbaert, M. (2017). Explaining human performance in psycholinguistic tasks with models of semantic similarity based on prediction and counting : A review and empirical validation. *Journal of Memory and Language*, *92*, 57-78.

Mao, S., & Willits, J. (2020). *Graphical vs. spatial models of distributional semantics.* PsyArXiv.

Marcus, G. F. (2020). The next decade in ai: Four steps towards robust artificial intelligence. *ArXiv*, *abs/2002.06177*.

Marr, D. (1982). Vision: A computational investigation into the human representation..

Matsuki, K., Chow, T. H. F., Hare, M., Elman, J. L., Scheepers, C., & McRae, K. (2011). Event-based plausibility immediately influences on-line language comprehension. *Journal of experimental psychology. Learning, memory, and cognition*, *37 4*, 913-34.

McClelland, J. L., John, M. F. S., & Taraban, R. (1989). Sentence comprehension: A parallel distributed processing approach. *Language and Cognitive Processes*, *4*.

McNamara, T. P., & Altarriba, J. (1988). Depth of spreading activation revisited: Semantic mediated priming occurs in lexical decisions. *Journal of Memory and Language*, *27*, 545-559.

McRae, K., de Sa, V. R., & Seidenberg, M. S. (1997). On the nature and scope of featural representations of word meaning. *Journal of experimental psychology. General*, *126 2*, 99-130.

McRae, K., Hare, M., Elman, J. L., & Ferretti, T. R. (2005). A basis for generating expectancies for verbs from nouns. *Memory & Cognition*, *33*, 1174-1184.

McRae, K., Spivey-Knowlton, M. J., & Tanenhaus, M. K. (1998). Modeling the influence of thematic fit (and other constraints) in on-line sentence comprehension. *Journal of Memory and Language*, *38*(3), 283–312.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. In *1st international conference on learning representations.*

Miller, G. A. (1992). Wordnet: A lexical database for english. *Commun. ACM*, *38*, 39-41.

Mitchell, J., & Lapata, M. (2010). Composition in distributional models. *Cognitive science*, *34*(8), 1388–1429.

Montague, R. (1970). Universal grammar. *Theoria*, *36*(3), 373–398. doi: 10.1111/j.1755-2567.1970.tb00434.x

Nelson, D. L., McEvoy, C., & Schreiber, T. A. (2004). The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, *36*, 402-407.

Osgood, C. E. (1952). The nature and measurement of meaning. *Psychological bulletin*, *49 3*, 197-237.

Padó, S., & Lapata, M. (2007, 06). Dependency-Based Construction of Semantic Space Models. *Computational Linguistics*, *33*(2), 161-199.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Conference on empirical methods in natural language processing.* Retrieved from https://api.semanticscholar.org/CorpusID:1957433

Perruchet, P., & Vinter, A. (1998). Parser: A model for word segmentation. *Journal of Memory and Language*, *39*, 246-263.

Perszyk, D. R., & Waxman, S. R. (2018). Linking language and cognition in infancy. *Annual Review of Psychology*, *69*, 231–250.

Peters, R., Kueser, J. B., & Borovsky, A. (2021). Perceptual connectivity influences toddlers' attention to known objects and subsequent label processing. *Brain Sciences*, *11*.

Peterson, J. C., Chen, D., & Griffiths, T. L. (2020). Parallelograms revisited: Exploring the limitations of vector space models for simple analogies. *Cognition*, *205*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Ravfogel, S., Goldberg, Y., & Linzen, T. (2019). Studying the inductive biases of rnns with synthetic variations of natural languages. In *North american chapter of the association for computational linguistics.* Retrieved from https://api.semanticscholar.org/CorpusID:80628431

Rayner, K., Warren, T., Juhasz, B. J., & Liversedge, S. P. (2004, November). The effect of plausibility on eye movements. *J. Exp. Psychol. Learn. Mem. Cogn.*, *30*(6), 1290–1301.

Resnik, P. (1996). Selectional constraints: an information-theoretic model and its computational realization. *Cognition*, *61*, 127-159.

