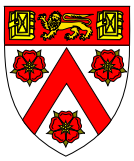
UNIVERSITY OF
CAMBRIDGE

Functional Distributional Semantics

Learning Linguistically Informed Representations from a Precisely Annotated Corpus

Guy Edward Toh Emerson



Trinity College

This dissertation is submitted on 20 August 2018 for the degree of Doctor of Philosophy

Abstract

Functional Distributional Semantics: Learning Linguistically Informed Representations from a Precisely Annotated Corpus

The aim of distributional semantics is to design computational techniques that can automatically learn the meanings of words from a body of text. The twin challenges are: how do we represent meaning, and how do we learn these representations? The current state of the art is to represent meanings as vectors – but vectors do not correspond to any traditional notion of meaning. In particular, there is no way to talk about *truth*, a crucial concept in logic and formal semantics.

In this thesis, I develop a framework for distributional semantics which answers this challenge. The meaning of a word is not represented as a vector, but as a *function*, mapping entities (objects in the world) to probabilities of truth (the probability that the word is true of the entity). Such a function can be interpreted both in the machine learning sense of a classifier, and in the formal semantic sense of a truth-conditional function. This simultaneously allows both the use of machine learning techniques to exploit large datasets, and also the use of formal semantic techniques to manipulate the learnt representations. I define a probabilistic graphical model, which incorporates a probabilistic generalisation of model theory (allowing a strong connection with formal semantics), and which generates semantic dependency graphs (allowing it to be trained on a corpus). This graphical model provides a natural way to model logical inference, semantic composition, and context-dependent meanings, where Bayesian inference plays a crucial role. I demonstrate the feasibility of this approach by training a model on WikiWoods, a parsed version of the English Wikipedia, and evaluating it on three tasks. The results indicate that the model can learn information not captured by vector space models.

Guy Edward Toh Emerson

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration. It is not substantially the same as any that I have submitted, or am concurrently submitting, for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution. I further state that no substantial part of my dissertation has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution. This dissertation does not exceed the prescribed limit of 60 000 words.

Guy Edward Toh Emerson
20 August 2018

Acknowledgements

Towards the end of my PhD studies, many people asked me if I was glad to be done. The truth is, regardless of the effort of writing up this thesis, I have really enjoyed my PhD!

Above all, I have to thank my PhD supervisor, Ann Copestake. You have been supportive throughout my studies (indeed, beginning with my master's studies), urging me to try the simple things first, helping me to keep sight of the bigger picture, and lending me many books from your bookshelf. Without your guidance, this thesis would certainly be less coherent.

The Computer Lab has been a welcoming environment – in particular, I should thank Lise Gough, for making sure things run smoothly, the Schiff Foundation, for funding my PhD, and of course all the people in the NLIP research group. While it is tempting to list everyone here, I will spare the reader from that,[1] and single out a few people: fellow supervisees Matic, Ewa, Alex, and Paula, postdocs Laura, Tamara, and Marek, and fellow PhD students Kris and Amandla. I have had many stimulating discussions with each of you, which have really enriched my time here. In particular, I should thank Kris for the many times we've gone to the Food Park (and above all the Wandering Yak), and for the many times you've run into my office eager to tell me about some new problem.

Beyond my department, there are also many people across the university who have contributed to my time here. I should thank everyone involved in the burgeoning Cambridge Language Sciences initiative,[2] for trying to create bridges between departments, and in particular Jane Walsh, for keeping the initiative running.

Beyond academia, there are also many people in Cambridge who have made the past four years so enjoyable. I should thank the Cambridge University Dancesport Team[3] for being such a wonderful distraction. In particular, I should thank my life partner and dance partner Mary, for your love and support – and for all the time you've spent listening to me ramble, complain, and get excited about my work.

Finally, I should acknowledge my own privilege in being in a position to happily pursue a PhD. I hope the world continues to develop so that knowledge is open, and so that anyone who wants to pursue a PhD feels able to do so. For my own place in the world, I want to thank my parents. You have supported me, encouraged me, and pushed me to be the best I can be.

[1] https://www.cl.cam.ac.uk/research/nl/people/
[2] https://www.languagesciences.cam.ac.uk/
[3] http://cudt.org/

Now that my viva is over (for which I should thank my examiners Paula Buttery and Katrin Erk, for the challenging questions and compelling suggestions), and now that this thesis is submitted, I look forward to the next steps beyond my PhD.

Contents

Chapter 1

Between Linguistics and Machine Learning

This thesis is about meaning – how to represent it, and how to learn it. The ultimate aim is to develop a framework which is both compatible with formal linguistic theory, and empirically testable using real-world data. My motivations are twofold: to shed light on what it means to know a language, and to push forward the limits of machine learning.

From the linguistic point of view, if we are to take lexical semantics seriously – that is, to have a theory that can model the meanings of words, including all their subtle connotations – then we cannot hope to write down all the details by hand. Traditional techniques are time-consuming, and variations of meaning difficult to pin down. Data-driven techniques are necessary to move from an abstract semantic theory to a fleshed-out model of a real language. Furthermore, an explicit computational model does more than just allow us to test the theory – as we will see, using probabilistic techniques can provide new insights on old problems.

From the machine learning point of view, a learning algorithm requires an objective. In the case of language, what should the objective be? In this thesis, I will not focus on one specific task – after all, people use language as a general purpose tool to communicate and to store knowledge, even in completely new domains. The objective, then, is to learn semantic representations that are generally useful, rather than tied to a specific task. However, without an end task in mind, the next question is: what semantic representations are we aiming to learn? As a guide, we can look to linguistics, and aim to learn structures that have proven useful in formal models of language.

In short, linguistics clarifies the goal, and machine learning provides the tools.

1.1 Synopsis

1.1.1 Core of the Thesis

In this thesis, I focus on *distributional semantics*, which has the goal of learning the meanings of words from a corpus. The basic idea is that the contexts in which a word appears give us information about its meaning.

How can formal semantics guide us in building a distributional semantic model? For this thesis, the most important guiding idea is that the meaning of a word should be represented by a *truth-conditional function* – a mapping from entities (objects in the world) to truth values (either true or false). From a Bayesian point of view, this suggests representing the meaning of a word as a function from entities to *probabilities* of truth. For example, the function for the word *cup* would return high probabilities for typical cups, middling values for entities near the boundary of the concept (such as mugs, glasses, and bowls), and low values for other objects. I call such a function a *semantic function*. Although such a function might at first seem esoteric to a machine learning audience, it will seem quite familiar if viewed as a *binary classifier*.

How can we learn such functions? The second guiding idea is to use *semantic dependency graphs* as representations for the meanings of sentences. Compared to other sentence representations used in formal semantics, they are more convenient for machine learning models – in particular, for *probabilistic graphical models*, which also use graph structures.

Semantic functions and semantic dependency graphs thus provide a link between formal semantics and machine learning. The model architecture is informed by semantic theory, but the model parameters are fully trained.

This thesis should be of interest to linguists: firstly, I give a probabilistic generalisation of model theory, and I show how this provides a novel mechanism for modelling *context dependence*; secondly, I show how a probabilistic model structure can be learnt using corpus data, which allows semantic theories to be tested on a larger scale.

This thesis should also be of interest to machine learning researchers: firstly, I empirically demonstrate that for distributional semantics, a functional model can perform better than a vector space model, particularly on difficult datasets; secondly, I explain how a functional model is *logically interpretable*, an important advantage compared to vector space models.

The core ideas in this thesis have been published in a series of papers (Emerson and Copestake, 2016, 2017a,b).

1.1.2 Outline of the Thesis

The remainder of this chapter introduces distributional semantics and model-theoretic semantics, two prominent approaches to semantics which I build on in this thesis. In recent years, distributional semantics has proven much more popular in the fields of computational linguistics

and natural language processing (NLP). While there have been repeated calls to arms to bring more linguistics into computational linguistics (for example: Spärck-Jones, 2007b; Church, 2011; Kay, 2014; Smith, 2017), and while there is a rich and active literature on integrating distributional and model-theoretic approaches (for an overview, see: Boleda and Herbelot, 2017), there remain many open problems. This chapter sets the scene for the rest of the thesis.

In Chapter 2, I discuss the goals of semantics, outlining a number of challenges that any theory of semantics should aim to deal with. To evaluate how well current theories address these challenges, I survey existing work in distributional semantics and model-theoretic semantics. This chapter motivates the framework developed in the rest of the thesis.

In Chapter 3, I introduce the framework. I begin by looking at model structures in model-theoretic semantics, and I propose a probabilistic generalisation of a model structure in §3.3. This generalisation lays the groundwork for the two core ideas discussed in §1.1.1 above. First, I define semantic functions in §3.4, as part of a probabilistic model structure. Second, I define a probabilistic graphical model for model structures in §3.5, which is structured using semantic dependency graphs and semantic functions. I apply this graphical model to distributional semantics in §3.6, which gives the framework of Functional Distributional Semantics. I conclude in §3.7 by looking at this framework in light of the goals given in Chapter 2.

In Chapter 4, I explain how the logical structure of the framework is useful. I first show, in §4.1, how Bayesian inference can provide an account of context-dependent meanings, maintaining the intuition behind linguistic accounts, but using a precisely defined mathematical mechanism. I then show, in §4.5, how Bayesian inference can be used to perform logical inference, and I prove an equivalence with traditional syllogistic logic.

In Chapter 5, I give an implementation of the framework, using a combination of Restricted Boltzmann Machines and feedforward neural networks. The model architecture is described in §5.1, while the rest of the chapter is dedicated to how to train the model. The main challenge is the large number of latent variables: when training a model on text alone, we do not observe the entities themselves, but only their textual descriptions. This means that, for every observed content word, there is an unobserved entity that the word describes. To make training tractable, I have adapted two approximate inference techniques: a Markov Chain Monte Carlo method is described in §5.3; and a Variational Inference method is described in §5.4. These approximate inference techniques are crucial in making tractable the logical inference calculations proposed in Chapter 4.

In Chapter 6, I test my framework, training a model on WikiWoods (a parsed version of the English Wikipedia), and giving experimental results on three tasks: measuring lexical similarity, measuring similarity in context, and finally an inference task involving composition of relative clauses. The results demonstrate that my framework can improve performance compared to a vector space model.

In Chapter 7, I extend the above approach to deal with multiple quantifiers, allowing us to

handle arbitrary propositions. This is an important strength compared to current distributional semantic models. Furthermore, while traditional logics deal well with quantifiers like *every* and *some*, that have clear truth conditions, they struggle to model vague quantifiers like *many*, as well as so-called *generic* sentences. In §7.4, I discuss how a probabilistic approach can provide a more natural account of vague quantifiers and generics.

Finally, in Chapter 8, I reflect on the achievements of the thesis, and give an outlook on future work – because the framework developed in this thesis is interpretable in linguistic terms and in logical terms, there is a clear and plausible path from this work to more general models of language.

1.2 Distributional Semantics

The aim of distributional semantics is to learn the meanings of linguistic expressions from a large corpus of text. The core idea, known as the **distributional hypothesis**, is that the contexts in which an expression appears give us information about its meaning. The hypothesis is often stated more narrowly, to say that similar words will appear in similar contexts – but in this thesis I will be interested in semantics beyond similarity. Fig. 1.1 illustrates the kind of information we might hope to learn.[1]

Why should we want to study distributional semantics? For a machine learning researcher, there is a very short answer: training a model requires data, and textual data is cheap, so we should try to use it.

For a linguist, the distributional hypothesis provides a methodology for studying language. The idea has roots in American structuralism (Harris, 1954) and British lexicology (Firth, 1951, 1957)[2], but it was not until the advent of modern computing and the availability of large machine-readable language resources that it began to be used in practice. In a notable early work, Spärck-Jones (1964) represented the meaning of a word as a boolean vector (based on the entries in a thesaurus), with similarity defined in terms of vector overlap.

As McNally (2017) and Lenci (2008, 2018) have argued, distributional representations can be used as surrogates for conceptual representations – but crucially, they can be calculated concretely. Used in this way, distributional data allows us to develop and test linguistic theories. Of course, distributional data cannot be enough to learn a full model of meaning, because it does not include grounded, non-linguistic data, such as sensory perception and motor control. However, it is *necessary* for a full model of the language of literate speakers, as a large proportion of L1 vocabulary learning comes from reading new words in normal text (Nagy et al., 1987; Miller and Charles, 1991). A distributional semantic model is not a complete model of meaning, but it is a good place to start.

[1] Examples were obtained under the terms of the British National Corpus End User Licence. For further information, see: http://www.natcorp.ox.ac.uk

[2] Firth used the term "collocational" rather than "distributional".

... of anticipating being hurt by another	horse	especially if some other rider comes ...
... was simply beaten by a better	horse	at the distance on the day ...
... can infer from these studies that	horses	reared with other horses in a ...
... studies that horses reared with other	horses	in a free and enriched environment ...
... people saying "Is that all your	horse	gets to eat?" in amazement. The ...
... and a cache of cattle and	horse	bones, while from Normangate Field a ...
... Bachelor's Button was a sterling good	horse,	especially at Ascot, but he was ...
... the same way as a domestic	horse	that it may be stabled with ...
... Attachment in 1790 – that is, one	horse	or two cows for each £4 ...
... grey hair as coarse as a	horse	's tail straying from her mob-cap ...

Figure 1.1: Ten instances of *horse* in the British National Corpus, with a window of six words either side. From these, we might learn that horses are animals used in racing and agriculture. Learning such information automatically is the goal of distributional semantics.

1.2.1 Vector Space Models

The most popular approaches to distributional semantics represent the meaning of a word as a **vector** – in other words, as an array of numerical values. The idea that meaning varies continuously along a number of dimensions has roots in certain schools of psychology (Osgood, 1952), but it is with the rise of distributional semantics, discussed in §1.2 above, that vector space models have become widespread (for an overview, see: Erk, 2012; Clark, 2015).

One method to build distributional vectors is a **count** approach, where we count the number of times words appear in different contexts. A simple type of context uses a **window** of words – for each instance of a target word, we observe N words before and N words after the target, as illustrated in Fig. 1.1 for $N = 6$. Each word observed in the window defines a context. In the above example, important context words would include: *rider*, *reared*, *cattle*, *Ascot*, *stabled*, *tail*. Contexts can also be defined in other ways, for example on the basis of syntactic structure. The choice of context is important, allowing vectors to capture either paradigmatic or syntagmatic relations, as discussed by Sahlgren (2006). Once the counts have been calculated across the whole corpus, we can process these counts in some way, such as calculating the pointwise mutual information (PMI), as proposed by Turney (2001), building on earlier work that used PMI to measure word association (Church and Hanks, 1990). There are many alternative ways to process the counts, and overviews of techniques are given by Turney and Pantel (2010) and Lapesa and Evert (2014).

An alternative is an **embedding** approach, where we define vectors as part of a machine learning model and optimise these vectors to perform some task. The vectors are typically part of a neural network, such as Mikolov et al. (2013)'s Skip-gram model, where the task is to predict the words in each window. Alternatively, they can be trained using probabilistic generative models, such as proposed by Ó Séaghdha and Korhonen (2014).

There are strong links between count and embedding approaches. For example, Levy and Goldberg (2014) prove that Mikolov et al.'s embedding model produces an approximate factorisation of a count-based PMI matrix. Leveraging this relationship, Levy et al. (2015a) optimise count models using techniques developed for embedding models. In a similar vein, Cotterell et al. (2017) prove that Mikolov et al.'s model performs exponential-family principal component analysis (EPCA) on a count matrix.

Vector space models have also been influential in other areas of NLP, such as to represent queries and documents in Information Retrieval (Salton, 1979). (For a history of this idea, see: Dubin, 2004; for a general history of IR, which situates the vector space model in a larger context, see: Spärck-Jones, 2007a.)

While vector space models have proven useful in a variety of tasks, vectors do not naturally impose the structure necessary to model various aspects of meaning, as I will argue in Chapter 2.

1.2.2 A Note on Terminology: Numerical Vectors and Algebraic Vectors

The term **vector** is ambiguous. In computer science, it refers to a linear[3] **array**. This would be referred to in mathematics as a **tuple**. What are often referred to in NLP as feature vectors or embedding vectors could be equally described as arrays.

In the mathematical sense, a **vector space** (over the real numbers) is a set with a specific kind of algebraic structure: vectors can be added to each other, and multiplied by real numbers; vector addition forms an abelian group; multiplication by real numbers is distributive over vector addition; and these operations are compatible with the usual addition and multiplication of real numbers. The study of vector spaces is known as **linear algebra**.

The algebraic vector space axioms would be violated by many feature vectors in NLP. For example, for any vector v, the vector space will also include $-v$, but many feature vectors and embeddings are assumed to be non-negative. However, this should not necessarily be seen as a problem – many useful numerical operations, such as elementwise multiplication of arrays, are not natural algebraic operations on a vector space. Indeed, Dubin (2004)'s account of the history of the vector space model in information retrieval can also be read as a cautionary tale, warning against the assumption that numerical arrays must be algebraic vectors.

A common motivation for using arrays is that we can define a notion of **distance** (or its inverse, **similarity**). Mathematically, such a space would be called a **metric space**.[4] In the absence of an established term for "element of a metric space", I will use the term "vector" in the sense of a numerical array where the notion of distance is important.[5] This is consistent

[3] Also known as "one-dimensional", where there is a second clash in terminology, since the mathematical "dimension" would be the number of entries, referred to in computer science as the "length" of the array.

[4] Cosine similarity does not induce a metric, but rather a "pseudometric", since it is insensitive to the magnitude of a vector. However, it does induce a true metric on the set of unit vectors.

[5] An alternative term might be "point", although that is used more generally for topological spaces. It is not common in the NLP literature, perhaps because terms like "word point" would not be particularly evocative.

with existing work, even where it is not explicitly acknowledged. For example, Support Vector Machines (SVMs) don't require algebraic structure for the input vectors, but rather a notion of distance, induced by the kernel function[6] (for the modern formulation of SVMs, see: Cortes and Vapnik, 1995).

The most popular distributional semantic models are vector space models. This ubiquity has led some authors to use the term **distributional** to refer to vector space models in general, rather than to models trained on corpus data. I will only use the term to refer to corpus-based models, which is the sense introduced by Harris. This can be contrasted with the term **distributed**, which typically refers to vector space models trained as part of a neural network, where the meaning is intuitively "distributed" across all dimensions. The dimensions in such a model can only be interpreted in the context of the network (unlike count-based vector models, where the dimensions correspond to contexts). Finally, the term **embedding** is often used to refer to distributed vector representations – each object of interest is mapped to ("embedded in") the vector space.

One approach to semantics which does exploit the algebraic notion of a vector space is the type-driven tensorial framework proposed by Coecke et al. (2010) and Baroni et al. (2014), where linear algebra plays a crucial role. This framework will be discussed in §2.5.3.

1.3 Model-Theoretic Semantics

A standard approach to formal semantics is model theory (for expositions, see: Cann, 1993; Allan, 2001; Kamp and Reyle, 2013). The popularity of model theory is due to its precisely defined notion of truth, and to its compatibility with first order logic and λ-calculus. This allows us to derive the meanings of complex expressions by composing the meanings of their parts, and allows us to evaluate the truth or falsehood of sentences. Model theory was formally defined by Tarski and Vaught (1956),[7] further developed by Montague (1973), and popularised by Partee (1975) and Dowty et al. (1981).

The basic idea is that linguistic expressions acquire meaning via interpretation in a **model structure**.[8] Each model structure includes a set of **individuals** (or **entities**) – intuitively, these represent objects or people in the world. The meaning of a content word is called a **predicate**, formalised as a function mapping from individuals to **truth values** – either **truth** or **falsehood** (also referred to as **falsity**). More precisely, an n-place predicate maps each n-tuple of indi-

[6] More precisely, a positive definite kernel is equivalent to an inner product in a feature space – this is known as the Moore-Aronszajn Theorem (Aronszajn, 1950). Although kernel methods require the feature space to be a vector space (in fact, a "reproducing kernel Hilbert space"), there is no such requirement on the input space. Furthermore, the mapping from the input space to the feature space does not need to cover the feature space, and so the algebraic structure of feature space does not induce algebraic structure on the input space. However, distances in the feature space induce distances in the input space (technically, a "pseudometric", since input vectors may be mapped to the same feature vector) – given a kernel K, we can define distances as $d(x, y) = \sqrt{K(x, x) - 2K(x, y) + K(y, y)}$.

[7] For an overview, see: Hodges, 2014.

[8] I use the term "model structure" rather than just "model" or "structure", to avoid the ambiguity in both terms.

viduals to a truth value. For example, the meaning of *dog* can be represented as a one-place predicate, that maps an individual to truth if the individual is a dog, and to falsehood otherwise. The meaning of *chase* can be represented as a two-place predicate, that maps a pair of individuals (x, y) to truth if x chases y, and to falsehood otherwise.

Alternatively, an n-place predicate can be formalised as an **extension** (or **denotation**) – the set of n-tuples of individuals for which the predicate is true. Note that for both of the above formalisations (truth-conditional functions and extensions), the interpretation of a predicate is dependent on a model structure. The difference between the two will be explored in §3.2.

At its simplest, a model-theoretic representation for a sentence includes a set of **variables** and a set of predicates taking variables as arguments. For example, we can represent the sentence *a dog chased a cat*, with two variables and three predicates: $\mathit{dog}(x)$, $\mathit{cat}(y)$, $\mathit{chase}(x, y)$.[9] We also need some way to combine the truth values for each of these predicates into a single truth value for the whole sentence. In the above case, we can simply use the logical and operator, $\wedge$. Finally, we need some way to match the variables to the individuals in a model structure, which is called **quantification**. The two simplest quantifiers are the **universal quantifier** $\forall$ (we need to match the variable against every individual) and the **existential quantifier** $\exists$ (we need to find a single individual for the variable). In the above example, both x and y can be existentially quantified. We can then represent the meaning of a sentence using a logical **proposition**, where all variables are quantified – given a model structure, a proposition can be evaluated to give a truth value. The meaning of the above example could be represented by the proposition $\exists x \exists y\ \mathit{dog}(x) \wedge \mathit{cat}(y) \wedge \mathit{chase}(x, y)$. This is true when the model structure includes two individuals, where one is a dog, the other is a cat, and the first chases the second.

Many extensions of model theory have been proposed, and a full review would be beyond the scope of this thesis. I will focus on the **extensional fragment** of language, in other words sentences directly asserting facts. I will not cover imperatives, questions, modals, performatives, attitude reports and so on, but I should note that much of the model-theoretic literature is compatible with my approach. Three extensions of model theory will be of particular importance in this thesis, and will be discussed in the following subsections.

1.3.1 Neo-Davidsonian Event Semantics

In the above, I described how the meaning of *chase* can be represented as a two-place predicate, which holds between pairs of individuals. However, it is often useful to be able to directly refer to **events** – in this case, the event of chasing.[10] For example, to model the semantics of *a dog chased a cat yesterday*, it's not enough to say that the dog and cat existed yesterday – we need

[9] For ease of exposition, I am assuming here that each word corresponds to a single predicate, but in general there may be lexical ambiguity, which will be discussed in §2.2.2.

[10] Following Bach (1986), some authors distinguish "events", "processes" and "states", as kinds of "eventualities". Maienborn (2005) further distinguishes "Davidsonian eventualities" and "Kimian states". I will not make such distinctions in this thesis, and use the term "event" in the wider sense.

to say that the chasing happened yesterday.

A neo-Davidsonian approach to event semantics (Davidson, 1967; Parsons, 1990) deals with this by treating events as individuals. Verbal predicates are one-place relations, which can be true of event individuals. Adverbials like *yesterday* can then take the event individual as an argument. The **participants** of an event are indicated by two-place relations, linking the event to the participant. For example, the above sentence could be represented with three individuals and five relations: $dog(x)$, $chase(y)$, $cat(z)$, $\text{ARG1}(y, x)$, $\text{ARG2}(y, z)$, $yesterday(y)$. Here, the ARG1 and ARG2 relations indicate the participants of the chasing event.

I will refer to ARG relations as **semantic roles**. In this thesis, I take these relations to be generic, shared across the whole lexicon – if the verb were changed in the above example (say from *chased* to *saw*), the ARG roles in the semantic representation would not change. An alternative is to use predicate-specific roles (for example: Pollard and Sag, 1994, pp. 28–29) – in the above example, ARG1 and ARG2 would be replaced by CHASER and CHASED. However, such roles can be straightforwardly generated from the predicate and the ARG role, so such an approach does not provide any additional representational power. Another alternative is to use an intermediate set of roles, as done in FrameNet (Baker et al., 1998) – in the above example, ARG1 and ARG2 would be replaced by THEME and COTHEME, where these roles generalise across a class of predicates (the "cotheme frame", comprised of predicates that indicate motion of two objects). However, Dowty (1991) argues that there is no small set of roles which could be used with a consistent semantic interpretation. This view is defended by Copestake (2009), and is the approach taken by the DELPH-IN[11] consortium – in particular, this includes the English Resource Grammar (ERG) (Flickinger, 2000, 2011), which was used to produce the WikiWoods corpus (Flickinger et al., 2010; Solberg, 2012), used in this thesis.

The line of reasoning that led us to associate event individuals with verbs can be extended to adjectives, adverbs, and prepositions. For example, *the dog ran surprisingly quickly* does not mean that the running was surprising, but rather that the speed was surprising. We can model this by associating an event with *quickly*. Taking this line of reasoning to its conclusion, every predicate applies to a separate individual, and individuals are linked by semantic roles. Viewing the meaning of a sentence as a logical proposition, this means that every predicate is associated with a unique variable, called its **intrinsic variable**.[12]

1.3.2 Situation Semantics

As stated above, the truth of a sentence can only be evaluated relative to a model structure. An important question is then: what does a model structure represent? It is often taken to represent a **possible world** – that is, everything in the universe, either as it actually is, or as it

[11] http://www.delph-in.net

[12] This idea was introduced by Oepen and Lønning (2006), using the term "distinguished variable". Copestake (2009) uses the term "characteristic variable". The term "intrinsic variable" was later agreed on as a neutral compromise; see: http://moin.delph-in.net/ErgSemantics/Basics#Intrinsic_Arguments

could be imagined to be. However, Barwise and Etchemendy (1987)[13] argue that if we treat natural language as describing the entire world, we will derive the wrong truth conditions – for example, if someone observes a card game and utters *Claire has the three of clubs*, it doesn't matter if there is another Claire elsewhere in the world who has the three of clubs; rather, it only matters if, in the current game, someone called Claire has the three of clubs. Barwise and Etchemendy conclude that the truth of a sentence is not evaluated against the entire world, but rather a small part of the world, called a **situation**.[14] This is a somewhat informal notion, since it may not always be clear exactly how a situation is delimited. Furthermore, sizes of situations vary enormously – for instance, the situations studied by astronomers and cellular biologists. The important point is that sentences generally do not describe the entire world.

A detailed framework in which sentences are about situations was proposed by Barwise and Perry (1983). However, as discussed by Stojanovic (2012), the term "situation semantics" now refers to a large class of semantic frameworks which are often formally quite different from Barwise and Perry's original proposal (for an overview of developments, see: Devlin, 2006; Kratzer, 2017).

As a simple approach in the spirit of situation semantics, we can take a situation to formally consist of a small number of related individuals, where situations may overlap with one another. Combining this view with neo-Davidsonian event semantics, described in §1.3.1 above, we can take a situation to consist of a set of individuals, including event individuals, along with semantic roles that relate the individuals to one another. Predicates can be true or false of individuals, and sentences can be true or false of situations.

1.3.3 Dependency Minimal Recursion Semantics

So far, I have represented meanings using predicate-argument structures. However, there are two natural questions that we can ask, following the twin motivations given at the start of this chapter. The first question is scientific: do such representations capture the range of meanings expressed in natural language? The second question is practical: are such representations easy to work with?

To make the first question more precise – our semantic representations should be expressive enough that utterances with different meanings can be given different representations, but not so expressive that we are forced to have many possible representations of each utterance. For example, it should be clear that *a dog chased a cat* and *a cat chased a dog* mean different things and so should be represented differently. On the other hand, our semantic represention should not force us to specify unexpressed details, such as how long the chase lasted, or how far apart the animals were – our representations should be **underspecified** with respect to such details.

[13] I have adapted the example they give on pages 121–122. See also pages 9–12, 28–30, 171–172.

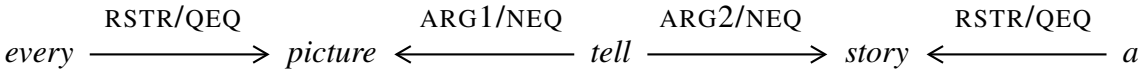
[14] Barwise and Etchemendy call this situation-specific notion of truth "Austinian truth", crediting Austin (1950). While Austin does use truth in this way, no explicit comparison is made with possible-world semantics.

$$\forall x\ \textit{picture}(x) \rightarrow \exists z \exists y\ \textit{tell}(y) \wedge \textit{story}(z) \wedge \textsc{arg}1(y, x) \wedge \textsc{arg}2(y, z)$$

(a) First-order logical proposition (for the most likely scope reading).

$$\begin{aligned}
l_1 &: \textit{every}(x, h_1, h_2),\ \ h_1\ \textsc{qeq}\ l_2 \\
l_2 &: \textit{picture}(x) \\
l_3 &: \textit{tell}(y, x, z) \\
l_4 &: a(z, h_3, h_4),\ \ h_3\ \textsc{qeq}\ l_5 \\
l_5 &: \textit{story}(z)
\end{aligned}$$

(b) MRS representation, underspecifying scope.



(c) DMRS representation, equivalent to the MRS representation above.

Figure 1.2: Comparison of semantic representations for the sentence *every picture tells a story*. For ease of exposition, I have suppressed details such as tense and number. In MRS, these are treated as properties of a variable; in DMRS, they are treated as properties of a node.

Predicate-argument structures naturally deal with these particular issues, but what about other issues?

In the above examples, we expressed the semantics using a set of relations, without any order between the relations. However, in more complicated utterances, some relations may need to take **scope** over others. For example, *Kim knows it didn't rain* and *Kim doesn't know it rained* should be represented differently – in the first case, the negation scopes over *rain* but not over *know*, but in the second case, the negation scopes over both. One of the motivations for developing Minimal Recursion Semantics (MRS) (Copestake et al., 2005) was to use the minimal amount of structure to encode scope. If a sentence allows multiple scope readings, these are underspecified – for example, a sentence with multiple quantifiers has different readings depending on the relative scope of the quantifiers, but it can be given a single MRS representation.

For example, Fig. 1.2a expresses the most likely scope reading of *every picture tells a story*, while the MRS in Fig. 1.2b does not determine the scope. Instead it simply constrains the scope, in the form of the two QEQ constraints. Scope will not play an important role in this thesis, and so I will postpose further explanation and discussion until Chapter 7 (in particular §7.2). The important point to note is that MRS aims to provide logical representations that encode all structural semantic distinctions expressed in natural language – nothing more and nothing less.

To achieve this aim, MRS introduces additional structure beyond the relations and variables we saw before. The aim of Dependency Minimal Recursion Semantics (DMRS) (Copestake, 2009) is to make the resulting structures easier to work with, which answers the second question posed above. With neo-Davidsonian event semantics (see §1.3.1), there is a one-to-one

mapping between predicates and their intrinsic variables. As suggested by Oepen and Lønning (2006), this allows us to have a variable-free semantic representation, where we do not need to distinguish variables and predicates – whenever we need to refer to a variable, we can refer to its corresponding predicate instead.

DMRS represents semantics using a **dependency graph**, consisting of a set of **nodes**, and a set of **links** (or **dependencies**)[15] from one node to another. An example is shown in Fig. 1.2c, which is equivalent to the MRS in Fig. 1.2b. Each predicate (along with its intrinsic variable) is represented as a node. Each semantic role is represented as a link. In addition, each quantifier is represented as a node, with a RSTR link to the node corresponding to the quantified variable. The underspecified scopal constraints used in MRS can be represented as secondary labels on the links (such as /QEQ and /NEQ). As with MRS, I will postpone further discussion until §7.2. In fact, since I do not need scope in most of this thesis, I will use a simplified version of DMRS, where I drop these secondary labels. This is akin to other simplified MRS-based dependency structures such as Elementary Dependency Structures (EDS) (Oepen and Lønning, 2006). For now, the important point is that the full DMRS is equivalent to MRS, which has a well-defined model-theoretic interpretation.

Identifying variables and predicates reduces the number of parts in a semantic representation, but DMRS gives us more than just that. Graphs are a convenient kind of structure, and many efficient graph-based algorithms exist. It is no coincidence that dependency graphs have become popular in NLP, with a variety of dependency graph formalisms. Indeed, comparison between dependency formalisms is an active area of research (for example: Oepen et al., 2016; Kuhlmann and Oepen, 2016). Unlike syntactic dependencies, such as Universal Dependencies (de Marneffe et al., 2014), DMRS abstracts over semantically equivalent expressions, such as *dogs chase cats* and *cats are chased by dogs*.[16] Furthermore, unlike other types of semantic dependencies, including Prague Dependencies (Hajičová, 1998; Bejček et al., 2013), Abstract Meaning Representations (Banarescu et al., 2013), and Universal Conceptual Cognitive Annotation (Abend and Rappoport, 2013), DMRS is interconvertible with MRS, which can be given a direct logical interpretation, as mentioned above.

Finally, MRS has been integrated with the ERG, a broad-coverage grammar of English,[17] which makes it useful in practice. Annotating a corpus with the aid of a grammar leads to higher inter-annotator agreement than directly annotating semantics (Bender et al., 2015), which in turn improves the performance of parsers trained on such corpora (Buys and Blunsom, 2017; Chen et al., 2018). Accurate parsers can then be used to automatically annotate large amounts of data, enabling experiments like those reported in this thesis.

[15] The term "arc" is also used by some authors. Within graph theory, the term "edge" is more common.

[16] More precisely, we can distinguish active and passive voice in terms of their information structure, but I will not deal with that in this thesis (for how to represent information structure in MRS, see: Song, 2017).

[17] Information on the semantic analyses in the ERG is available online, in the documentation produced by Flickinger et al. (2014): `http://www.delph-in.net/esd`

Chapter 2

Modelling Meaning in Distributional Semantics and Model-Theoretic Semantics

As stated in Chapter 1, this thesis is about meaning, so it will be helpful to first clarify what I intend to cover. How should "meaning" be isolated as an object of study?

Koller (2016) contrasts "top-down" and "bottom-up" approaches to semantics. A top-down approach begins with an overarching goal, and tries to build a theory to reach that goal. A bottom-up approach begins with existing techniques, and tries to extend them where possible. Koller observes that model-theoretic semantics is largely top-down, but distributional semantics is largely bottom-up, concluding that "truth-conditional semantics hasn't reached its goal, but at least we knew what the goal was". In contrast, they point out the difficulty in evaluating a bottom-up theory, if there is no goal in mind – "Bottom-up theories are intrinsically unfalsifiable... We won't know where distributional semantics is going until it has a top-down element."

Koller proposes task-based goals for distributional semantics, but this raises a problem – even if we successfully build a model for one task, can we be confident that our model will generalise to another task? In order to develop a model that we could expect to work generally, I will instead take a more long-term perspective. I take the top-down goal to be to characterise the meanings of all utterances in a language.[1] To make this goal more precise, in the following sections I will elaborate on several aspects of meaning which could be considered crucial. I will readily admit that this is an ambitious goal, and I do not claim to reach it in this thesis. Nonetheless, by making the goal explicit, we can assess whether we are heading in the right direction, and we can assess what still needs to be done.

To reach the above goal, what would a semantic model need to have? Many linguists have

[1] To put it another way, we could see the ultimate task-based goal as the task of general-purpose communication with people. In principle, we might try to construct a series of increasingly difficult tasks which build up to this ultimate goal, where the solution for one task is useful for solving the following task. To be more difficult, each task would have to introduce a new challenge compared to the previous ones. Designing such a series of tasks would require identifying important challenges, in a similar way to the overview given in this chapter.

weighed in on this question. For example, Lewis (1970) asserted, "Semantics with no treatment of truth conditions is not semantics." Here, Lewis articulates a necessary requirement of any theory of semantics: it needs to be able to deal with truth conditions. In other words, it needs the notions of truth and falsehood, and it needs to characterise when a statement is true or false.[2]

In the following sections, I discuss a number of challenges which any theory of semantics would need to deal with, and review work in distributional semantics and model-theoretic semantics that aims to answer these challenges.

2.1 Meaning and the World

Language is always *about* something. In this section, I discuss challenges in connecting a semantic theory to things in the world.

2.1.1 Grounding

As Harnad (1990) discusses, if the meanings of words are defined only in terms of other words, these definitions are circular. One goal for a semantic theory is to explain how language relates to the world, including sensory perception and motor control – this process of connecting language to the world is called **grounding**.[3] This includes connecting abstract concepts to the world, although such connections are necessarily more indirect (for further discussion, see: Blondin-Massé et al., 2008; Pecher et al., 2011; Pulvermüller, 2013; Barsalou et al., 2018).

A model-theoretic approach might seem easily grounded, since each individual in a model structure could be a real-world individual. However, the details of how this should be done is not usually articulated in formal semantic accounts – to truly understand a predicate, we need to know its extension in any possible model structure, but if we are presented with a new situation containing a new set of individuals, how do we decide what the extensions are? This is not a trivial task, and should not be brushed aside.

As Harnad points out, a symbolic approach (which would include model theory) could be combined with a neural network architecture. So, could distributional vectors be connected to the world? On its own, a vector does say anything about how it was produced or how it should be interpreted. However, a purely distributional model is not grounded, as it is constructed only using textual information, which has no direct link with the physical world.

[2] In quoting Lewis, I am not committing to their account of truth using possible worlds. Here, I mean to separate semantic problems from proposed solutions to those problems. There are certainly alternative views of truth, such as Brandom (2000)'s inferentialism, where truth is secondary to inference, and Barwise and Perry (1983)'s relation theory of meaning, where truth is secondary to correlations between situations.

[3] A stronger form of the symbol grounding problem considers how an agent could autonomously establish a connection between symbols and the world – in other words, how it could recognise that a symbol is about the world, without already knowing that it is (Steels, 2008; Taddeo and Floridi, 2005, 2007). Here, I discuss a weaker form of the problem – I assume that an agent can already recognise something as being a linguistic signal, about the world, and the challenge is knowing the details of how to match the signal to the world.

In principle, there are several ways we could try to ground a distributional semantic model. The simplest way is to train a distributional model as normal, then combine it with a grounded model. For example, Bruni et al. (2011) **concatenate** distributional vectors and image feature vectors – viewing vectors as arrays, we append one to the other, giving a longer array. This approach has also been applied to other senses. For example, Kiela et al. (2015) use olfactory information, and Kiela and Clark (2017) use both visual and auditory information. However, while there is grounded information in the sensory dimensions, concatenation leaves the distributional dimensions ungrounded, because we cannot relate them to the world. It is tempting to respond to this criticism by looking for correlations between the distributional and sensory features. For example, Bruni et al. (2014) perform singular value decomposition (SVD) on concatenated vectors, and Silberer and Lapata (2014) train an autoencoder on concatenated vectors. However, there is no guarantee that every distributional feature will correlate with sensory features. Distributional features without correlations will remain ungrounded. Indeed, in the above works, distributional vectors were found to be more useful than sensory vectors, at least when evaluating on similarity datasets. This suggests that there might be more information in the distributional vectors than the sensory vectors, which means it would not be possible to ground all dimensions by directly finding correlations – at the very least, it would be necessary to come up with more sophisticated ways to relate distributional features to sensory features.

A second approach is to train distributional vectors as normal, and then to interpret them via a mapping to grounded representations. For example, Lazaridou et al. (2014) and Bulat et al. (2016) learn a mapping from distributional vectors to visual vectors (and vice versa). However, grounding by interpretation runs into the same problem as grounding by correlation – there is no guarantee that every dimension can be directly interpreted in this way. As mentioned above, this is a serious problem if there is more distributional information than sensory information.

In a similar vein, Mitchell et al. (2008) map distributional vectors to fMRI scans of human brain activity, and Făgărăşan (2015) map distributional vectors to "feature norms" (McRae et al., 2005), which are manually produced lists of properties (for example, the feature norm for *apple* includes the property *is edible*). While these are potentially interesting ways to interpret distributional vectors, both map to ungrounded spaces – given an fMRI scan, we need to understand how brain activations are grounded, and given a property expressed in natural language, we need to understand how that linguistic expression is grounded.

Finally, a third approach is **joint learning** from both distributional and grounded data – we define a single model, whose parameters are learnt based on both sources of data. For example, Feng and Lapata (2010) train a Latent Dirichlet Allocation (LDA) model (Blei et al., 2003) for both words and "visual words" (clusters of visual features). Lazaridou et al. (2015) use a Skip-gram model (Mikolov et al., 2013) to jointly predict both words and images (more precisely, to predict vector representations of images, which were produced using a pre-trained convolutional net). Kiros et al. (2014) embed both text and images in a single space, training a recurrent

net to process captions, and a convolutional net to process images. Unlike pure distributional models, which look for patterns in the co-occurrence of words, these joint models will prefer co-occurrence patterns that match the sensory data. For this reason, I believe joint learning is the right approach to ground distributional data – we can connect our semantic representations to grounded data from the outset, rather than trying to make such connections after the fact.

However, we need to make sure that all distributional features are grounded. For example, with Feng and Lapata's LDA model, there is nothing stopping some topics from almost entirely generating words rather than "visual words". Similarly, with Lazaridou et al.'s joint Skip-gram model, there is nothing stopping some embeddings from almost entirely predicting words rather than images. Conversely, we also need to make sure that we fully make use of distributional data, rather than discarding patterns that are difficult to ground. For example, Kiros et al.'s joint embedding model aims to embed sentences in a way that is useful for matching them to images. It is not obvious how this approach could be extended so that we can learn sensible embeddings for sentences that cannot be easily depicted in an image. This leads us to the following question – how should a joint architecture be designed, so that we can fully learn from distributional data, but while also ensuring that the semantic representations are fully grounded?

In the following section, I discuss how words relate to the world. Clarifying this relationship should help us to design architectures which we can reasonably expect to produce fully grounded representations.

2.1.2 Concepts and Referents

How do meanings relate to the world? The model-theoretic answer is that we can describe the world in terms of individuals, and we can represent meaning in terms of extensions (sets of individuals). Situation semantics (discussed in §1.3.2) further clarifies that an utterance relates to a situation, rather than the entire world. The challenge, as mentioned at the start of §2.1.1 above, is to be able to relate an utterance to a *new* situation, where the individuals have not previously been observed. One goal for a semantic theory is to have representations that are not directly expressed in terms of extensions, so that we can generalise to new situations.

To achieve this, we need to distinguish a **concept** (roughly speaking, the meaning of a word) from a **referent** (an individual in the concept's extension).[4] The importance of this distinction has been noted for some time (for example: Ogden and Richards, 1923).[5] Following Murphy

[4] In the psychological literature, the term "category" refers to a set of individuals (for example: Smith and Medin, 1981; Murphy, 2002). For a category corresponding to some concept, this is synonymous with "extension".