Ri, R., & Tsuruoka, Y. (2022). Pretraining with artificial language: Studying transferable knowledge in language models. In *Annual meeting of the association for computational linguistics.* Retrieved from https://api.semanticscholar.org/CorpusID:247597105

Rogers, T. T., & McClelland, J. L. (2004). Semantic cognition: A parallel distributed processing approach..

Rotaru, A. S., Vigliocco, G., & Frank, S. (2018). Modeling the structure and dynamics of semantic processing. *Cognitive Science*, *42*, 2890 - 2917.

Rubin, T. N., Kievit-Kylar, B., Willits, J. A., & Jones, M. N. (2014). Organizing the space and behavior of semantic models. *CogSci ... Annual Conference of the Cognitive Science Society. Cognitive Science Society (U.S.). Conference*, *2014*, 1329-1334.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533-536.

Rumelhart, D. E., & Todd, P. M. (1993). Learning and connectionist representations..

Sadeghi, Z., McClelland, J. L., & Hoffman, P. (2015). You shall know an object by the company it keeps: An investigation of semantic representations derived from object co-occurrence in visual scenes. *Neuropsychologia*, *76*, 52 - 61. Retrieved from https://api.semanticscholar.org/CorpusID:2392276

Sahlgren, M. (2006). The word-space model : Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces..

Scott, R. M., & Fisher, C. (2009). Two-year-olds use distributional cues to interpret transitivity-alternating verbs. *Language and Cognitive Processes*, *24*, 777 - 803. Retrieved from https://api.semanticscholar.org/CorpusID:23014002

Seidl, A., Indarjit, M., & Borovsky, A. (2023). Touch to learn: Multisensory input supports word learning and processing. *Developmental science*, e13419.

Smith, E. E., Shoben, E. J. J., & Rips, L. J. (1974). Structure and process in semantic memory: A featural model for semantic decisions. *Psychological Review*, *81*, 214-241.

Steyvers, M., & Tenenbaum, J. B. (2005). The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cognitive science*, *29 1*, 41-78.

Tabullo, Á. J., Arismendi, M., Wainselboim, A., Primero, G., Vernis, S., Segura, E. T., . . . Yorio, A. (2012). On the learnability of frequent and infrequent word orders: An artificial language learning study. *Quarterly Journal of Experimental Psychology*, *65*, 1848 - 1863.

Trueswell, J. C., Tanenhaus, M. K., & Garnsey, S. M. (1994). Semantic influences on parsing: Use of thematic role information in syntactic ambiguity resolution. *Journal of Memory and Language*, *33*, 285-318.

Trueswell, J. C., Tanenhaus, M. K., & Kello, C. T. (1993). Verb-specific constraints in sentence processing: separating effects of lexical preference from garden-paths. *Journal of experimental psychology. Learning, memory, and cognition*, *19 3*, 528-53.

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*, 327-352.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vulić, I., Ponti, E. M., Korhonen, A., & Glavaš, G. (2021, August). LexFit: Lexical fine-tuning of pretrained language models. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)* (pp. 5269–5283). Online: Association for Computational Linguistics.

Wang, D., & Eisner, J. (2016). The galactic dependencies treebanks: Getting more data by synthesizing new languages. *Transactions of the Association for Computational Linguistics*, *4*, 491-505.

White, J. C., & Cotterell, R. (2021). Examining the inductive bias of neural language models with artificial languages. In *Annual meeting of the association for computational linguistics.*

Willits, J. A., Amato, M. S., & MacDonald, M. C. (2015). Language knowledge and event knowledge in language use. *Cognitive Psychology*, *78*, 1-27.

Willits, J. A., Seidenberg, M. S., & Saffran, J. R. (2014). Distributional structure in language: Contributions to noun–verb difficulty differences in infant word recognition. *Cognition*, *132*, 429-436.

Zettersten, M., Wojcik, E. H., Benitez, V. L., & Saffran, J. R. (2018). The company objects keep: Linking referents together during cross-situational word learning. *Journal of memory and language*, *99*, 62-73.