[5] This may remind some readers of the work of Frege (1892) or Peirce (1867), but I do not want to make the same distinction here. Frege distinguishes *Sinn* ("sense") and *Bedeutung* ("reference"). However, the Sinn of a linguistic expression is not fully formalised, and it is described as independent of any person's mind, a claim which I do not want to make. Meanwhile, the Bedeutung of a sentence is taken to be a truth value, rather than a situation (see §1.3.2 and Barwise and Perry, 1983, pp. 22–26). Peirce proposes a triadic structure of signs, and distinguishes "interpretants" and "objects". However, the interpretant is itself a sign, which leads to an infinite chain of signs, as Peirce acknowledged. This is aesthetically curious, but of little practical use.

(2002, pp. 4–5), I use the term "concept" without committing to a particular theory of concepts.

There is no quick fix for an extensional model theory. Some accounts contrast extensions with some notion of **intension**. For example, we might define an intension to be the set of properties common to all members of a extension (for example: Arnauld and Nicole, 1662; Jones, 1911).[6] This is sometimes called the **classical theory** of concepts (for example: Murphy, 2002; Margolis and Laurence, 2011). Representing a concept by a set of properties allows us to deal with new situations, because a new individual can be classified on the basis of its properties. However, Wittgenstein (1953, §66–71) argues against such a view of meaning, using the German word *Spiel* ("game" or "play") as an example – there is no set of necessary and sufficient conditions that characterises all types of Spiel. For example, compare chess, solitaire, tennis, and children kicking a ball against a wall. Rather than a common set of properties, we can observe a **family resemblance** between these different types of Spiel. Furthermore, Rosch (1975, 1978) experimentally demonstrated typicality effects which suggest that concepts are not structured as sets of necessary properties – some referents are seen as being more typical examples of a concept than other referents. For example, experimental participants judged oranges and apples to be more typical examples of fruit than avocadoes and pumpkins. So, while representing a concept as a set of properties may allow generalisation to new individuals, it is too simplistic a model.

Carnap (1947) defined an intension to be a function from possible worlds to extensions. This avoids the above problem, because for any possible world, the intension gives us the extension in that world. However, representing a concept by such a function just moves the problem further on. How do we explicitly represent such a function? Since there are infinitely many possible worlds, it would be psychologically impossible to enumerate an extension for each possible world, so we need a finite representation of the function. This unfortunately takes us back to our original problem – how do we determine an extension for each possible world?

The above definitions of "intension" do not solve the problem of how to determine extensions in new situations. However, given the philosophical and psychological arguments that the structure of concepts is complicated, perhaps it would be unreasonable to expect a purely formal solution to the problem. If concepts have too many details to write down by hand, we may need machine learning techniques to fill in the details. In this light, distributional semantics seems like a promising approach to constructing conceptual representations, as mentioned in §1.2.

However, even supposing we can construct grounded vectors, as discussed in §2.1.1 above, there is still the question of how to relate such a concept vector to individuals in the world. One option is to embed both concepts and individuals in the same vector space. In this case, we need some way to decide how close the vectors need to be, before we say that the individual is part of the concept's extension. A second option is to embed concepts and individuals in distinct vector spaces. In this case, we need some way to relate the two vector spaces. In both cases, we

[6] Arnauld and Nicole use the terms "compréhension" and "étendue", instead of "intension" and "extension".

need additional structure beyond vectors.

One solution to this problem is to represent a concept, not by a point, but by a **region** – a subset of the space, with a well-defined boundary. Individuals embedded inside the region are part of the extension, and individuals embedded outside the region are not. However, without any constraints on possible regions, knowing that one point is inside the region would tell us nothing about any other point being inside or outside. This would make it difficult to generalise, and hence difficult to learn such a concept. Learning will be further discussed in §2.4, but for now we can note that some kind of constraint is necessary. Gärdenfors (2000, 2014) argues in favour of a specific constraint, that a concept should be modelled as a **convex** region – any straight line between two points in the region will lie entirely inside the region. Given examples of a concept, the convexity assumption would allow us to generalise to points between the examples (the simplex spanning those examples).

Building on this idea, McMahan and Stone (2015) learn representations of colour terms, which are grounded in a well-understood perceptual space. To allow uncertainty, they model the meaning of a colour term as a distribution over convex regions. To make such distributions tractable, they assume cuboidal regions, where the faces are independently distributed. This work shows the feasibility of representing meaning in this way, but it may be challenging to scale up their model to larger domains, where suitable representations of individuals may not be known. For distributional semantics, we do not observe individuals at all.

Rather than representing a concept as a region, an alternative approach is to represent a concept as a **binary classifier** – a function that maps each possible input into one of two classes. In this case, the inputs represent individuals, and one class is the concept, while the other class is everything else. This approach ties in with a view of concepts as abilities, as proposed in some schools of philosophy (for example: Dummett, 1976, 1978; Kenny, 2010), and some schools of cognitive science (for example: Murphy, 2002, pp. 1–3, 134–138; Zentall et al., 2002).

Within NLP, some authors have also suggested representing concepts as classifiers. Larsson (2013) represents the meaning of a perceptual concept as a classifier of perceptual input, in the framework of Type Theory with Records (see §2.5.4). Schlangen et al. (2016) train image classifiers using captioned images, and Zarrieß and Schlangen (2017a,b) build on this, using distributional similarity scores to help train such classifiers, by generalising one label of an image to other similar labels. However, these approaches do not learn distributional representations.

Both of the above approaches to representing concepts (as regions and as classifiers) allow us to model how concepts relate to individuals, including how to determine the concept's extension in a new situation. Links between the two approaches will be explored in §3.4.1.

Such representations have seen little use in distributional semantics. However, Erk (2009a,b) learns distributional representations in the form of regions. They start with normal word vectors, then use these to produce context-specific vectors (see §2.3.3 below), then use these to learn regions of vector space. However, as such a model requires pre-trained vectors, some

information may have already been lost in constructing these vectors. To avoid losing information, it would be preferable to learn semantic representations directly. This would also make it easier to implement joint learning, as argued for in §2.1.1 above.

2.2 Lexical Meaning

Language is made up of words.[7] Every speaker of a language will presumably have some kind of mental lexicon, that contains the meaning of every word they know. In this section, I discuss challenges in representing the meanings of individual words.

2.2.1 Vagueness

In many cases, individuals can fall along a continuum without a sharp cutoff between a predicate being true or false. For example, we can place colours (in human perception) in a three-dimensional space of hue, saturation, and brightness.[8] We can smoothly change between two colours (say, *pink* and *red*), but there is not an exact point where we switch from one term to the other. This is called **vagueness** (or **gradedness**) – predicates have borderline cases where truth values are unclear. One goal for a semantic theory is to account for this.

Vagueness is the basis of the **Sorites Paradox** (for an overview, see: Hyde and Raffman, 2018). Suppose we have one point along a continuum where a predicate is true, and another point where it is false (for example, one shade of colour that is red, and one that is not red). Suppose further, that if two points on the continuum are sufficiently close, they are indiscriminable – the difference is not perceptible, so the predicate must take the same truth values for those two points. This leads to a paradox, since we can find a sequence of points, where each is indiscriminable with the next, but the predicate is true at the start of the sequence, and false at the end. A good account of vagueness should avoid this paradox.

Vagueness has also found experimental support For example, Labov (1973) investigated the boundaries between concepts like *cup*, *mug*, and *bowl*, asking participants to name drawings of objects in different contexts of use. For typical instances of a concept (such as a typical cup, being used to drink coffee), the term was consistently applied; meanwhile, for objects that were intermediate between two concepts (for example, an object that's wide for a cup but narrow for a bowl), terms were used inconsistently. For these borderline cases, a single person may even make different judgements at different times (McCloskey and Glucksberg, 1978).

There is a large literature on extending model theory to account for vagueness (for an overview, see: Sutton, 2013, chapter 1; Van Deemter, 2010). A direct approach is to use **fuzzy**

[7] I use "word" in a pre-theoretic way, to avoid clunkier terms like "lexical item" or "listeme". I am not committing to a precise definition, which may be problematic (Haspelmath, 2011) and unnecessary (Emerson and Copestake, 2015). In any case, it is not important for this thesis, which is about semantics, not syntax.

[8] As argued by Saunders and Van Brakel (1997), this is a simplification of how colour terms are actually used. Nonetheless, even this simplified model of colour presents us with an interesting modelling challenge.

truth values, as proposed by Zadeh (1965, 1975) – truth values are no longer binary, but rather values in the range $[0, 1]$, where 0 represents definitely being false, 1 represents definitely being true, and intermediate values represent a range of borderline cases. Fuzzy logic has not seen much use within computational linguistics. One exception is Bergmair (2010), who introduced the framework of Monte Carlo Semantics, using fuzzy truth values to allow graded inferences.

Alternatively, we can stick to binary truth values (truth and falsehood), but represent uncertainty about a truth value as a **probability**. Sutton (2017) contrasts two places where we can put this uncertainty. One option is to say that the meaning of each lexical item is a non-vague predicate, but any speaker is uncertain about what this meaning is. For example, a speaker would believe that *red* corresponds to a precise range of colours, but would be uncertain about which range of colours this is. The other option is to say that predicates are inherently vague, so that they assign a probability of truth to each individual. For example, a speaker could be certain about the meaning of *red*, but remain uncertain about whether a particular colour should be considered red – there is simply variation in how the term is used. Lassiter (2011) and Fernández and Larsson (2014) take the former approach, while Sutton (2015, 2017) takes the latter. Both approaches can model vagueness, and avoid the Sorites Paradox.

At the level of a single predicate, there is not much to decide between fuzzy and probabilistic accounts. However, we will see in §2.3.2 that they behave rather differently at the level of sentences. I will further discuss the two kinds of probability (and non-probabilistic notions of uncertainty) in §3.4.2, after presenting my framework.

Uncertainty has also been incorporated into distributional vector space models. Vilnis and McCallum (2015) extend Mikolov et al.'s Skip-gram model to allow uncertainty, by representing meanings as Gaussian distributions over vectors. Barkan (2017) incorporate uncertainty into Skip-gram using Bayesian inference – rather than viewing learning as optimising word vectors, they view learning as finding the posterior distribution over word vectors, given the observed data. They approximate this posterior as a Gaussian distribution (in order to keep calculations tractable), so both of these approaches produce the same kinds of object. Balkır (2014), working within the type-driven tensorial framework (see §2.5.3), uses the quantum mechanical notion of a "mixed state" to naturally model uncertainty in a tensor. For example, this approach replaces vectors by matrices.

While these approaches represent uncertainty, it is challenging to use them to represent vagueness, which was defined above in terms of truth values. This basic problem is this: a distribution allows us to *generate* instances of a concept, but how can we go in the other direction, to *recognise* instances of a concept? It is tempting to classify a point using the probability density at that point – we might say that points with higher probability density are more likely to be classified as instances of the concept. However, if we compare a more general term (like *red*) with a more specific term (like *scarlet*), we run into a problem – a more general term will have its probability mass spread more thinly, and hence have a lower probability density than

the more specific term, even if both terms could be considered true. I argued in §2.1.2 that we need to represent predicates as regions of space or as classifiers, and while a distribution over a space might at first sight look like a region of space, it is a different kind of object. The contrast between the two will be discussed in more detail in Chapter 3.

2.2.2 Polysemy

The meaning of a word can often be broken up into a number of different uses, each called a **sense**. When these senses are related, they are called **polysemous**, and the phenomenon is called **polysemy**. For example, *school* can refer to a physical building, or to an institution composed of the staff and students. However, these two senses are related, since a school building is used by a school institution. This can be contrasted with **homonyms**, which are unrelated senses. For example, *school* can also refer to a group of fish, with identical spelling and pronunciation to the education sense(s). However, the fish sense and education sense(s) are not connected (and in fact have different etymologies). All of the above senses of *school* are also **lexicalised** – they are established uses of the word, which a proficient speaker would presumably have committed to memory, rather than inferring them from the context. In contrast, suppose a speaker sees a number of planes, flying in a way that reminds them of a school of fish. If the speaker referred to the planes as a "school of planes", a listener might understand the phrase in context, even if they had never anticipated such a use of the term "school". I will discuss context-dependent meaning in §2.3.3, and in this section I will focus on lexicalised meaning. For a semantic theory that aims to capture all lexicalised meanings (which includes conventional metaphors), one goal is to model how a word can have a range of polysemous senses.

In model theory, one solution is to define a separate predicate for each sense, so that each sense has different truth conditions. However, deciding on a discrete set of senses is difficult, and practical efforts at compiling dictionaries have not provided a solution. Indeed, the lexicographer Sue Atkins bluntly stated, "I don't believe in word senses".[9] Although the sense of a word varies across individual usages, there are many ways that we could divide usages into a discrete set of senses, a point which has been made by a number of authors (for example: Spärck-Jones, 1964; Kilgarriff, 1997, 2007; Hanks, 2000; Erk, 2010). To quantify this intuition, Erk et al. (2009, 2013) asked annotators to judge the similarity between dictionary senses, as well as the similarity between individual usages. The similarity judgements suggest that usages cannot always be neatly clustered into discrete senses, implying that word senses do not have clear boundaries between each other. A good lexical semantic theory should therefore be able to capture variation in meaning without resorting to finite sense inventories.

Alternatively, we could represent all of the polysemous senses together as a single predicate. Indeed, Ruhl (1989) argues that even highly frequent terms with many apparent senses, such as *bear* and *hit*, can be assigned a single underspecified meaning, with the apparent diversity of

[9] Kilgarriff (1997) and Hanks (2000) both quote Atkins.

senses explainable from the context. They conclude that we should initially assume that each word has a single sense, and resort to multiple senses only when we fail to identify a single sense. The challenge would then be, firstly, to accurately represent such meanings without overgeneralising to cases where they wouldn't be used, and secondly, to model how meanings become specialised in context, revealing different facets. The latter half of this challenge will be further discussed in §2.3.3 below.

In vector space models, we have a similar problem. Standard approaches to distibutional semantics produce a single vector that encodes all observed occurrences of the word, which combines all senses into a single vector. An alternative is to use multiple vectors to represent different senses (for example: Schütze, 1998; Rapp, 2004), but this falls prey to the same criticism raised against model-theoretic semantics above.

However, I have already argued in the above sections that we should move away from vector space models that represent each word as a single point. As discussed in §2.2.1 above, some previous work has instead replaced points with distributions, and these approaches have also been applied to modelling word senses. For example, Athiwaratkun and Wilson (2017) use a mixture of Gaussians, extending Vilnis and McCallum's model described in §2.2.1 above, to allow multiple senses. However, a mixture of a Gaussians ultimately models a fixed number of senses (one for each Gaussian), and so this also falls prey to the above criticism of finite sense inventories. In principle, modelling a word as a distribution could be done in a way that avoids this criticism, but this would require moving beyond finite mixture models. In the type-driven tensorial framework (see §2.5.3), Piedeleu et al. (2015) use mixed quantum states, similarly to Balkır's approach discussed in §2.2.1 above. Although they only propose using this approach for homonymy, it seems plausible that it could be extended to polysemy as well, although care would be needed to avoid finite sense inventories.

If a word is represented by a region of space, or by a classifier, we don't have the problem of finite sense inventories, since we can continuously vary between nearby points. I will discuss how polysemy can be modelled under this approach in §3.7.2, after introducing my framework.

2.2.3 Hyponymy

The above two sections (§2.2.1 and §2.2.2) discussed representing the meaning of a single word. However, words do not exist on their own, and one goal for semantic theory is to describe relations between them. A classic relation is **hyponymy**,[10] which describes when one item (the **hyperonym**) has a more general meaning than another (the **hyponym**).[11] Words that share a hyperonym are called **co-hyponyms**.

In model theory, hyponymy can be defined straightforwardly – a predicate P is a hyponym

[10] This is also referred to as "lexical entailment", making a link with logic (see §2.3.2).

[11] An alternative term for "hyperonym" is "hypernym". This is unfortunate, since "hypernym" and "hyponym" sound the same in my dialect.

of another predicate Q, if the extension of P is a strict subset of the extension of Q. This directly formalises the idea that P has a more specific meaning than Q.

In distributional semantics, hyponymy is more challenging. Given two vectors, it is not clear how to say if one is more general than the other. Nonetheless, there have been proposals to measure hyponymy of vectors. These are generally based on the **Distributional Inclusion Hypothesis**, which states that a hyperonym occurs in all the contexts of its hyponyms (for example: Weeds et al., 2004; Geffet and Dagan, 2005). For vectors produced using the count method, we might say that a vector with more nonzero entries is more general, as that term has appeared in more varied contexts. Several hyponymy measures have been proposed based on this intuition. For example, Kotlerman et al. (2009, 2010) define the "balAPinc" measure, which combines a measure of feature overlap with a measure based on information retrieval. Herbelot and Ganesalingam (2013) view a word vector as defining a distribution over contexts, and propose using Kullback-Leibler (KL) divergence to determine hyponymy. Rei (2013) gives an overview of hyponymy measures, and proposes a weighted cosine measure, which proved effective for hyponym generation.

For embedding vectors, it is not obvious that such measures can be used, as the dimensions do not directly correspond to contexts. Nonetheless, the dimensions can be viewed as combinations of contexts (as discussed in §1.2.1). Indeed, Rei and Briscoe (2014) empirically finds that embedding vectors perform almost as well as count vectors.

However, the Distributional Inclusion Hypothesis can be questioned. Following the Gricean Maxim of Quantity (Grice, 1967), a speaker is likely to choose an expression with a degree of generality appropriate for the context, and hence hyponyms are unlikely to appear in the same contexts as more general terms. Rimell (2014) points out that some contexts are highly specific. For example, *lion* is a hyponym of *animal*, so we would predict the contexts of *lion* to also appear as contexts of *animal*. However, while *mane* is a likely context of *lion*, it is not a likely context of *animal*. To avoid this problem, Rimell uses a notion of topic coherence to determine hyponymy, showing that the contexts of a general term minus those of a hyponym are still coherent, while the converse is not true.

Even in recent shared tasks on hyponym detection (Bordea et al., 2016; Camacho-Collados et al., 2018), many systems make use of pattern matching, following Hearst (1992). For example, after observing a string of the form *X such as Y*, we might infer that *Y* is a hyponym of *X*. In the above shared tasks, the best performing systems did not rely solely on distributional vectors, but used pattern matching as well. This illustrates the difficulty in determining hyponymy from distributional vectors alone.

If we move away from count vectors, there are other options. One option is to build the hyponymy relation into the definition of the space. For example, we might say that vectors closer to the origin are more general. Vendrov et al. (2016) use non-negative vectors, where one vector is a hyponym of another if it has a larger value in every dimension. They train a

model on WordNet (Fellbaum, 1998), which contains a hierarchy of hyponymy relations. Li et al. (2017) extend this, employing joint learning on both WordNet and raw text. However, for a hierarchy like WordNet, there are exponentially more words lower in the hierarchy. This cannot be embedded in Euclidean space without words lower in the hierarchy being increasingly close together. To avoid this problem, Nickel and Kiela (2017) propose using hyperbolic space, where the volume of space increases exponentially as we move away from the origin. They train a model on WordNet, trying to place hyponyms close to their hyperonyms. In principle, hyperbolic embeddings could be trained on text as well, but to my knowledge this has not been tried. However, this approach does not generalise to non-tree hierarchies – for example, WordNet gives *bass* as a hyponym of *singer*, *voice*, *melody*, *pitch*, and *instrument*. Requiring that *bass* is represented close to all its hyperonyms would force them all to be close as well, which we do not want, as they are in very different parts of the WordNet hierarchy (including both physical and abstract entities).

If hyponymy is not built into the space, we can view hyponymy classification as a supervised learning task. For example, Weeds et al. (2014) train an SVM to classify if a pair of words exhibit hyponymy or co-hyponymy. Rei et al. (2018) train a neural network to predict hyponymy, using the HyperLex dataset (Vulić et al., 2017). While such approaches might be useful for downstream tasks, or useful as a way to evaluate the quality of semantic representations, a supervised method cannot explain how people learn hyponymy relations in an unsupervised way. Furthermore, this effectively treats hyponymy as an opaque relationship between feature vectors. This makes it difficult to analyse why one vector is classified as a hyponym of another, and makes it unclear whether the trained classifier will generalise to new domains. Indeed, Levy et al. (2015b) found that such classifiers mainly learn which words are common hyperonyms.

If we move away from representing words as vectors, it can be easier to define hyponymy. As discussed by Erk (2009a,b) and Gärdenfors (2014, §6.4), modelling meaning as a region of space provides a natural definition – P is a hyponym of Q if the region for P is contained in the region for Q. Modelling meaning as a probability distribution also allows a notion of hyponymy, although it is slightly harder to define than for regions, because a distribution over a smaller region also assigns more probability mass to that region. Vilnis and McCallum (2015) propose using KL-divergence to measure hyponymy. Athiwaratkun and Wilson (2018) build on this, using a thresholded version of KL-divergence, and training on WordNet. Balkır (2014) proposes using a quantum mechanical version of KL-divergence, in the type-driven tensorial framework (see §2.5.3). This has been extended to phrases and sentences (Balkır et al., 2015; Sadrzadeh et al., 2018).

A region-based approach to hyponymy is ultimately the approach taken in this thesis. Not only does it give a simple definition, but representing a concept as a region is also motivated for other reasons, discussed elsewhere in this chapter. I will further discuss hyponymy using regions in §3.7.2, after I have introduced my own framework.

2.3 Sentence Meaning

In the previous section, I discussed meaning at the level of individual words. I now turn to challenges in representing meaning at the level of sentences.[12]

2.3.1 Compositionality

A notable feature of language is that it is **productive** – a fluent speaker of a language can easily understand a sentence they have never heard before, as long as they know each of the words in the sentence. This means that one goal for a semantic theory is to be able to *derive* the meaning of a sentence from its parts, so that it can generalise to new combinations of words. This is known as **compositionality**.

Compositionality should not be confused with **disambiguation** – words often have multiple senses (as discussed in §2.2.2 above), and when two expressions are composed to form a larger expression, they also mutually disambiguate one another. For example, the word *school* is ambiguous between its educational and group-of-fish senses, but it is disambiguated once combined with words like *building* or *fish*. Kartsaklis et al. (2013) discuss how composition and disambiguation have often been conflated in the distributional semantics literature. The focus in this section is on deriving semantic representations for larger expressions. Disambiguation can be seen as a kind of context dependence, which I discuss in §2.3.3 below.

A strength of model theory is that compositional mechanisms have been developed that can deal with a wide range of constructions. The classic approach is to use λ-calculus, although alternatives exist, including a composition algebra for Minimal Recursion Semantics (Copestake et al., 2001; Copestake, 2007). Importantly, these compositional mechanisms are tightly constrained, which means that relatively simple composition rules can derive the semantics for complex sentences. Furthermore, this compositional process results in logical representations, as will be discussed in §2.3.2 below.

Distributional semantic models generally learn meanings for individual words. Supposing this can be done well, we have the challenge of how to compose representations of words, to construct representations of larger phrases. Vector spaces are not equipped with compositional operations that correspond to compositional operations in formal semantics. One option is to assume that the meanings of phrases should also be represented as vectors. Then, our first decision is whether phrases use the same space as for words.

If we use the same vector space for words and phrases, the challenge is then to find a composition function that maps a pair of vectors to a new vector. Mitchell and Lapata (2008, 2010) compare a variety of such functions, but they find that componentwise multiplication is in fact as

[12] Just as with "word", I use "sentence" in a pre-theoretic way. For some languages, such as Thai, sentence boundaries are not indicated in the standard orthography (for discussion, see: Aroonmanakun, 2007). Even for a language like English, sentence segmentation is not trivial (for discussion, see: Palmer, 2000).

good as or better than the other functions they consider,[13] despite being commutative, and hence insensitive to word order. The unexpected effectiveness of componentwise multiplication has been replicated in a number of other studies (for example: Baroni and Zamparelli, 2010; Blacoe and Lapata, 2012; Rimell et al., 2016). However, it is unclear how to adapt multiplication so that word order is taken into account, and Polajnar et al. (2014b) demonstrate that performance degrades with sentence length.

Alternatively, we can try to use a sentence vector space which is distinct from the word vector space. To do this, we need some way to define a sentence space, and this is generally done by taking a task-based perspective – words are combined into sentence representations, which provide useful features for solving some task. For example, a recurrent neural network (RNN) processes text one token at a time, updating a hidden state vector at each token. The final hidden state can be seen as a representation of the whole sequence. To make the composition more linguistically informed, the recurrent steps in the RNN can also be defined to follow a tree structure, rather than linear order (for example: Socher et al., 2010, 2012). The parameters in the RNN can be optimised either for a particular supervised task, such as machine translation (for example: Cho et al., 2014), or for an unsupervised objective to model the input, as in an autoencoder (for example: Hermann and Blunsom, 2013). However, as argued at the beginning of this chapter, if we take a task-based perspective, it is difficult to know if the approach will generalise. Even if the network successfully learns to encode as a vector all the semantics relevant for a particular task, we may not be confident that the same neural architecture will work for another task.

Rather than representing all words as vectors, the type-driven tensorial framework represents words as tensors (see §2.5.3). This framework is naturally compositional, as the tensors are defined so that tensor contraction matches predicate-argument structure. In this framework, there is one vector space for nouns and another for sentences. Polajnar et al. (2015) explore distributional sentence spaces, where dimensions are defined by co-occurrences, just as for standard distributional vectors. This corresponds to the first approach above, where the same vector space is used for words and phrases. In principle, the task-based approach could also be applied to this framework, but I am not aware of any work that has done this.

However, a weakness with all of these approaches is that they map sentences to a fixed finite-dimensional space. As Mooney (2014) put it, "You can't cram the meaning of a whole %&!$# sentence into a single $&!#* vector!" More precisely, as we increase sentence length, the number of possible sentences with distinct meanings increases exponentially – simple examples can be constructed with coordination or relative clauses, such as *the dog chased the cat*, and *the mouse saw the cat which scared the dog*, and so on. For general-purpose semantics, each of these meanings should be kept distinct, which leaves us two choices. If distinct meanings are to

[13] Ganesalingam and Herbelot (2013) give a mathematical analysis of Mitchell and Lapata's composition functions, explaining the poor performance of tensor products and circular convolution.

be kept a certain distance apart from one another, then the magnitudes of sentence vectors need to increase exponentially with sentence length, so that the meanings can be distinguished.[14] Alternatively, if we allow distinct meanings to be arbitrarily close to one another, then we need to record each vector component to many significant digits in order to accurately represent the meaning. In this case, the fine-grained structure of the space is important for meaning. However, small changes to model parameters would cause drastic changes to this fine-grained structure. Although both of the above approaches are possible in theory,[15] I do not know of any work that has explored either in practice. Without doing this, we are forced to view sentence vectors as the output of lossy compression.[16]

Although it may be useful in many situations to compress information, I do not believe that a semantic theory should force composition to involve compression. Full and detailed semantic representations should also have their place. This is particularly important if we would like the theory to have continuing relevance at a discourse level. It would be absurd to represent, as vectors of comparable size, both a five-word sentence and the entirety of the English Wikipedia. However, this leaves open the question of how we *should* represent sentence meaning. In the following section, I will turn to logic as a guide for constructing sentence representations.

2.3.2 Logic

Sentences can be used to express complex thoughts, and build chains of reasoning. **Logic** formalises this, and one goal for a semantic theory is to support the logical notions of **truth**, discussed in §1.3, and **entailment**, which holds when one proposition follows from another. In model-theoretic semantics, entailment is defined in terms of truth – given that one proposition is true, we can ask whether another proposition must also be true. In contrast, **proof-theoretic** semantics takes entailment to be the primary notion (for example: Brandom, 2000) – a proposition is only true relative to some set of propositions which a speaker is already committed to. Under either view, the process of deciding if an entailment holds is called **inference**.[17]

One of the strengths of model theory is its logical foundation, allowing us to evaluate the truth of a proposition in a model structure. However, the choice of logic varies between model-theoretic approaches. I argued in §2.2.1 above, that probabilities of truth and fuzzy truth values allow natural accounts of vagueness, and it is indeed possible to construct corresponding logics.

In probability logic, propositions have probabilities of being true or false, and there is a joint distribution for the truth values of all propositions (for expositions, see: Adams, 1998; Demey

[14] This argument can be formalised information-theoretically. Consider sending a message as an D-dimensional vector, though a noisy channel. If there is an upper bound K to the magnitude of the vector, then the channel has a finite *channel capacity*. The capacity scales as K^D, which is only polynomial as a function of K.

[15] If cosine similarity is used, vector magnitudes are ignored, so only the second approach is possible.

[16] This conclusion has been drawn before (for example: Goodfellow et al., 2016, p. 370), but I believe my argument is novel, and makes the statement more precise.

[17] Icard (2014) breaks this down further, separating *what* entailments hold, *how* an entailment can be inferred, and *why* an agent would infer one entailment rather than another.

et al., 2013). In fuzzy logic, propositions have fuzzy truth values, and classical logical operators (such as: $\wedge$, $\vee$, $\neg$) are replaced with fuzzy versions (for expositions, see: Hájek, 1998; Cintula et al., 2017). Fuzzy operators act directly on truth values – for example, given the fuzzy truth values of propositions p and q, we can calculate the fuzzy truth value of the disjunction $p \vee q$. In contrast, in probability logic, given probabilities of truth for p and q, we cannot calculate the probability of truth for $p \vee q$, unless we know the joint distribution. A problem with the fuzzy approach, observed by Fine (1975), comes when we consider propositions like $p \vee \neg p$. Intuitively, this should be true regardless of what p is, but fuzzy logic can give a truth value below 1. This makes fuzzy logic less appealing (or at least, harder to interpret). However, Hájek et al. (1995) prove that fuzzy logic can provide upper and lower bounds on probabilities.

In computational linguistics, there are a couple of relatively well-developed frameworks for probabilistic semantics, which will be discussed in §2.5.4. Although they can take advantage of work on probabilistic logic, they do not provide a methodology for distributional semantics.

For current approaches to distributional semantics, logic is a challenge, as vector spaces do not inherently have any logical structure. For the task called **recognising textual entailment** (RTE), state-of-the-art models implicitly take a proof-theoretic approach, by framing entailment as a classification task – given a pair of sentences, called the **premise** and **hypothesis**, the task is to decide whether the premise entails the hypothesis, contradicts it, or neither. Datasets include SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018). A variety of neural network architectures have been applied to this task, achieving relatively high performance (for example: Yin et al., 2016; Rocktäschel et al., 2016; Wang and Jiang, 2016; Cheng et al., 2016). However, it is difficult to analyse direct approaches that do not use an explicit logic. In fact, although performance appears high, Gururangan et al. (2018) suggest that this may be due to artifacts from the annotation process. They find that, only looking at the hypothesis and not the premise, a simple model can achieve 67% on SNLI and 53% on MultiNLI, which is much higher than the majority class baseline (34% and 35%, respectively). This suggests that performance on such datasets may be an overestimate of the ability of neural models to perform inference.

If we want to explicitly represent logical structure, there are a few options. One is to build a hybrid system, which combines a vector space with a logic. For example, Herbelot and Vecchi (2015) aim to give logical interpretations to vectors. They consider a number of properties (such as: *is_edible*, *has_a_handle*, *made_of_wood*), and for each, they learn a mapping from vectors to values in the range $[0, 1]$, where 0 means the property applies to no instances of the concept, and 1 means it applies to all instances. For example, for the property *is_edible*, we want to predict a high value for the *plum* vector, and a low value for the *tricycle* vector. This is an interesting way to probe what information is available in distributional vectors, but it is unclear how it could be generalised to deal with individuals (rather than the entire extension of a concept), or to deal with complex logical propositions, (rather than atomic properties).

Garrette et al. (2011) and Beltagy et al. (2016) incorporate a vector space model into a

Markov Logic Network (Richardson and Domingos, 2006), a kind of probability logic. If two predicates have high distributional similarity, they add a probabilistic inference rule saying that, if one predicate is true of an individual, the other predicate is likely to also be true. The higher the similarity score, the stronger the inference rule. This approach allows us to use distributional vectors in a well-defined logical model, but it assumes we can interpret similarity in terms of inference (an assumption explored by Erk (2016) and discussed in §2.5.2). As argued in §2.1 above, pre-trained vectors may have already lost information, so it would be preferable to learn logical representations directly.

Lewis and Steedman (2013) use distributional information to cluster predicates. For example, *author* and *write* could be considered to encode the same underlying predicate, since if x is the author of y, then x wrote y (and vice versa). Using corpus data, they cluster predicates that are observed to hold of the same pairs of individuals – for example, *author*(*Rowling*, *Harry Potter*) and *write*(*Rowling*, *Harry Potter*). This uses distributional information directly, rather than pre-trained vectors. Although Lewis and Steedman use a classical logic with the underlying predicates (one for each cluster), we could in principle use a similar approach to produce weighted inference rules in the style of Garrette et al.. However, this approach needs to be generalised so that it can learn from arbitrary sentences, and not just from named entities.

A second option is to define a vector space with a logical interpretation. Copestake and Herbelot (2012) propose a vector space where dimensions correspond to logical expressions being true of an individual in a situation. This allows a direct connection with formal logic and model theory. However, for this approach to enjoy good logical properties, we must generalise from an **actual** distribution (based on observed utterances) to an **ideal** distribution (based on truth of logical expressions). I am not aware of work that has implemented such a system.

Grefenstette (2013) gives a logical interpretation to the type-driven tensorial framework (see §2.5.3), using the sentence space to model truth values, and the noun space to model a domain of N individuals. However, they prove quantifiers have nonlinear behaviour,[18] so cannot be expressed using tensor contraction. This restricts the class of logics compatible with the framework, ruling out standard model-theoretic semantics.

Finally, a third option is to use logical representations instead of vectors. For example, Andreas et al. (2016a,b) represent meaning as a *neural network*, rather than as an *input* to a network. These networks can be composed, following a logical form, and trained on a supervised task. However, to my knowledge, a logical approach has not been tried in distributional semantics. This thesis is an attempt to do that.

[18] Their proof assumes that universally quantifying over a predicate with empty extension should be treated as trivially true. However, the same conclusion holds for predicates with non-empty extension. For example, consider the predicates, *cat*, *dog*, and *animal*, and consider two individuals, where one is a cat and an animal (but not a dog), and the other is a dog and an animal (but not a cat). Assuming *all cats are cats* is true, and *all cats are dogs* is false, and assuming the universal quantifier is a third-order tensor, we can derive that *all cats are animals* is a superposition of true and false. This is because the predicate for *animal* is a superposition of *cat* and *dog*.

2.3.3 Context Dependence

The flipside of compositionality is **context dependence** – the meaning of an expression often seems to depend on what other words it occurs with. For example, a *small elephant* is not a *small animal*, but a *large ant* is – the interpretations of *small* and *large* depend on the nouns they modify. Even the same expression can seem to have different meanings depending on the situation in which it is used. For example, the size of a *large animal* may be quite different when buying a pet and when visiting the zoo. One goal for a semantic theory is to model how meaning depends on linguistic context and on extralinguistic context.

Following Recanati (2012), I use **standing meaning** to refer to the context-independent meaning of a linguistic expression, and **occasion meaning** to refer to the context-dependent meaning of an expression in a particular occasion of use.[19] However, it is important to note that every usage occurs in *some* context (for discussion, see: Searle, 1980; Elman, 2009), so a standing meaning must be seen as an abstraction across usages, rather than a usage in a "null" context. A speaker who knows a word knows its standing meaning, but whenever they use or hear the word, it will have an occasion meaning.

In model theory, meanings are defined in terms of extensions, but we have just seen how *small* doesn't have a fixed extension (for discussion of adjectives, see: Lahav, 1989, 1993; Blutner, 1998; McNally, 2016a). One solution is to represent such terms as functions from extensions to extensions (for example: Parsons, 1970; Kamp and Partee, 1995), so that *small* maps the set of mice to the set of small mice, and so on. Another solution is to make such terms **indexical** – that is, they are functions from a context to an occasion meaning (for example: Kaplan, 1979, 1989; Recanati, 2012). Both of these approaches avoid the problem, but as with the definition of "intension" that we saw in §2.1.2, these push the problem further down the road, since they invoke more complicated objects without detailing how to represent them. It would be implausible to represent such functions by enumerating an extension for each possible argument, so we are still left with the question of how to determine an extension given a context.

An alternative is to represent a standing meaning as a probability distribution over extensions, so that Bayesian inference gives us a concrete way to calculate an occasion meaning. Such an approach has been successfully applied to cases such as *large* and *small* (Lassiter and Goodman, 2015; Goodman and Frank, 2016). However, this has only been tested on small hand-written models, and for higher-dimensional spaces, such distributions become challenging to represent and learn.

Faced with challenging real-world examples of context dependence, McNally (2016b) concludes that traditional model-theoretic approaches leave out too much of lexical semantics, and

[19] This adapts Quine (1960), who contrasts "standing sentences" and "occasion sentences". The truth of an occasion sentence depends on some preceding "stimulus" (which we might take to be a situation), while the truth of a standing sentence does not. I am interested here in breaking apart the meaning of an occasion sentence in terms of its occasion meaning (which depends on a stimulus) and its standing meaning (which does not). If standing sentences exist, they can be seen as degenerate occasion sentences, where the occasion meaning is constant.

suggests that distributional semantics might fill the gap. As we have seen, standard distributional approaches produce a single vector for each word. One approach to treat such a vector as a standing meaning, and modify it to produce occasion meanings (for an overview, see: Dinu et al., 2012). For example, Erk and Padó (2008) and Thater et al. (2011) modify vectors according to syntactic dependencies. Erk and Padó (2010) build a context-specific vector based on the most similar contexts in a corpus. However, it is an open question how such approaches could be generalised to allow other kinds of context, including extralinguistic context.

Dinu and Lapata (2010) interpret a vector as a probability distribution over a set of latent senses, where each component is the probability of a particular sense. We can then find the contextualised meaning by conditioning this distribution on the context. A probabilistic approach can be more easily generalised to other kinds of context, because probabilistic models can be defined in a modular way, and probability theory gives us tools for combining sources of information – we could condition a distribution on both linguistic and extralinguistic context. However, Dinu and Lapata's approach ultimately relies on a finite number of senses, which we want to avoid, as discussed in §2.2.2 above.

An alternative to modifying distributional vectors is to define a probabilistic generative model, so that occasion meanings are generated based on standing meanings. For example, Lui et al. (2012)'s "per-lemma" model uses Latent Dirichlet Allocation (LDA) (Blei et al., 2003). An occasion meaning is a distribution over context words, and a standing meaning is a prior over occasion meanings. Occasion meanings vary continuously (as mixtures of "topics", each topic being a word distribution). A separate model is trained for each target word. Chang et al. (2014) add another generative layer, allowing them to train a single model across all target words. A standing meaning is a distribution over senses (particular to a target word), which are distributions over topics (common to all target words), which are distributions over context words. We first draw from the distribution over senses, then draw topics and words. However, in this model, a single sense chosen in each context, which means we have a finite sense inventory. In principle, the above approaches could be combined, so that occasion meanings are sense mixtures (allowing them to vary continuously), but topics are shared (allowing words to be trained together). I am not aware if this has been tried.

Skip-gram can also be interpreted as a generative model, generating context words based on a target word. While we can see a word vector as a standing meaning, no part of the model can be seen as an occasion meaning. Bražinskas et al. (2018) add another generative layer, first generating a latent vector based on the target word, and then generating context words based on the latent vector. We can see a latent vector as an occasion meaning, and a word's distribution over latent vectors as a standing meaning.

As well as being easier to generalise to other kinds of context, probabilistic models have one further advantage. In approaches that modify vectors, occasion and standing meanings are represented as the same type of object (a vector). Even Herbelot (2015), who explicitly con-

trasts individuals and kinds, embeds both in the same space. In contrast, probabilistic generative models represent them as different types of object. This captures the fact that a standing meaning is an abstraction across usages. In this thesis, I aim to combine the sound theoretical basis of models like Lassiter and Goodman's with the robustness of models like Bražinskas et al.'s.

2.4 Learning Meaning

In the above sections (§2.1–2.3), I considered goals for a semantic theory, in terms of how to represent meanings. One final goal is to be able to **learn** (or **train**) such representations based on observed data. We might add a further goal to make the semantic model psychologically plausible, but I will not have much to say in this thesis about how people actually learn. The focus in this section is to consider whether models expressive enough to capture semantics are can also be implemented and used in practice. Models can usually be analysed in terms of their **parameters** (or **weights**), independent values that must be set based on the data.[20] The parameters should be set so that the model can **generalise** to new data, rather than only describe the data it was trained on. Difficulties can come from the number of parameters, or from the computational cost of determining their values.

Work on model theory mostly focuses on formal properties, without proposing a machine learning model. One way to link model theory to machine learning is to explicitly define a model structure. For example, Young et al. (2014) use a set of images to stand for situations. Each image is annotated with a set of captions, which can be taken to be true for that image. They use rewrite rules to produce simplified captions, many of which are true for multiple images. They define the "visual denotation" of a caption to be the set of images it describes. To generalise to new images, it is necessary to train an image processing model (for example: Vinyals et al., 2015; Xu et al., 2015). However, this dataset only includes captions at the level of the whole image. Some datasets go further, and annotate individuals in an image (for example: Escalante et al., 2010; Lin et al., 2014), or even spatial relations between individuals (for example: Hürlimann and Bos, 2016). While such datasets are certainly useful, it would be difficult to annotate a dataset at the level of detail often seen in model theory. Furthermore, for abstract concepts, it is difficult to even say what kind of annotated dataset would be useful. For a model-theoretic machine learning model to scale to cover an entire language, I believe that model structures must be learnt, rather than explicitly annotated.

Distributional semantics is firmly rooted in data-driven approaches. Vector-based models are efficient to learn, and one way they have been made even more efficient is to share parameters between words, which reduces the total number of parameters. This is called **sharing statistical strength**. One example is singular value decomposition (SVD, often called Latent

[20] I do not mean to ignore nonparametric models – for example, HDP (Teh et al., 2006) is a nonparametric version of LDA, where the number of topics is not fixed. However, a technical discussion of nonparametric models would be out of place here. For an introduction, see: Gershman and Blei, 2012.

Semantic Analysis when applied to co-occurrence vectors: Deerwester et al., 1990; Landauer and Dumais, 1997) – latent dimensions are shared across the vocabulary, and word meanings are represented in terms of a small number of latent dimensions, rather than a large number of context dimensions. As mentioned in §1.2.1, Skip-gram can also be seen as performing dimensionality reduction.

However, if we move beyond vector space models, learning may become more difficult. Representing meaning as a higher-order tensor is challenging, because of the number of parameters. For an N-dimensional vector space, a K-order tensor has N^K parameters. This is particularly relevant for the type-driven tensorial framework (see §2.5.3). For some function words that would seem to need high-order tensors, it is possible to give analyses so that tensors do not need to be learnt – for example, Sadrzadeh et al. (2013) analyse relative pronouns using Frobenius algebras. However, this approach cannot be applied to content words. For example, ditransitive verbs like *give* are represented as fourth-order tensors. There has been some work towards alleviating this problem, such as forcing tensors to be of low rank (Fried et al., 2015), using a matrix for each argument separately (Paperno et al., 2014), or using a matrix applied to the elementwise product of the arguments (Polajnar et al., 2014a). Using low-rank tensors introduces a tradeoff between expressiveness and dimensionality, while using matrices loses interactions between arguments.

Work on representing meaning as a probability distribution generally constrains the family of distributions, in order to reduce the number of parameters and allow efficient learning. Vilnis and McCallum (2015), Barkan (2017), and Bražinskas et al. (2018) use Gaussian distributions (see §2.2.1 and §2.3.3), but restrict the covariance matrices to be diagonal. This means that, for an N-dimensional space, each distribution can be represented by $2N$ parameters (N for the mean, and N for the diagonal covariances). Using a larger family of distributions allows a more expressive model, but makes learning more challening. Athiwaratkun and Wilson (2017) use a Gaussian mixture model, to capture word senses. For K senses, each with diagonal covariances, the number of parameters scales as NK, which is still manageable. In principle, we can push the limits of these models by considering increasingly flexible families of distributions.

Expressive probabilistic models often include unobserved random variables associated with each data point. For example, topic models like LDA associate a topic with each token. Bražinskas et al.'s model associates a context vector for each usage of a target word. These are called **latent variables**, and they make learning more challenging, since exact calculations of probabilities require summing over all possible values for each latent variable. In practice, approximate techniques must be used, such as Gibbs sampling or variational inference, which will be the topics of §5.3 and §5.4. One important benefit of such models is that they allow words to share statistical strength, because many words can use latent variables with the same value. Wang et al. (2017) show that an LDA model provides a promising approach to learning a distributional representation from a single example (**one-shot** learning).

Representing meaning as a region has not been much explored in distributional semantics. As discussed in §2.1.2, Erk (2009a,b) learns regions, but based on pre-trained word vectors. In principle, it would be possible to directly learn regions, in a similar way to the above approaches that learn distributions. As with distributions, there would be a similar tradeoff between expressiveness and number of parameters.

Finally, there are a few options discussed in the previous sections which have not been applied to distributional semantics. Representing meaning using a probabilistic logic has been proposed (see §2.3.2 and §2.5.4), but existing work has focused on hand-written models in small domains. Representing meaning as a classifier has been proposed and implemented (see §2.1.2), but not for learning distributional representations. Representing meaning as an ideal distribution of true propositions has been proposed (see §2.3.2), but not implemented. For all three of the above approaches, learning such representations from distributional data is a challenge, because much of the structure is latent – logical structure in the first case, referents in the second case, and truth values in the third. This thesis takes a step towards making such learning feasible.

2.5 Existing Frameworks

In the final section of this chapter, I will look at a few general approaches to semantics, in the light of the goals discussed above. After all, if there already is a framework that does what we need, this thesis would not be necessary.

2.5.1 Extensions of Vector Space Models

In every section above, we saw examples of vector space models being extended to deal with different goals. A sensible question is then, can't we just combine these extensions?

The trouble is that these various extensions take vector space models in different directions, and it's often not at all clear how to combine them. Introducing one kind of structure to a vector space model can be incompatible with another kind of structure. For example, one proposal for hyponymy is to use a distribution over a vector space (see §2.2.3), and one proposal for compositionality is to use a recurrent neural net (see §2.3.1). How should an RNN process a sequence of distributions? One option would be to sample a vector for each token, run the RNN over these vectors, and repeat this for a number of samples. Another option would be to input the parameters of the word distributions directly to the RNN. I am not aware of work exploring either of these options (or any other).

I should be clear – I am not arguing that it's impossible to combine the various approaches discussed in this chapter. However, doing this is not at all obvious, and I believe that the burden of proof is on showing that it *can* be done, rather than showing that it *can't* be done. In this thesis, I have tried to bring together various lines of thought into one coherent model. Alternative proposals would be welcome.

2.5.2 Hybrid Approaches

Faced with the above difficulty of combining different types of model, an alternative response is to keep different parts distinct, and build a hybrid model. We saw this most clearly in §2.3.2, combining a distributional approach with a logical approach. Lewis and Steedman (2013) use distributional information to cluster predicates, and then perform classical logic on clustered predicates. Beltagy et al. (2016) use distributional vectors to calculate similarity, and then add weighted inference rules to a Markov Logic Network. In both cases, the distributional component of the model can be calculated on its own, and then fed into a logical system.

However, the components of a hybrid system are not completely distinct. Even if we can calculate the distributional component on its own, we can still ask what contribution it makes to the rest of the system, and whether we could improve this contribution. Erk (2016) examines what distributional vectors bring to Beltagy et al.'s system, and argues that distributional similarity (for a suitably tuned vector space model) gives us similarity in terms of property overlap between individuals. For example, if we know that the vectors for *alligator* and *crocodile* are similar, then the individuals in their extensions are likely to share many properties. If property overlap is what we hope to get from distributional semantics, then it seems sensible to aim for this directly, rather than trying to find which kind of co-occurrence it correlates best with.

The downside of directly aiming for richer representations (such as representations that include properties of individuals) is that they can be harder to learn, as discussed in §2.4. In this thesis, I take a step towards making this feasible.

2.5.3 The Type-Driven Tensorial Framework

Coecke et al. (2010) and Baroni et al. (2014) introduce a framework for compositional distributional semantics, which I have referred to as the type-driven tensorial framework (for a short introduction with simple notation, see: Maillard et al., 2014). As mentioned in the introduction, in §1.2.2, this framework makes use of the algebraic notion of a vector space, and draws techniques from linear algebra. Each lexical item has a semantic type, and a corresponding syntactic type in a Categorial Grammar (for an introduction, see: Steedman and Baldridge, 2011). For the semantic types, nouns are represented in one vector space, and sentences are represented in a second vector space. Other types are represented as higher-order tensors, where tensor contraction (a generalisation of matrix multiplication) corresponds to argument structure. For example, an intransitive verb is a second-order tensor (a matrix), mapping a noun to a sentence; a transitive verb is a third-order tensor, mapping two nouns to a sentence; an adjective is a second-order tensor (a matrix), mapping a noun to a noun; and so on.

This framework is an impressive attempt to build a practical distributional model that keeps long-term top-down goals in mind, and as a result, I have discussed it in most of the previous sections. The missing sections are §2.1, on how meanings relate to the world (a challenge for

all distributional approaches), and §2.3.3, on context dependence (there is a mechanism for composing representations, but this does not use information from any surrounding context).

Clark et al. (2016) give an overview of developments, and highlight three key challenges – how to choose a sentence space, how to learn high-order tensors, and how to represent closed-class words. I believe all three of these are serious. I already argued against sentence vectors in §2.3.1. As for learning high-order tensors, not only is this already a challenge, but several of the suggested extensions propose using mixed states, which would double the order of the tensor, and hence square the number of parameters – for a ditransitive verb like *give*, such a mixed state would require an 8th-order tensor. Finally, for closed class words, we saw in §2.3.2 that quantifiers seem to have nonlinear behaviour. Hedges and Sadrzadeh (2017) provide an alternative account which can deal with quantifiers, but this comes at the expense of using a vector space whose dimensions correspond to *sets* of individuals, so we have 2^N dimensions for a model structure containing N individuals.

Krishnamurthy and Mitchell (2013) sketch how a type-driven approach could use operations beyond tensor contraction, but I am not aware of this work being followed up. Successors to the type-driven tensorial framework may have to move away from tensors, and the resulting framework may end up looking quite different. Semantics is sadly too nonlinear.

2.5.4 Probabilistic Semantics

A semantic theory that uses probabilistic logic would seem able to meet the challenges of both lexical semantics (see §2.2) and sentence-level semantics (see §2.3), and there are already existing frameworks for probabilistic semantics.

Goodman and Lassiter (2015) use probabilistic programs to represent both world knowledge (a program can generate a distribution over situations) and linguistic knowledge (a program can generate truth values for utterances in given situations). They present several hand-written programs to show how the two can interact. Each word's meaning is represented by a program, and these can be composed to produce a representation of a sentence. This is a powerful framework, but without constraints on what programs are allowed, learning such a model distributionally would be challenging.

Cooper et al. (2015) introduce a probabilistic version of Type Theory with Records (TTR) (Cooper, 2005). In this framework, the basic notions are situations and types. In classical TTR, situations are either instances of a type or not, while in probabilistic TTR, judgements of a situation being of a certain type are made probabilistically. Cooper et al. show how probabilistic types for complex expressions can be constructed compositionally, and they present a hand-written grammar in this framework. Although they do discuss learning, they assume access to a much richer input than simply corpus data.

While I agree with much of what is presented in both of these frameworks, they do not provide a clear starting point for distributional semantics.

Chapter 3

Formal Framework of Functional Distributional Semantics

In this chapter, I show how model theory can be recast in a probabilistic setting. The aim is to define a family of probability distributions that capture classical model structures as a special case, while also allowing structured representations of the kind used in machine learning. I use this probabilistic generalisation of model theory to define a probabilistic graphical model for distributional semantics, where semantic dependency graphs provide an important link between formal semantics and machine learning. The framework developed in this chapter forms a basis for the rest of the thesis.

3.1 Summary of Classical Model Theory

Before presenting my framework, it will be helpful to summarise **classical** (non-probabilistic) model theory, as presented in §1.3, as well as the challenges discussed in Chapter 2. A model structure represents a situation. It contains individuals (including event individuals), as well as semantic roles (ARG1, ARG2, ARG3, ARG4) from one individual to another.[1]

The meanings of content words (concepts) are represented as predicates. Each predicate takes a truth value (truth or falsehood) for each individual. A predicate can be formalised either as a truth-conditional function (a mapping from individuals to truth values) or as an extension (the set of individuals for which the predicate is true).

[1] By making semantic roles part of a situation, I make the simplifying assumption that the structure of a situation is isomorphic to the structure of a semantic dependency graph. In many cases, this is unproblematic, and the ARG1 and ARG2 roles can be seen as roughly corresponding to Dowty (1991)'s notions of "proto-agent" and "proto-patient". However, in the general case, the assumption might not hold. For example, if we compare *Mary sold a book to John*, and *John bought a book from Mary*, we have two descriptions of the same situation, but with different role labels: for *sell*, Mary is the ARG1 and John the ARG3; while for *buy*, John is the ARG1 and Mary the ARG3. A more accurate theory would need to distinguish situation structure from semantic dependencies, but for distributional semantics, the assumption of isomorphism is good enough.

The meanings of sentences are represented as logical propositions, which take a truth value for each situation. Each predicate in a proposition has a unique intrinsic argument, which is a variable ranging over individuals. Each variable must be quantified, so that the proposition can take a truth value. Such propositions can be represented as semantic dependency graphs (DMRS graphs).

As we saw in the last chapter, the strengths of this approach are in logic and compositionality. The main challenge that we need to address is how to determine truth values in a new situation. Enumerating individuals does not generalise, and the classical theory of concepts has empirical problems. Furthermore, this should be done in a way that allows a natural account of vagueness and polysemy, and allows representations to be learnt from observed data.

3.2 Individuals and Pixies

If individuals are atomic elements, without any further structure, then extensions and truth-conditional functions are almost identical. An extension is a subset of the individuals in the model structure, while a truth-conditional function is the indicator function for this subset: individuals in the extension are mapped to 1, and other individuals to 0. Converting between these two representations is trivial. A simple example is given in Fig. 3.1, where there are several individuals (and no semantic roles, for simplicity).[2] The extension for the predicate for *pepper* is a subset of the individuals, as indicated in Fig. 3.2. Its truth-conditional function maps these individuals to truth, and maps the other individuals to falsehood.

However, if individuals are structured objects, extensions and truth-conditional functions

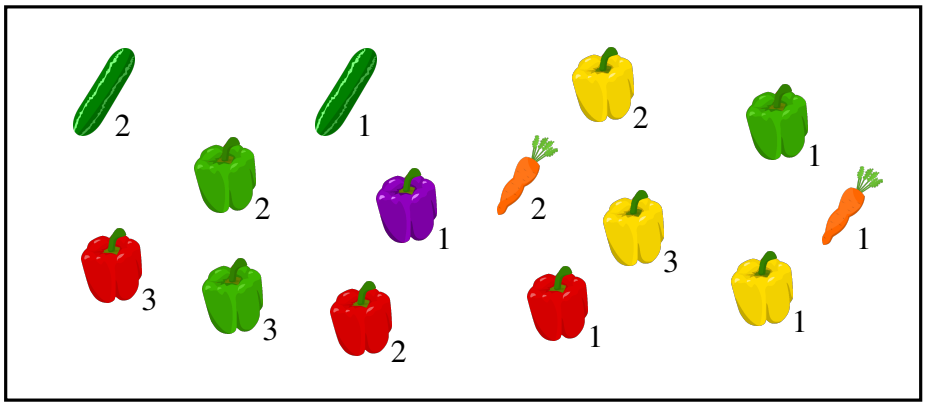


Figure 3.1: A simple model structure, with 14 individuals and no semantic roles. Subscripts are used to distinguish individuals with identical features, but are otherwise not meaningful. The position of each individual is arbitrary.

[2] The images used in figures in this chapter are modified versions of images available on Openclipart on a Creative Commons Zero 1.0 Public Domain License:
https://openclipart.org/detail/229817/bell-pepper,
https://openclipart.org/detail/229814/cucumber,
https://openclipart.org/detail/229825/carrot

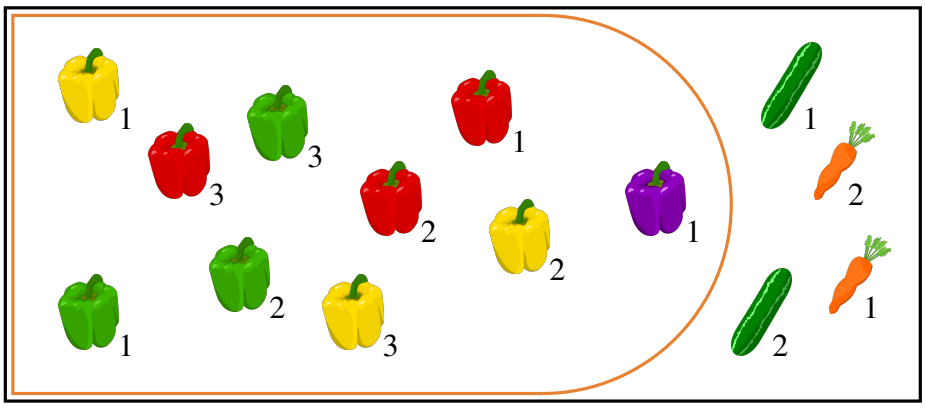


Figure 3.2: The orange line indicates the extension of the *pepper* predicate, for the individuals in Fig. 3.1. As the position of each individual is arbitrary, they have been re-arranged for clarity for this predicate.

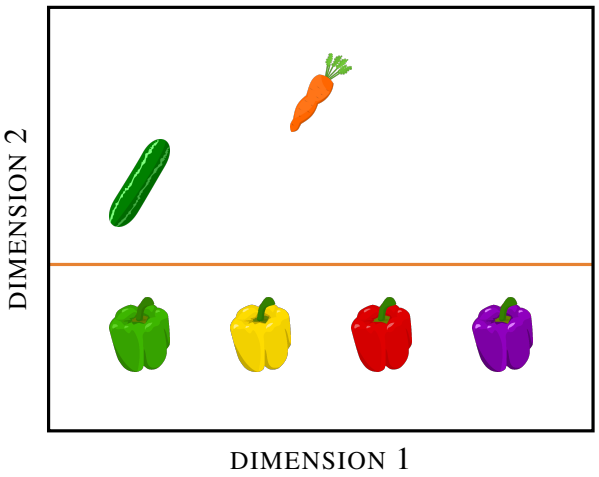


Figure 3.3: A simple two-dimensional semantic space. Pixies that correspond to at least one individual in Fig. 3.2 are indicated. Unlike Fig. 3.2, the position of each pixie in this space indicates its features. The orange line is the decision boundary for the *pepper* predicate.

may have rather different representations. To represent the structure of individuals, I will use a **semantic space**,[3] where each point in the space represents a possible individual, including information about its features. I will use the term **pixie** to refer to a point in the semantic space, as it is intuitively a "pixel" of the space.[4] Each individual has a pixie representation. Two individuals which have exactly the same features will correspond to the same pixie. Conversely, many pixies may not correspond to any individual. It is important to note that a semantic space can be defined at different levels of granularity – a coarse-grained space (with few dimensions) will force more individuals to have the same or similar pixies, while a fine-grained space (with more dimenions) will allow more individuals to be distinguished. While more fine-grained spaces are more expressive, they may also make learning more challenging.

[3] Gärdenfors (2000, 2014) uses the term "conceptual space".
[4] In Emerson and Copestake 2016, the term "entity" was overloaded to refer to both individuals and pixies.

An example is given in Fig. 3.3, which embeds the individuals of Fig. 3.2 into a two-dimensional space. Except for the purple pepper individual, each individual shares a pixie with at least one other. Conversely, much of the space (such as the top right corner) does not correspond to any individuals. Each dimension can be thought of as a **feature**, and in this space, dimensions 1 and 2 roughly correspond to colour and shape, respectively. However, I should stress that, in the general case, the dimensions of a semantic space do not need to correspond to any natural language expressions. We will use a semantic space to define the meaning of predicates, and not the other way round.

To represent the *pepper* predicate as an extension, we need to enumerate the set of ten individuals shown in Fig. 3.2. However, as a truth-conditional function, it can be represented much more simply, by referring to the semantic space, as shown in Fig. 3.3 – it is true given a low value of dimension 2, and false given a high value of dimension 2. Not only is this a more succinct representation, but it also generalises to new situations, because it refers to pixies, rather than individuals. As long as new individuals can be embedded in the semantic space, we can apply this truth-conditional function.

3.3 Probabilistic Model Structures

The above discussion assumed complete knowledge, both of the situation, and of the truth values. Each of these assumptions can be relaxed, using probability theory to represent uncertainty.

As discussed in §2.2.1, predicates are often vague, with uncertain truth values. For example, perhaps we shouldn't be so bold as to assume that the purple vegetable in Fig. 3.2 would definitely be classified as a pepper. For a speaker who has never seen or heard of such a thing, they might not be certain how to classify it. For instance, what if it is actually an unrelated species? This is illustrated in Fig. 3.4, where the extension of the predicate is uncertain. We can capture this by representing the *pepper* predicate as a probabilistic truth-conditional function – rather

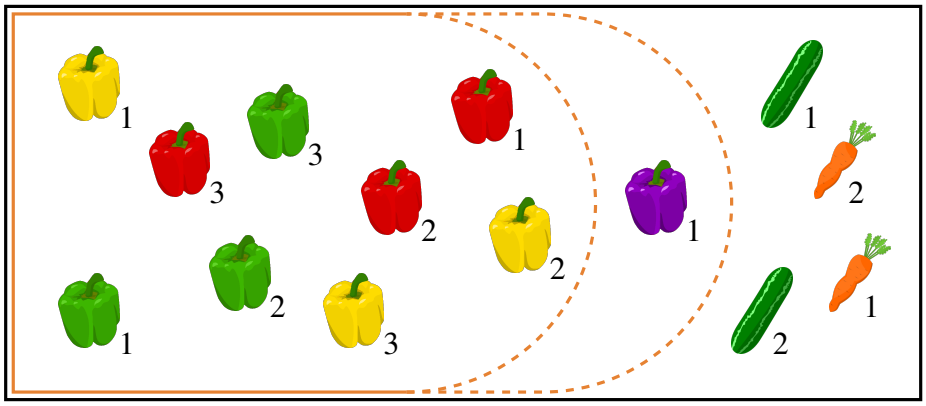


Figure 3.4: A vague model structure. The *pepper* predicate has an uncertain truth value for the purple individual.

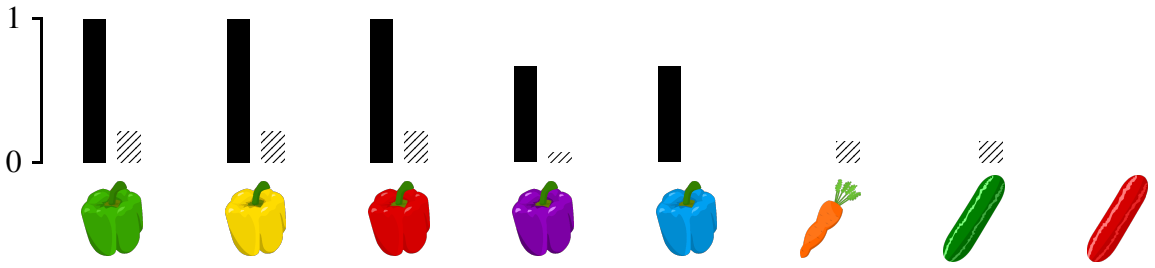


Figure 3.5: A probabilistic version of the model structure in Fig. 3.4. For simplicity, only eight pixies from the semantic space are shown.
Solid bars: the truth-conditional function for *pepper* is replaced by a function from pixies to probabilities of truth. Pepper pixies with atypical colours have intermediate probabilities.
Shaded bars: the set of individuals is replaced by a distribution over pixies. Pixies corresponding to more individuals have higher probabilities. The blue pepper pixie and red cucumber pixie correspond to no individuals, so have probabilities of 0.

than mapping each pixie to a certain truth value, it maps each pixie to a value in the range $[0, 1]$, representing the probability of truth. This is illustrated by the solid bars in Fig. 3.5. As well as the six pixies that correspond to individuals in Fig. 3.4, two additional pixies are shown, to illustrate how truth values can be assigned to new individuals.

As well as being uncertain about truth values, a speaker may also be uncertain about the situation.[5] To formalise this under a classical model-theoretic approach, we could consider a set of possible situations. However, some possible situations could be much more likely than others, so it is natural to consider a distribution over situations, rather than simply a set. It would be difficult to depict a distribution over situations of a similar size to the one in Fig. 3.2, so for ease of illustration, we can consider simpler situations, consisting of single individuals. In particular, we can consider the 14 sub-situations of Fig. 3.2 containing a single individual. A distribution over these situations is illustrated by the shaded bars in Fig. 3.5.[6]

From the formal linguistic point of view, a distribution over situations might seem irrelevant to the notion of truth. However, as argued in §2.4, one goal for a semantic theory is for it to be learnable. When learning a semantic model, we do not know in advance what situations should be included. Intuitively, as a learner discovers what kinds of situations exist, they update their semantic model appropriately. For common pixies, it is important to get classifications right, but for rare pixies, it doesn't matter much. Having a distribution over situations allows a learner to distinguish situations which are important from those which are not.

[5] This could be uncertainty about the details of a particular situation, or uncertainty about a situation randomly sampled from the world. Probability theory can encompass any level of uncertainty, and a hierarchical model can be used distinguish them, if desired (for discussion, see: Lassiter, 2017).

[6] As with the distinction between individuals and pixies, we can distinguish between situations as collections of individuals and as collections of pixies. Each individual-situation has a corresponding pixie-situation, and a pixie-situation is an equivalence class of individual-situations for a given semantic space. The distribution in Fig. 3.5 is over pixie-situations, with higher values for those corresponding to more individual-situations.

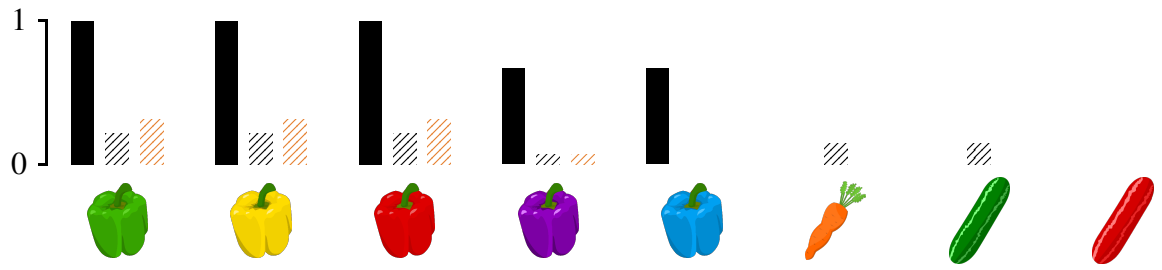


Figure 3.6: An example of Bayesian inference. We are interested in: an individual X, represented by some pixie $x \in \mathcal{X}$; and the truth value T of the *pepper* predicate for that individual.
Solid bars: the semantic function $\mathbb{P}(T = \top \mid X = x)$ represents a speaker's belief about whether each pixie x can be considered to be a pepper.
Black shaded bars: the prior $\mathbb{P}(X = x)$ represents the speaker's belief about an individual, based on their world knowledge. It encodes how much they expect to observe an individual with particular features.
Orange shaded bars: the posterior $\mathbb{P}(X = x \mid T = \top)$ represents their belief about an individual, if they know the *pepper* predicate is true of it. The probability mass is split between the pepper pixies, but skewed towards typical colours, and excluding colours believed impossible.

3.4 Semantic Functions

I will refer to a probabilistic truth-conditional function as a **semantic function**. In machine learning terms, we can also view such a function as a binary classifier, as discussed in §2.1.2. For example, the semantic function for *pepper* is a classifier separating pepper pixies from non-pepper pixies. In this section, I will discuss benefits of this approach.

To formalise the discussion in the previous section, we can consider a joint distribution over situations and truth values. To begin with, we can consider the simple situations shown in Fig. 3.5, which only contain a single individual. So, we have two random variables: a variable X, representing the features of the individual, whose value is a pixie x in the semantic space $\mathcal{X}$; and a variable T, representing the truth value for the *pepper* predicate for that individual, whose value is either $\top$ (truth) or $\bot$ (falsehood). A semantic function t is a *conditional* distribution over truth values, *given a pixie*. We can write this as shown in (3.1). Note that, while the semantic function takes values in a range, there are still only two truth values (truth and falsehood). This is in common with probability logic, rather than fuzzy logic.

$$t(x) = \mathbb{P}(T = \top \mid X = x) \tag{3.1}$$

While a semantic function does not represent a distribution over pixies, it can be used to define one, if we have a prior distribution $\mathbb{P}(X = x)$. We can then apply Bayes' Rule, as shown in (3.2). In contrast with existing work that represents lexical meaning as a probability distribution (see §2.2.1), this means that we can combine a semantic function with domain knowledge or contextual knowledge encoded in the prior $\mathbb{P}(X = x)$. If lexical meaning is

represented as a distribution, other kinds of knowledge cannot be combined in such a natural way.

$$\mathbb{P}\left(X=x \mid T=\top\right) \propto \mathbb{P}\left(T=\top \mid X=x\right) \mathbb{P}(X=x) \tag{3.2}$$

Fig. 3.6 gives an example of Bayesian inference. The division between $\mathbb{P}(X = x)$ and $\mathbb{P}\left(T=\top \mid X=x\right)$ can be seen as a division between world knowledge and conceptual knowledge. The two are not completely independent, since there is a joint distribution over the two random variables, but this can still be a useful distinction. As purple and blue are atypical colours for a pepper, a speaker might be less willing to label such a vegetable a pepper, but not completely unwilling to do so – this conceptual knowledge belongs to the semantic function for the predicate. In contrast, after observing a large number of peppers, we might conclude that blue peppers do not exist, and purple peppers are rare, while green, yellow, and red peppers are common – this world knowledge belongs to the probability distribution over pixies.

This example also shows a clear contrast between a semantic function on a space and a distribution over the space. As peppers come in many colours, the semantic function $\mathbb{P}\left(T=\top \mid X=x\right)$ should take a high value for any of these colours. In contrast, to define a probability distribution $\mathbb{P}\left(X=x \mid T=\top\right)$ over pepper pixies, we must split probability mass[7] between different colours, which means we only have a small probability of each. The value of a distribution at a point depends on the size of the space, but the value of a semantic function does not. The two represent different things. A distribution over a semantic space represents uncertainty about the features of an individual, for which there is a correct but unknown answer. A semantic function represents an underspecified concept, where many individuals with different features could be equally regarded as instances of the concept.

Representing lexical meaning as a semantic function rather than a distribution places an emphasis on classification, rather than generation. This would suggest that a speaker might be able to classify instances of a concept without being able to generate instances. Rather than speculating about blue peppers, we can also turn to empirical evidence that people learn to classify without learning to generate (for example: Nickerson and Adams, 1979; Jones, 1990; Lawson, 2006; Blake et al., 2015). In a striking set of experiments, Wong et al. (2018) investigated knowledge of two forms of lowercase *g*: the looptail ⟨g⟩, and the opentail ⟨ɡ⟩. Both forms are common in printed English, so we would expect English speakers to have acquired both. However, only opentail ⟨ɡ⟩ is common in handwriting. In Wong et al.'s first experiment, 38 participants were interviewed about lowercase letter forms, but only 1 could write looptail ⟨g⟩ correctly. In their second experiment, 16 participants were given a text written with looptail ⟨g⟩, and asked to find all words containing the letter *g*. The text was then removed, and the participants were asked to write the letter form they had just read. Half wrote an open-

[7] In fact, as colours lie in a continuous space, a distribution over pixies would be better represented with a probability density function rather than with a probability mass function. The value of a probability density function at a point would further depend on the parametrisation of the space, while the value of a semantic function would not. Fig. 3.6 uses a finite number of pixies, rather than a continuous space, for ease of illustration.

Figure 3.7: Wong et al. (2018) presented participants with four letter forms resembling looptail ⟨g⟩, and asked them to choose the correct one. The proportion of participants choosing each variant is shown underneath it. The first variant is the correct looptail ⟨g⟩.

tail ⟨g⟩, and only 1 wrote looptail ⟨g⟩ correctly. In their final experiment, 44 participants were asked to identify the correct form of looptail ⟨g⟩ from a set of distractors, as shown in Fig. 3.7. However, only 28% were correct, and 56% made the same wrong decision, choosing the letter form with the tail reversed.

Wong et al.'s results illustrate the difference between classification and generation. Despite a perfect ability to classify looptail ⟨g⟩ when reading, participants struggled to generate it when writing. The results of the third experiment are particularly telling, because most participants preferred the tail-reversed form, despite it not existing in real text. However, we can explain this in terms of a distribution over pixies (letter forms) and a semantic function for the letter *g*. Three of the forms are previously unseen pixies, but all four forms have relevant features for the letter *g*. The most popular form has its "ear" at the top right, evoking looptail ⟨g⟩, while its reversed tail has a connecting stroke on the right, evoking opentail ⟨g⟩. To be proficient at reading, it is sufficient to classify observed letter forms correctly (including classifying looptail ⟨g⟩ as *g*), but it is not necessary to learn an accurate distribution over letter forms. In other words, it is sufficient to learn semantic functions which are accurate on observed pixies, but it is not necessary to learn an accurate distribution over pixies. While a speaker may learn to generate some concepts, it is simplest to model a concept just as a semantic function – this can be represented more simply than a distribution over pixies, it can account for the above experimental results, and it gives us a direct connection with logic and formal semantics.

In the following two subsections, I will examine semantic functions in more detail, first contrasting them with regions of space, and then considering what kind of uncertainty the probabilities represent. A reader who is comfortable with the above definition of semantic functions, and who would like to see how they can be used for distributional semantics, can safely skip these subsections, jumping straight to §3.5.

3.4.1 Regions of Semantic Space

In §2.1, I concluded that two sensible ways to represent a concept are as a region of space and as a binary classifier. In this section, I will explain how they are two views of the same thing. The additional detail that can now be added, based on the discussion in this chapter, is that we are considering a *semantic space*, where each point is a *pixie* (rather than an individual).

For any region of space, we can define a non-probabilistic binary classifier, which classifies all pixies in the region as instances of the concept, and all pixies outside the region as not. Conversely, for any non-probabilistic binary classifier,[8] we can define a region of space composed of all pixies that are classified as instances of the concept. For example, a support vector machine (SVM) is a non-probabilistic classifier – on one side of the decision boundary (the "inside"), pixies are classified as instances of the concept, while on the other side (the "outside"), pixies are classified as not.

Once we introduce uncertainty, the correspondence becomes subtler, because a distribution over regions is more general than a semantic function (a probabilistic binary classifier). Intuitively, knowing about regions of space is *global* information, but knowing about truth values for particular pixies is *local* information. Given a distribution over regions, knowing about truth values in one part of the space can tell us about truth values in another distant part of the space.

More precisely, given a pixie-valued random variable X, and a region-valued random variable A, we can define a binary random variable T to be true when $X \in A$ and false otherwise. This means that we can define a semantic function based on a distribution over regions, by summing (or integrating) over the distribution: $t(x) = \sum_{a \ni x} \mathbb{P}(A = a)$.

Conversely, if we know the conditional distribution for T given X, this does not tell us the distribution for A. This is easiest to see if we consider a separate truth value T_x for each pixie x. A semantic function t fixes the *marginal* distributions of these random variables as $\mathbb{P}(T_x = \top) = t(x)$, but it does not fix their *joint* distribution. If we know the joint distribution, then we can define the region to be the pixies whose truth value is true: $A = \{x \in \mathcal{X} \mid T_x = \top\}$. Given a semantic function, there are many ways that we could define a joint distribution, and hence a distribution over regions. One extreme would be to sample the truth values independently for each pixie. The other extreme would be to sample a threshold value K uniformly from $[0, 1]$, and choose truth when the semantic function's value is above the threshold. This would give the region $A = \{x \in \mathcal{X} \mid f(x) > K\}$.

These two options introduce different amounts of covariance between different pixies. The first extreme (independently sampling truth values) has no covariance at all – indeed, we may not even want to use the term "region", as it would almost certainly not have a clear boundary. The other extreme (sampling a threshold) maximises covariance – for any pair of pixies, their truth values have the maximum probability of agreement. By allowing different levels of covariance between these two extremes, we can define different distributions over regions.[9] It may be desirable to have high covariance between nearby points, but low covariance between distant points. For example, it would seem incoherent to classify one reddish-orange shade as red, but a slightly redder shade as not red. However, a classification of a reddish-orange shade

[8] I assume that the classifier is a constant function except at the decision boundary.

[9] This is reminiscent of Gaussian processes (for an exposition, see: Rasmussen and Williams, 2006), which can be defined in terms of a mean function and covariance function. A semantic function can be seen as a mean function, which additionally needs a covariance function in order to define a distribution over regions.

would not seem to affect a classification of a reddish-purple shade.

Parametrising a distribution over regions in terms of a semantic function plus covariance would provide a natural way to capture this kind of local coherence of truth values. Furthermore, it also provides a more efficient parametrisation than directly trying to represent a distribution over regions – a semantic function is a map from an N-dimensional semantic space to the interval $[0, 1]$, but a distribution over regions (represented by their boundaries) is a map from the $(N-1)$-dimensional sphere to the N-dimensional semantic space, which is a much harder function to represent and to learn. The covariance function is more complicated than a semantic function, being a map from a *pair* of pixies, rather than a single pixie, but it could plausibly be hard-coded (exploiting the geometry of the space), or shared across predicates.

Finally, it is worth considering how constraints on regions look, when converted to into constraints on semantic functions. As discussed in §2.1.2, Gärdenfors (2000, 2014) proposes representing meanings as *convex* regions. We could convert this into a constraint on semantic functions by saying that for any threshold k, the semantic function defines a convex region (following the high-covariance distribution over regions given above). Many simple kinds of probabilistic classifier would satisfy this constraint – for example, one layer neural networks (including sigmoid functions and radial basis functions) define convex regions.

However, we can question whether concepts should be represented as convex regions. If convexity is defined on a perceptual space, then the claim is empirically false – even in the low-dimensional space of colour, McMahan and Stone (2015) find expressions like *greenish* whose meanings are nonconvex, and Kay et al. (1997) similarly describe how some languages have terms which could be glossed as "peripheral red". Sidestepping such objections, Gärdenfors (2014, §2.5) describes how "higher-level" dimensions can be defined in terms of more basic ones, and suggests that the convexity requirement can apply in an abstract space defined by such higher-level dimensions. A concept that is non-convex in a perceptual space might be convex in some abstract space. However, without some constraint on how we can define abstract dimensions, this obliterates the convexity requirement, because for any region, we can define an abstract space in which that region becomes convex. So, the convexity requirement also requires a constraint on defining abstract dimensions. This becomes quite natural when viewing the process as classification – defining a set of abstract dimensions looks the same as defining a layer in a neural network, where each unit in the network is one dimension. The final layer of the network is convex (as already discussed), and constraining the definition of abstract dimensions reduces to constraining the layers of the network. Dimensions that are used by many functions ("domains" in Gärdenfors's terminology) can be seen as parameter sharing.

To summarise, regions of space and binary classifiers are equivalent, where uncertainty in a region corresponds to covariance in classifications of pixies. This equivalence additionally allows us to recast the convexity requirement in terms of constraints on neural network models.

3.4.2 Uncertainty

Having introduced probabilities of truth, it is natural to ask how these probabilities are supposed to be interpreted. Bayesian probability theory uses probability to represent beliefs about the world, where there is some correct but unknown answer. For example, if a physicist is uncertain about the mass of an electron, they can perform a suitable experiment, and after observing the results, they would reduce their uncertainty. However, when it comes to linguistic or conceptual knowledge, it's not clear this is the same kind of uncertainty. If a learner is uncertain about the meaning of a predicate, what experiment could they perform? Is there a correct meaning for the learner to infer?

One influential non-probabilistic account of vagueness, called supervaluationism, holds that a predicate does have a correct set of truth conditions, but these truth conditions are unknown (for example: Fine, 1975; Keefe, 2000, chapter 7). However, simply using a *set* of boundaries leads to the problem of higher-order vagueness – if a predicate specifies a set of precise boundaries, we can reasonably ask where the boundaries begin and end. The set of boundaries therefore has its own boundary, and this higher-order boundary may also need to be vague. Intuitively, some boundaries are more plausible than others. While there might not be a clear boundary between red and orange, as we go further towards orange, we would be increasingly certain that we've crossed the boundary. Following this intuition, we can replace a *set* of boundaries by a *distribution* over boundaries, which avoids the problem of higher-order vagueness, as demonstrated by Lassiter (2011). Representing a predicate as a distribution over regions can be seen as an improvement on supervaluationism that maintains its core insight.

Given such a distribution, it would be tempting to say that a learner could ask fluent speakers for truth value judgements, and thereby reduce their uncertainty. However, for a continuous semantic space, it would require infinitely many observations to precisely determine the region in which the predicate is true. Even a finite space could require too many observations, if it is sufficiently high-dimensional. So if precise truth conditions are impossible to learn, what does linguistic knowledge actually consist of? Barwise and Perry (1983, pp. 16–19) suggest that linguistic meaning is a relation between utterance events and aspects of the world, determined by the way language is used by a speech community. In other words, meaning comes from a convention. It is arbitrary, in the sense of de Saussure (1916), but agreed on by the community (for further discussion of convention, see: Millikan, 1998). There can be variation between different speakers in a speech community, but even in the ideal case where speakers never contradict one another, a linguistic convention is necessarily vague, because a precise convention would not be learnable.

Motivated by these concerns of communication and learning, Sutton (2013)[10] argues that the meaning of a predicate is a probabilistic correlation between uses of the predicate and sit-

[10] See §4.3 for a discussion of correlation, §5.2–5.4 for learning, and §6.3 for a situation-theoretic formalisation.

uations in the world. It expresses the probability that a speaker would classify an individual as an instance of the predicate. Vagueness is not uncertainty about the world (the world separate from the linguistic acts themselves), but uncertainty about how to generalise a linguistic convention to a new situation. This view of linguistic uncertainty can be contrasted with uncertainty about precise truth conditions, where precise boundaries exist, but are unknown. However, as discussed above, precise boundaries are not learnable.

Mathematically speaking, this distinction might seem to be splitting hairs. Uncertainty about precise truth conditions can be represented as a distribution over regions of space – and as I demonstrated in the previous section, this is equivalent to a semantic function with covariance. Indeed, Lassiter and Goodman (2015) use a probabilistic framework to model pragmatic language understanding, without committing to a particular interpretation. The important point is that a distribution over truth values can be taken as a starting point for communication. A speaker's decision about what to say and a listener's inference about a situation are based on the distribution as a whole (and how it interacts with knowledge of the situation). Truth values are important stepping stones for the process of communication, but the speaker's ultimate aim is to update the listener's beliefs about the situation. Sutton (2017) argues that separating communicative success (which is external to semantics, but builds on it) from probabilistic classification (which is internal to semantics) avoids the apparent paradoxes of vagueness.

To summarise the above discussion, probabilities of truth can be seen as subjective beliefs, but rather than expressing expectations about the state of the world, they express expectations that a linguistic convention could be used in a certain way. These beliefs are uncertain, firstly because of variation in the linguistic convention, and secondly because of the need to generalise to new situations. While the first source of uncertainty could be removed in a hypothetical homogeneous speech community that strives to have a precise convention, the second source of uncertainty is inherent to learning any language that describes a world large enough to require generalisation.

3.5 A Probabilistic Graphical Model for Probabilistic Model Structures

In this section, I define a probabilistic model which generates situations and truth values, following the probabilistic generalisation of model theory presented in §3.3. In particular, I define a **probabilistic graphical model**, which is a succinct way of stating independence assumptions between random variables. A graphical model does not specify a joint distribution over the random variables, but rather gives constraints on a distribution. A graphical model consists of a graph, where each **node** is a random variable, and where **edges** indicate probabilistic dependence. An **undirected edge** means that the two nodes are dependent on each other, and a **directed edge** means that the child node (the start of the edge) is conditionally dependent on

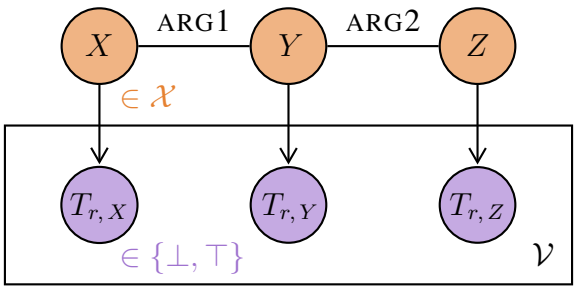


Figure 3.8: A situation containing three individuals, each represented by a pixie.
Top row: random variables X, Y, Z, whose values are pixies in a semantic space $\mathcal{X}$, jointly distributed according to the semantic roles.
Bottom row: each predicate r in the vocabulary $\mathcal{V}$ is randomly true or false of each individual, following the predicate's semantic function.

the parent node (the end of the edge). Graphical models usually only use one type of edge – but we will need both.

A fully connected graph would not give any constraints on the joint distribution – independence assumptions correspond to *missing* edges. For any node, we know its distribution once we know the values for the nodes it's connected to (more precisely, its parent nodes and nodes connected by an undirected edge). Repetition of nodes can be indicated using a **plate**, a box drawn around the repeated nodes. In the bottom right corner of the plate, I will write the set which is iterated over.

A graphical model is shown in Fig. 3.8, for a situation with three individuals (X, Y, and Z), and two semantic roles (ARG1 and ARG2). This might represent an event (Y) and two participants in the event (X and Z). This graphical model only generates situations with this particular configuration of semantic roles, but I will explain later in this section how to generalise this to other kinds of situation. I will refer to the configuration of semantic roles in a situation as the **situation structure**, which can be represented as a directed graph, where the edges are labelled with semantic roles, but the nodes are not labelled.

The three nodes in the top row represent the individuals. The joint distribution for the these nodes represents knowledge about which situations are likely or unlikely. Each node in the top row is a pixie-valued random variable, representing the features of one of the individuals. There is also an undirected edge for each semantic role. The other edges in the graph are directed, pointing away, which means that the distribution for the pixie nodes is completely determined by the two undirected edges corresponding to semantic roles. This is so that the distribution over situations is defined without reference to any predicates – intuitively, the world is the same however we choose to describe it.

In machine learning, probabilistic graphical models usually use directed edges, because it makes inference easier (for example, LDA is a directed model). However, the edges between the pixie nodes in Fig. 3.8 are undirected, to avoid stipulating causal structure amongst the pixies in a situation. It might be tempting to suppose that the event is generated first, and then the participants of the event – but how should this generalise to multiple events? The edges between pixie nodes are undirected, so that this can easily generalise to situations of any size. I should also stress that undirected edges are compatible with "directed" semantic roles – the probability distributions do not need to be symmetric, which maintains the asymmetry of the

semantic roles. I have avoided indicating the direction of the semantic roles in Fig. 3.8, in order to make it clear that the edges are undirected in the probabilistic sense. In the semantic sense, the ARG1 and ARG2 roles are from Y to X and from Y to Z.

The two undirected edges together mean that we can factorise the joint distribution, as shown in (3.3). This factorisation means that X and Z are conditionally independent given Y. At first, this might seem like a strong assumption, but they are only independent if the value of Y is fully known. If all the variables are unobserved, we can have rich interactions between the three variables, which will be explored in Chapter 4. A concrete distribution of this form will be given in Chapter 5.

$$\mathbb{P}(X=x, Y=y, Z=z) \propto \mathbb{P}(X=x, Y=y)\,\mathbb{P}(Y=y, Z=z) \tag{3.3}$$

The three nodes at the bottom of Fig. 3.8 are truth values. The plate indicates that these are repeated for each predicate in the vocabulary $\mathcal{V}$, so we have a separate truth value for each predicate for each individual. Each truth value node has a single directed link coming from one pixie node. This means that the probability of truth only depends on the value of that pixie node, and not on the any other pixie node, or any other truth value. I will write $T_{r,X}$ for the truth value of a predicate r for a pixie-valued random variable X. The predicate's semantic function t_r determines the conditional distribution of this variable, as shown in (3.4). As with classical model theory, many predicates can be true of the same individual. For example, if the situation determined by the values of X, Y, and Z represented a dog chasing a cat, then nodes like $T_{dog,\,X}$, $T_{animal,\,X}$, and $T_{pursue,\,Y}$ would be true (with high probability), while $T_{democracy,\,X}$ or $T_{dog,\,Z}$ would be false (with high probability).

$$\mathbb{P}(T_{r,\,X} = \top \mid X = x) = t_r(x) \tag{3.4}$$

The graphical model in Fig. 3.8 defines a probabilistic model structure as described in §3.3 – the joint distribution over pixies gives us a distribution over situations, and we have a probabilistic truth value for each predicate for each individual. While this graphical model only generates situations with a particular situation structure, we can write down a similar graphical model for any situation structure. We introduce one pixie node for each individual, along with one undirected edge for each semantic role. We then introduce a truth value node for each predicate for each individual, with a directed edge from the pixie node to the truth value node. Although this means we have a separate graphical model for each situation structure, we can share parameters between the graphical models, using the same parameters whenever we see the same semantic role. A concrete way to do this will be presented in Chapter 5. Finally, if we have a distribution over situation structures (there are many ways to define a distribution over graphs), then we can define a generative process where we first draw a situation structure, and then use the corresponding graphical model.

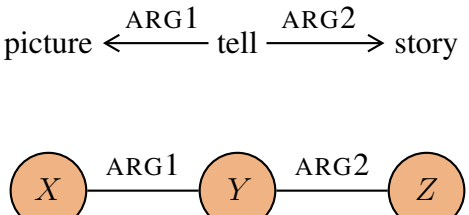


Figure 3.9: A simplified DMRS graph, which could be generated by Fig. 3.10 below. Such graphs are observed during training.

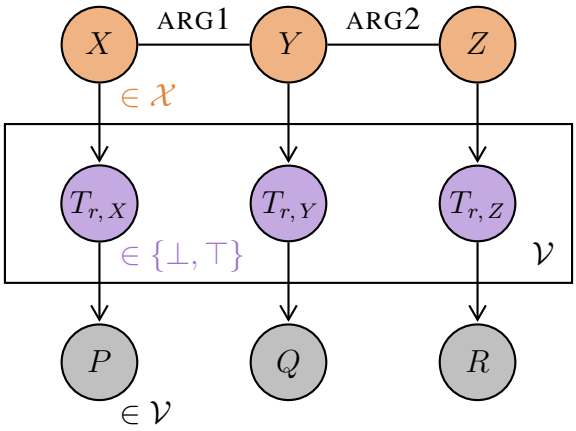


Figure 3.10: Probabilistic graphical model for Functional Distributional Semantics.
Top row: individuals represented by pixie-valued random variables X, Y, Z, jointly distributed according to the DMRS links.
Middle row: for each individual, each predicate r in the vocabulary $\mathcal{V}$ is randomly true or false according to its semantic function.
Bottom row: for each individual, we randomly generate one predicate, out of all predicates true of the individual. Only these nodes are observed.

3.6 Functional Distributional Semantics

So far, I have motivated and introduced a probabilistic version of model theory. I now turn to using it for distributional semantics. This will allow us to learn semantic functions and distributions over situations, based on data observed in a corpus. The basic idea is that observed utterances are true of some situation. After observing many utterances, we can learn both what kinds of situations are likely to exist, as well as how these situations are likely to be described.[11]

I will define a probabilistic graphical model which incorporates the above probabilistic generalisation of a model structure, and which can generate semantic dependency graphs like that in Fig. 3.9. The aim is to train the model in an unsupervised[12] way on a parsed corpus – that is, to optimise the model parameters to maximise the probability of generating the dependency graphs in the corpus. Although we cannot directly observe the model structure, we can define *latent* variables that represent individuals and truth values, which allows us to *indirectly* learn a model structure based on distributional information. In a certain sense, this is the inverse of supervised learning of classifiers – here, we observe the labels, but not the objects being classified. The fact that learning a classifier in this way is possible at all relies on structural information. While we don't observe the individuals in a situation, we can leverage structural linguistic knowledge in the form of semantic dependency graphs. These graphs might not tell us the features of the individuals, but they do give us dependencies between the individuals.

[11] There will be unfortunate biases in both halves, depending on the corpus – biases in the situations that the authors chose to talk about, and biases in how the authors chose to describe them. Identifying such biases is important for real-world applications of NLP, but that is beyond this scope of this thesis.

[12] I use the term "unsupervised" in the machine learning sense, following Ghahramani (2004): supervised learning requires both inputs and outputs, while unsupervised learning requires only inputs. The aim of supervised learning is to find a mapping from inputs to outputs, while the aim of unsupervised learning is to find structure in the inputs. Although I assume that dependency graphs are annotated in the training corpus, these are not desired outputs, but rather part of the input. Learning is unsupervised in this sense.

DMRS graphs are useful for this purpose, because they have a direct logical interpretation,[13] as explained in §1.3.3. DMRS nodes correspond to predicates and variables, and DMRS links correspond to semantic roles. Quantifiers (such as *the*, *every*, *a*) will be discussed in Chapter 4, and covered in more detail in Chapter 7. For the rest of this chapter, they will be dropped, and the reader may assume that all variables are existentially quantified.

I will use the term **topology** to refer to the structure of links in a DMRS graph. This is a graph where the links are labelled, but the nodes are not. Neglecting quantifiers, the topology of a DMRS graph corresponds to situation structure.[14] This means that we can extend the graphical model presented in the last section to generate DMRS graphs. As before, the graphical model only generates graphs of a particular topology, but a distribution over topologies would let us define a generative process for DMRS graphs of any topology.

The basic assumption is that each DMRS observed in a corpus is true, and corresponds to an unobserved, latent situation. This will not always be the case, but for text in an encyclopaedia, this assumption will hold most of the time. Each DMRS node corresponds to an individual, and each DMRS link corresponds to a semantic role. The graphical model in Fig. 3.10 generates dependency graphs with the topology of a transitive sentence – the predicates (P, Q, R) can be seen at the bottom, and the dependency links (ARG1, ARG2) can be seen at the top. For example, P, Q, and R might correspond to *pictures*, *tell*, and *stories*. While the predicates are observed, the individuals and truth values are not. Each observed predicate (grey nodes at the bottom) has a corresponding latent pixie (orange nodes at the top), as well as latent truth values for all other predicates in the vocabulary (purple nodes in the middle).

The directed edges in Fig. 3.10 mean that the generative process goes from the top to the bottom. First, we define a joint distribution over pixies and truth values, as described in §3.5. Then, for each pixie node (which represents an individual), we define a predicate node. Each predicate node is conditionally dependent on the truth values of *all* predicates, for that individual. The process of choosing a predicate out of all true predicates may be complex, potentially depending on speaker intention and other pragmatic factors. However, a simple option is to choose a predicate at random out of all true predicates (potentially weighting the predicates, so frequent predicates are generated more often). In §7.4, I discuss an approach to pragmatics that is compatible with this framework, which would allow a more sophisticated choice of predicate based on the truth values. For now, we can assume this simpler process.[15]

Putting this all together, the graphical model generates DMRS graphs in three stages. First, we generate a latent situation. Second, we generate latent truth values for each individual. Third, we generate a single predicate for each individual, which is what we observe in the corpus.

[13] A different formalism for semantic dependency graphs could be used, as long as there is a similar logical interpretation, which allows us to relate the graphs to a (probabilistic) model structure.

[14] This is a result of assuming that semantic roles are part of the situation (see §3.1). A more accurate model would need to separately represent DMRS topology and situation structure, and explain how the two are related.

[15] This doesn't define what happens if all predicates are false. For a large vocabulary, it is unlikely they are *all* are false, but for completeness, we can choose a backoff distribution, such as sampling from the whole vocabulary.

3.7 Assessment against Top-Down Goals

Having presented my functional framework for distributional semantics, I now return to the goals given in Chapter 2, and assess how well my framework can deal with them. Some of these are discussed in detail in subsequent chapters, and some have already been discussed in this chapter, but I collect an overview here for ease of reference.

3.7.1 Language and the World

As discussed in §2.1, one goal for a semantic theory is to relate its representations to the real world. In common with all distributional approaches, Functional Distributional Semantics learns from text alone, and so it is not grounded.

However, a functional model does make a clear distinction between concepts and referents – concepts are represented as semantic functions (probabilistic binary classifiers), while referents are represented as pixies. As explained in §3.4.1, a semantic function can also be seen a distribution over regions (when combined with a covariance function), which unites these two views of concepts.

The distinction between concepts and referents means that a functional model can be grounded in a more principled way than a vector space model. If we have some way to ground the semantic space, the semantic functions are naturally grounded. Although it is beyond the scope of this thesis, it would be possible to jointly train a functional model on both distributional data and grounded data such as labelled images. The crucial point is that we can use the same space to represent both the grounded data and latent pixies. This would mean that we can train semantic functions, both on the corpus data using the model described here, and on the labelled images using supervised learning.

3.7.2 Lexical Meaning

In §2.2, I discussed three aspects of lexical meaning – vagueness, hyponymy, and polysemy. Vagueness is built into the definition of a semantic function, and has been much discussed in this chapter. A simple illustration of the vagueness of a semantic function is shown in Fig. 3.11, where the function defines a series of larger regions (see §3.4.1).

With its close link to model theory, a functional model can simply borrow the definition of hyponymy in terms of subsets, the only difference being the use of a semantic space, rather than a set of individuals. A smaller region of space is more specific. We can define one semantic function as being a hyponym of another if it takes a smaller value at every point in the semantic space. An example is shown in Fig. 3.12. Although I am using probability theory rather than fuzzy logic, for reasons explained in §2.3.2, this definition is equivalent to Zadeh (1965)'s definition of the subset relation between fuzzy sets.

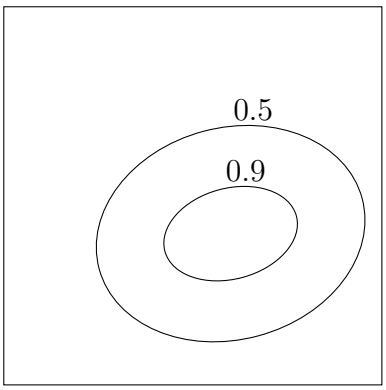


Figure 3.11: Vagueness.
A **contour plot** of a simple semantic function over a two-dimensional space. Each contour shows where the function takes a given value.

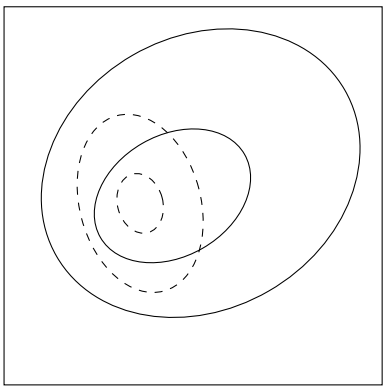


Figure 3.12: Hyponymy.
The contours are nested – the more general semantic function (solid lines) always has a higher value than the more specific one (dashed lines).

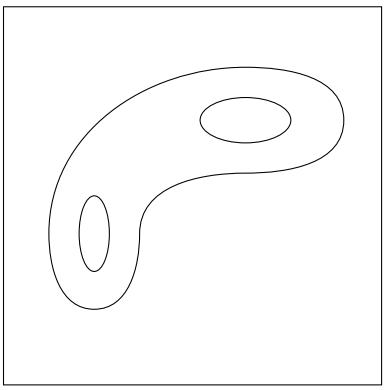


Figure 3.13: Polysemy.
This semantic function has two local maxima, each of which could be considered a sense. However, the two senses are linked.

Polysemy is harder to formally define, since our goal is to avoid finite sense inventories, as explained in §2.2.2. However, a semantic space allows a more fluid notion of sense, because both pixies and semantic functions can vary continuously. One way to look at the senses of a semantic function is to consider local maxima. If there is only a single maximum, as in Fig. 3.11, then we could say that the function has a single sense. However, if there are multiple maxima, as in Fig. 3.13, then we could consider each of these as a sense. Under this view, a sense could be very specific (the function takes high values only in a small region) or very general (it takes high values in a large region). The latter case allows us to represent highly underspecified senses, as argued for by Ruhl (1989). However, the flexibility of defining a function means that we can also accommodate cases where trying to define a highly underspecified sense might overgeneralise. In the example in Fig. 3.13, the two local maxima are linked together (so these senses are related) but the overall region has an irregular shape, so it would not be possible to define the region by simple constraints on each dimension. This allows us to capture concepts which have no simple set of necessary and sufficient properties, but which still exhibit family resemblance.

3.7.3 Sentence Meaning

In §2.3, I discussed three aspects of sentence meaning – logic, compositionality, and context dependence. As Functional Distributional Semantics builds on DMRS graphs, there is a natural link to logic. Truth values are an intrinsic part of the probabilistic model, and it is possible to use them to perform logical inference, as will be explained in more detail in §4.5. Furthermore, the DMRS graphs have come from the output of the ERG, which builds its semantics compositionally. Unlike a vector space model, a semantic dependency graph is not bounded in

size, so we do not have the problem of representing sentence meaning with a fixed number of dimensions. I will discuss composition in more detail in §4.4

Finally, because the model is probabilistic, it is easier to define context dependence, in a similar vein to the other probabilistic models discussed in §2.3.3. As a simple example, suppose we would like the model to generate *dogs chase cats* and *cats chase mice* but not to generate *dogs chase mice* and *cats chase cats*. In other words, we would like to capture a probabilistic dependence between the verb's arguments. In a vector space model, where each predicate is represented by a single vector, it is not clear how to capture this. However, by separating predicates from pixies, we can have two different event pixies which the *chase* predicate is true of, where one event pixie co-occurs with a dog-pixie ARG1 and cat-pixie ARG2, while the other event pixie co-occurs with a cat-pixie ARG1 and a mouse-pixie ARG2. This allows the model to generate DMRS graphs with a probabilistic dependence between arguments. This idea can be taken further, and used to describe context-dependent meaning, as I will explain in Chapter 4.

3.7.4 Learning Meaning

Finally, our last goal is for the model to be trainable in practice. The probabilistic graphical model presented in §3.6 was designed so that it can be trained on a corpus of DMRS graphs. A detailed implementation of this model will be given in Chapter 5, along with learning algorithms. Compared to vector space models, the additional expressiveness of a functional model does come at a cost. As we will see, the main challenge is the large number of latent variables – for every observed predicate, we have a latent pixie and latent truth values for all other predicates. This is also more challenging than a latent topic model like LDA. Firstly, while LDA only has one latent topic per token, my model has $|\mathcal{V}|$ latent truth values (where $\mathcal{V}$ is the vocabulary), which could easily number in the tens of thousands. Secondly, while a latent topic only has a small number of possible values, a latent pixie has a large number of possible values, since it lies in a high-dimensional semantic space. As will be explained in §5.3 and §5.4, approximate inference algorithms will be crucial for allowing the framework to be used in practice.

As mentioned at the end of §2.4, this thesis takes a step towards making it feasible to learn expressive semantic representations from distributional data. Using latent pixies allows us to extend representing meaning as a classifier to a distributional setting where referents are not observed. Using latent truth values can be seen as an approach in the spirit of Copestake and Herbelot (2012)'s ideal distributions, where logical forms need to be generalised from observed cases to unobserved ones. However, one part of the model that is not treated as latent is the logical structure, since this is included as part of the annotations in the training data. In principle, it could be possible to combine my approach with an unsupervised parsing algorithm, but that would be beyond the scope of this thesis. The approach I am taking here makes a clear separation between high-level logical structure and detailed lexical knowledge. While there are existing resources where the logical structure is largely correct, my aim in this thesis is to

fill the gap in lexical knowledge, learning semantic representations which are not only as fine-grained as vector space representations, but also expressive enough to meet our other goals for a semantic theory.

3.7.5 Comparison with Existing Frameworks

Finally, it is worth comparing my framework to the existing work discussed in §2.5. As I have argued above, a functional model can deal with a number of important semantic challenges. This allows us to use a single coherent model, rather than extending vector space models in different directions, or trying to building hybrid models.

Although I have found the type-driven tensorial framework a thought-provoking source of inspiration, the framework I have presented is fundamentally different. I have rejected the use of sentence spaces and linear maps (as explained in §2.3.1, §2.3.2, and §2.5.3), and I have proposed a uniform representation of predicates as unary functions, rather than using a system of semantic types. Applying each function only to its intrinsic argument removes the need for higher-order tensors. The most crucial difference, I believe, is the use of latent variables and probability distributions, which allow interesting nonlinear behaviour.

Compared to Goodman and Lassiter (2015)'s probabilistic framework, I have proposed a generative model that is simpler in two ways. Firstly, word meanings are represented as semantic functions, rather than as probabilistic programs. Secondly, the probabilistic graphical model in Fig. 3.8 jointly generates several pixies "in one step", rather than using a probabilistic program. On the other hand, pixies also introduce complexity, since they are structured objects lying in a semantic space, which may be high-dimensional. It could be possible to draw ideas from Goodman and Lassiter's work to make the probabilistic graphical model more complex, but this would need to be done carefully for it to be feasible for distributional semantics.

Finally, my framework has much in common with Cooper et al. (2015)'s probabilistic TTR framework. In particular, a semantic function corresponds to a TTR type and applying a semantic function gives a probabilistic type judgement. Both semantic functions and probabilistic type judgements are defined using conditional probabilities. However, an important innovation in my work is the use of a semantic space, which is defined without reference to any predicates. Although Cooper et al. do give some discussion of features, they do not make the semantic space explicit. Each pixie could in principle be described as a type (and so the whole semantic space could be described as a space of types), but then we would need to be careful to distinguish pixie types (which represent the world) from linguistic types (which represent how a speaker describes the world). It for this reason that I have avoided the term "situation type" (and Barwise and Perry (1983)'s related term "abstract situation"), and instead coined the term "pixie". Larsson (2013) and Fernández and Larsson (2014) are explicit in their use of a perceptual space, but they only use 1- or 2-dimensional spaces. In this thesis, I use a high-dimensional space and aim to learn all semantic functions from the data.

Chapter 4

From Bayesian Inference to Logical Inference

In Chapter 3, I defined a probabilistic graphical model for distributional semantics, which incorporates a probabilistic version of model theory. In this chapter, I discuss some theoretical benefits of this framework. First, in §4.1–4.3, I show how it allows a natural account of context dependence, which both gives it a practical advantage over vector space models, and also sheds light on context dependence as a linguistic phenomenon. I then discuss, in §4.4, how this kind of context dependence interacts with semantic composition, and finally in §4.5, how it can be used for probabilistic logical inference.

A Note on Notation

To make the equations easier to follow, I will use a more succinct notation. Previously, I have been careful to distinguish random variables (such as X), values of random variables (such as x), and events[1] of a random variable taking a value (such as $X = x$). However, for long equations, this notation can easily become verbose and hard to read. A more succinct notation, common in the NLP and machine learning literature, is to write $\mathbb{P}(x)$ instead of $\mathbb{P}(X = x)$. In other words, the event of a random variable taking a value is represented just by the value, with the random variable understood from the context. This notation is convenient for pixie-valued and predicate-valued random variables.

For truth-valued random variables, it is difficult to understand the variable from context, because we may have many such variables, but there are only two possible values. However, we can extend the notational distinction between upper and lower case, and write $\mathbb{P}(t_{r,X})$ instead of $\mathbb{P}(T_{r,X} = \top)$, for any predicate r and pixie-valued random variable X. This exploits the fact that truth has a distinguished status.

[1] In the probabilistic sense, not the semantic sense.

4.1 Context Dependence as Bayesian Inference

As discussed in §2.3.3, context dependence is a challenging semantic phenomenon. In particular, one goal for a semantic theory is to represent both occasion meanings (meanings in particular usages) and standing meanings (lexicalised abstractions over usages).

Searle (1980) discusses an interesting set of examples, focusing on the word *cut*. They note how a gardener cutting grass involves a very different kind of cutting from a child cutting a cake. There is something common to both events, but they involve different tools and different physical actions – driving a lawnmover, or slicing with a knife. However, Searle also notes how there are also less obvious interpretations of these expressions. For a gardener who sells turf (a section of soil containing living grass, which can be sold as a ready-grown lawn), cutting grass could also refer to cutting out an area of grass, including the soil.[2] This kind of cutting would more closely resemble cutting a cake, as both involve slicing out a section. Searle concludes that the interpretation of an expression is only possible given a background of assumptions.

Elman (2009) approaches context dependence from a cognitive science perspective, also discussing the word *cut*, and noting how there is a clear dependence between the participants of a cutting event (in particular, the agent, patient, instrument, and location). For example, specifying the agent induces a strong expectation about possible patients – we expect chefs, lumberjacks, and surgeons to cut different things. Elman concludes that meaning is a cue for understanding a situation, but language is always understood in context, incorporating world knowledge and extralinguistic information. In particular, experimental evidence contradicts a two-stage process where a listener first constructs a logical representation of a sentence and then incorporates contextual information.

Although both of the above authors give rather negative conclusions, I believe that both of their concerns can be dealt with using a probabilistic model, as long as we are careful to separate the logical form of a sentence from the situation which it describes. In particular, I propose to represent the standing meaning of a predicate with a semantic function, and an occasion meaning by a posterior distribution over pixies. Calculating an occasion meaning then reduces to Bayesian inference, conditioning on the context, which could include both linguistic and extralinguistic information. We can maintain a single standing meaning for *cut*, which can be used in Bayesian inference because it is defined as a conditional probability distribution. Meanwhile, background assumptions can be encoded in a prior distribution over situations. The combination of the two allows varied occasion meanings to fall out naturally.

In the simplest case, we have a situation containing a single individual X (a pixie-valued random variable). If we know the predicate r is true of X, we can apply Bayes' rule, as shown in (4.1). We can immediately see that the occasion meaning (the posterior distribution for X) depends on both world knowledge (the prior for X) and the semantic function t_r (the conditional

[2] There are other interpretations of *cut grass*, such as *adulterate marijuana*, but I'll focus on Searle's examples.

distribution for $T_{r,X}$ given X). This simple case of Bayesian inference can be naturally extended to condition on more context. In the more general case, we have a joint distribution not just over a single pixie and a single truth value, but over multiple pixies and multiple truth values.

$$\mathbb{P}(x \mid t_{r,X}) \propto \mathbb{P}(x)\,\mathbb{P}(t_{r,X} \mid x) \tag{4.1}$$

$$= \mathbb{P}(x)\, t_r(x) \tag{4.2}$$

The above equation gives an occasion meaning when we know some predicate is true. We can also consider an occasion meaning when we observe a speaker uttering a predicate. Rather than conditioning on a truth value $t_{r,X}$, we can condition on an observed predicate r, as in (4.3). If the predicate was generated randomly out of all predicates true of X, these two equations will give very similar results. A more nuanced way to choose a predicate will be discussed in §7.4.

$$\mathbb{P}(x \mid r) \propto \mathbb{P}(r \mid x)\,\mathbb{P}(x) \tag{4.3}$$

The above Bayesian account takes to heart the conclusion of both Elman and Searle that language cannot be understood independently of world knowledge. Here, world knowledge is encoded as a prior distribution over situations – and without a prior, we cannot construct a posterior. This means that, without world knowledge, we cannot construct occasion meanings. Even though Elman words their article provocatively, claiming that we should represent lexical semantics "without a lexicon", the context-dependent phenomena they describe can be accounted for using a lexicon of semantic functions. Having now motivated and sketched out a Bayesian account of context dependence, in the following section I give details of how this works using the graphical model of Functional Distributional Semantics.

4.2 Context Dependence in Functional Distributional Semantics

In this section, I use the probabilistic graphical model presented in §3.6 as Fig. 3.10, repeated here as Fig. 4.1 for convenience. This models events with two participants, such as a gardener cutting grass or a child cutting a cake. To model more participants (such as an instrument or location) we can use a larger graph, but I focus here on just these two participants.

World knowledge is encoded in the prior distribution over situations. This is represented in the top row of Fig. 4.1, where we have a joint distribution for the three pixie nodes. Intuitively, if we know the features of an event, we know the likely features of its participants (and vice versa). More formally, for any value y for the variable Y, we have a distribution over pixies for each of its semantic roles (and vice versa) – for example, we can calculate $\mathbb{P}(X = x \mid Y = y)$.

The standing meanings of *cut*, *grass*, and so on are represented by semantic functions. These functions are fixed, and given a situation, they determine probabilities for the truth values in the

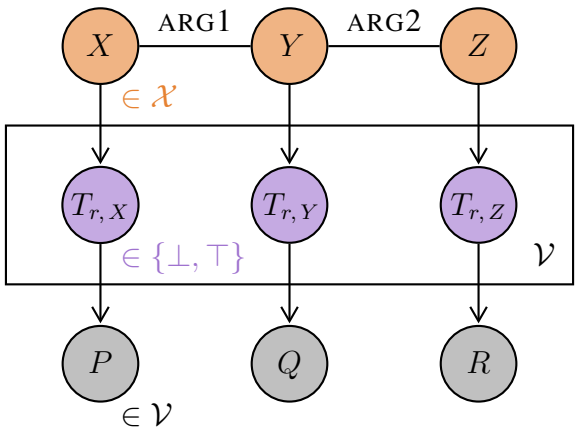


Figure 4.1: Probabilistic graphical model for Functional Distributional Semantics.
Top row: individuals represented by pixie-valued random variables X, Y, Z, jointly distributed according to the DMRS links.
Middle row: for each individual, each predicate r in the vocabulary $\mathcal{V}$ is randomly true or false according to its semantic function.
Bottom row: for each individual, we randomly generate one predicate, out of all predicates true of the individual. Only these nodes are observed.

middle row of Fig. 4.1. Importantly, we can have high probabilities of truth for quite different situations. For example, *cut grass* should be true of both mowing a lawn and slicing out a section of turf (more precisely, *cut* should be true of the event, and *grass* should be true of the event's ARG2).

An observed utterance is represented by the bottom row of Fig. 4.1. Each observed predicate has a corresponding latent pixie in the top row. The occasion meaning of each observed predicate is the posterior distribution of its corresponding pixie node. Given a prior over situations, and given observed predicates, we can calculate the posterior over situations (a joint distribution over pixies) as shown in (4.4).

$$\mathbb{P}(x, y, z \mid p, q, r) \propto \mathbb{P}(p \mid x)\, \mathbb{P}(q \mid y)\, \mathbb{P}(r \mid z)\, \mathbb{P}(x, y, z) \tag{4.4}$$

We are now in a strong position to respond to Searle and Elman's concerns. After observing *cut* and *grass*, the posterior over situations will have ruled out situations where those predicates are false (such as baking a cake or riding a bike), leaving high a probability for situations where those predicates are true. For most people most of the time, the prior over situations would assign a much higher probability to lawn-mowing situations than turf-slicing situations, and this will also be true for the posterior. However, if the listener has reason to have a high probability for turf-slicing situations (such as if they sell turf), then the situation would be reversed. In this way, the same standing meaning can lead to different occasion meanings, depending on the background assumptions of the listener.

Furthermore, because each pixie node is connected to all predicate nodes (via other pixie nodes and truth value nodes), the posterior distribution of each pixie node depends on *all* observed predicates. This means that after observing *lumberjack* and *cut*, we can already form a posterior for the final pixie node (for example, a tree pixie should be more likely than a grass pixie) – knowing that *lumberjack* was used to describe X restricts what values it is likely to take, which in turn restricts the likely values for Y, which in turn restricts the likely values for Z. Lexical knowledge and world knowledge interact to produce a prediction about the final

participant in the event. More precisely, we can calculate the marginal distribution for Z as shown in (4.5), where the joint distribution $\mathbb{P}(x, y, z)$ for the pixie nodes is crucial.

$$\mathbb{P}(z \mid p, q) \propto \sum_{x,y} \mathbb{P}(p \mid x)\, \mathbb{P}(q \mid y)\, \mathbb{P}(x, y, z) \tag{4.5}$$

While I have only discussed the immediate linguistic context, this approach generalises to other kinds of contexts. Since an occasion meaning is a posterior distribution, we could in principle condition on any kind of observation. The only requirement is that we have a joint distribution over all the random variables. Probabilistic graphical models provide a flexible way to extend a distribution to include more variables. For example, by adding observed nodes connected to the pixie nodes, we could condition on both linguistic and extralinguistic context.

This well-defined extensibility gives this account of context dependence a strong advantage over classical accounts such as those given by Kaplan (1979, 1989) and Recanati (2012). In these approaches, the standing meaning of a word (Kaplan's "character") is represented as a function from contexts to occasion meanings (Kaplan's "content"). However, without specifying what kind of function this is, this only sketches a solution in very general terms. What kind of context should be considered, and how is this function calculated? The above Bayesian account can be seen as providing a specific mathematical mechanism to construct such a function. Furthermore, by using Bayesian inference, we can naturally extend this to any kind of context, rather than stipulating a fixed set of arguments for the function. A Bayesian account of context dependence maintains the intuition behind the classical account, but is both more precise in its computational mechanism and more general in its dependence on arbitrary context.

Finally, I should note one computational difficulty. To calculate occasion meanings, we need to calculate posterior distributions. However, in the general case, calculating the posterior is intractable, a problem that we will also run into when training the model, as will be explained in §5.2. In practice, we need to approximate the posterior, and doing this will allow us to construct approximate occasion meanings, as explained in §5.4.1.

4.3 Disambiguation

As mentioned in §2.3.1, word sense disambiguation can be seen as a kind of context dependence. For a word that has a number of different senses, the context can make certain senses more likely than others. In the above examples, however, we could model *cut* has a having a single underspecified sense, following Ruhl (1989). While the context narrowed the occasion meaning to a particular kind of cutting, we would need a fine-grained sense inventory in order to see this as disambiguation.

I will now consider a case where it is easier to argue that there are distinct senses, because one sense is more abstract than the other. The term *bus* is polysemous between a vehicle and

a scheduled service. The senses are related, because a bus service necessarily involves a bus vehicle. However, a service scheduled at a particular time (say, *the 3:15 bus*) might use different vehicles on different days. In contrast, *car* has a vehicle sense, but not a service sense.[3] The preposition *on* is also polysemous (for discussion of its spatial senses, see: Herskovits, 1986), and I will discuss two of its senses here: a physical location (supported by a surface), and following a route or path (e.g. *on course*, *on my way*). In contrast, the preposition *in* has a location sense, but not a route sense.

The two senses of *bus* and *on* mutually disambiguate one another,[4] so that *on the bus* could mean either being on the roof of a vehicle, or riding a scheduled service (presumably inside the vehicle). In contrast, *on the car* and *in the bus* only refer to physical locations. Even if we want to represent the standing meanings of *on* and *bus* as each having a single underspecified sense, it is clear that their combination can lead to two distinct occasion meanings.

We can model this using the graphical model in Fig. 4.1, where Y is the prepositional event, and Z is the vehicle or service. The distribution defined by the ARG2 role should assign a high joint probability to Y being a location and Z being a vehicle, and also to Y being a route and Z being a service, but assign a low joint probability to the other combinations. The semantic function for *bus* would take high values for both vehicle pixies and service pixies, and the function for *on* would take high values for both location pixies and route pixies. After observing that Q is *on* and R is *bus*, the joint posterior over the pixies Y and Z would have two local maxima, one for each of the two meanings discussed above. In contrast, replacing *on* with *in* or replacing *bus* with *car* would lead to a distribution with a single maximum.

This kind of disambiguation relies on both world knowledge (the prior over situations) and lexical semantic knowledge (the semantic functions for predicates). If we define a sense to be a local maximum, as tentatively suggested in §3.7.2, then the number of occasion senses relies on both kinds of knowledge. Even if we represent a term like *bus* with a single standing sense, its interaction with other predicates may lead to a posterior distribution over situations where we can identify distinct occasion senses.

4.4 Semantic Composition

Context dependence is often discussed in relation to compositionality. As this functional framework builds on DMRS, we can compose semantic representations using the MRS composition algebra (Copestake et al., 2001; Copestake, 2007).[5] However, a functional model gives us a new interpretation of the DMRS graphs. We can see a DMRS graph equipped with semantic functions as a probabilistic binary classifier of situations.

[3] Except for some dialects, where *car* can mean *tram*.

[4] This observation is due to Martin Kay (p.c.).

[5] Admittedly, composing DMRS graphs using the MRS composition algebra is somewhat indirect. A composition algebra for DMRS has been drafted but has not yet been published.

In the simplest case, a graph with a single node is a classifier of situations containing a single individual – applying the semantic function to the individual can be seen as applying the graph to the situation. For example, *it is raining* would be represented by a DMRS graph with a single node, which can be used to classify raining situations. For a graph with multiple nodes, we can make use of the fact that DMRS has a logical interpretation. In this section, I will consider the case where all variables in the corresponding logical proposition are existentially quantified (for example, *a gardener is on a bus*). This special case is enough to illustrate compositionality and give a flavour of what it means in a probabilistic model. I will consider other quantifiers in the subsequent section, and give a fuller account in Chapter 7.

For a graph with multiple nodes (neglecting quantifiers), we can define a classifier by first aligning the graph topology to the situation structure, and then applying each semantic function to the corresponding individual. If all the functions return truth, then the classifier for the whole graph returns truth. This definition gives us a straightforward way to compose classifiers. We can see the composed classifier as the standing meaning of a phrase.

The generative model in Fig. 4.1 doesn't directly represent such a classifier. However, we can interpret the last step of the generative process as choosing between composed classifiers that return truth for the latent situation – the DMRS nodes are the predicate nodes in the bottom row, and their corresponding truth values are in the middle row (along with truth values for all other predicates). A set of predicates can only be generated if they all return truth, which would mean that the combined classifier would also return truth. Intuitively, the graphical model first generates a situation, and then generates a DMRS graph which is true of the situation.

So how does context dependence emerge as we compose DMRS graphs? If we start from two DMRS graphs, we can consider the two posterior distributions over situations defined by those graphs. Once we compose these two graphs, we have a new posterior distribution, over larger situations. However, this posterior is not the same as independently combining the posteriors of the two subgraphs. As the pixie nodes of the two subgraphs are now linked together, we have a joint distribution for all the pixie nodes, which depends on all the observed predicates. This means that, as we build a composed DMRS graph, we modify the posterior distributions at every step. In this way, we can see semantic composition as simultaneously composing the logical structure and refining the context-dependent meanings.

While composition of logical structures can be done without world knowledge, composition of occasion meanings relies on it. When two DMRS graphs are composed, we have one or more semantic roles linking the pixies of the two graphs. The posterior distribution for the composed graph depends on the new role(s), but information about these roles is part of world knowledge and not the lexicon. We can see this in terms of what McNally and Boleda (2017) term "conceptual afforded" composition and "referentially afforded" composition. Composition of DMRS graphs relies on knowledge of concepts stored in the lexicon – for example, verbs like *rain* describe events without any participants, so the DMRS graph cannot compose with an

argument, while verbs like *give* describe events with several participants, so the DMRS graph can compose with several arguments. In contrast, composition of occasion meanings relies on knowledge of referents in a situation – as we have seen in the examples earlier in this chapter, the posterior over situations is highly sensitive to the world knowledge encoded in the prior.

4.5 Logical Inference

The previous discussion considered inference about pixies, given observed predicates. In this section, I turn to inference about truth values, given other truth values, which is the domain of logical inference. The probabilistic graphical models we have been working with contain a node for the truth of each predicate for each individual. Using these nodes, we can convert logical propositions into statements about probabilities. Similarly to the above account of context dependence, we will use conditional probabilities defined using Bayesian inference.

A simple example of the kind of logical inference we might be interested in is whether the truth of one predicate implies the truth of another. To begin with, we can consider the simple case of a situation containing a single individual. If we know that one predicate is true of this individual, what does this tell us about other predicates? We can model this kind of inference using the probabilistic graphical model in Fig. 4.2, which is a special case of a probabilistic model structure, when there is only one individual X, and the vocabulary only contains two predicates, a and b. Unlike the previous sections of this chapter, we are not generating predicates – we are considering a probabilistic model structure, and attempting to perform the same kinds of logical inference that could be done in a classical model structure.

Using the graphical model in Fig. 4.2, we can calculate the probability of one predicate being true of X, given that the other predicate is true: $\mathbb{P}\left(t_{b,X} \mid t_{a,X}\right)$. To calculate this, we must marginalise out X, because the model defines the joint probability including X: $\mathbb{P}\left(x, t_{b,X}, t_{a,X}\right)$. This is analogous to removing bound variables when calculating the truth of quantified expressions in classical logic, a correspondence which will be taken further in Chapter 7.

The truth values are conditionally independent given X, but once we marginalise out X, the truth values are no longer independent. Intuitively, knowing one truth value tells us something about the latent pixie, which in turn tells us something about the other truth value. Marginalising out X requires summing over the semantic space $\mathcal{X}$, which is intractable in the general case, a difficulty that was previously noted for calculating occasion meanings. I will present a variational inference algorithm in §5.4, and apply it to logical inference in §5.4.2, allowing us to efficiently (but approximately) calculate such probabilities.

These conditional probabilities maintain a close link to classical logic, allowing us to set up an equivalence[6] between logical propositions and statements about conditional probabilities.

[6] Unless we define conditional probabilities given zero-probability events, the equivalence requires the logic to have "existential import", which means that a proposition involving *every A* entails that *some A* exists. This follows from the definition of conditional probability, $\mathbb{P}\left(B \mid A\right) = \mathbb{P}\left(A \wedge B\right) / \mathbb{P}\left(A\right)$, which is only defined if $\mathbb{P}\left(A\right) \neq 0$.

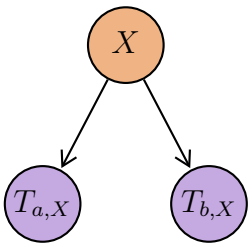


Figure 4.2: A simple probabilistic model structure with a single individual X, and two predicates a and b. We can recast logical inference in terms of conditional probabilities, such as $\mathbb{P}(t_{b,X} \mid t_{a,X})$, the probability that b is true, given a is true.

Firstly, we can say that universally quantified propositions correspond to conditional probabilities being equal to 1. For example, (4.6) and (4.7) are equivalent. Intuitively, conditioning on $T_{a,X}$ means restricting to those pixies x for which the predicate a is true. If the conditional probability of $T_{b,X}$ being true is 1, then the predicate b is true for all of those x.

$$\forall x \in \mathcal{X},\ a(x) \Rightarrow b(x) \tag{4.6}$$

$$\mathbb{P}(t_{b,X} \mid t_{a,X}) = 1 \tag{4.7}$$

Similarly, we can say that existentially quantified propositions correspond to conditional probabilities being nonzero. For example, (4.8) and (4.9) are equivalent. Intuitively, if there a nonzero probability of $T_{b,X}$ being true, then there is some pixie for which it is true. Setting up an equivalence with a conditional probability might seem surprising, because the logical proposition is symmetric in a and b, but the conditional probability is not. However, if we have $\mathbb{P}(t_{b,X} \mid t_{a,X}) > 0$, then we also have $\mathbb{P}(t_{a,X} \mid t_{b,X}) > 0$. Writing it as a conditional probability gives the same form to (4.7) and (4.9), and generalises better, as we will see in Chapter 7.

$$\exists x \in \mathcal{X},\ a(x) \wedge b(x) \tag{4.8}$$

$$\mathbb{P}(t_{b,X} \mid t_{a,X}) > 0 \tag{4.9}$$

Furthermore, classical rules of inference hold under the above equivalence. For example, from $\mathbb{P}(t_{b,X} \mid t_{a,X}) = 1$ and $\mathbb{P}(t_{c,X} \mid t_{b,X}) = 1$, we can deduce that $\mathbb{P}(t_{c,X} \mid t_{a,X}) = 1$. In classical syllogistic logic, this is known as the "Barbara" syllogism – from $\forall x,\ a(x) \to b(x)$ and $\forall x,\ b(x) \to c(x)$, we can deduce that $\forall x,\ a(x) \to c(x)$. Proofs of the general equivalence and this special case are given in §4.5.1 below. Admitted, this is only equivalence with a syllogistic logic, which is quite restricted. I will extend this approach to deal with propositions with multiple quantifiers in Chapter 7. However, we can already see some benefits of a probabilistic approach – in a sense, the probabilities are more "fine-grained" than the logical propositions, because probabilities lie in the range $[0, 1]$, but the universal and existential quantifiers only need to know if the probabilities are 0, 1, or an intermediate value.

In practice, the probabilities will never be exactly 0 or exactly 1,[7] because a distributional model will only learn soft constraints, as discussed in §5.1.2. In some cases, we can get around this – for example, if we are dealing with a fixed set of individuals, and some truth values are observed (say, if they are part of the common ground in a discourse). In other cases, it doesn't matter if probabilities are always intermediate, because the probability itself can be informative – for example, $\mathbb{P}\left(t_{b,X} \mid t_{a,X}\right) = 0.999$ would mean that, if $a(X)$ is true, it is *almost always* the case that $b(X)$ is also true. The conditional probability represents the degree to which a implies b, in an intuitive sense: the higher the value, the closer we are to *every*; and the lower the value, the closer we are to *no*.

One use of such conditional probabilities is to define a measure of lexical similarity. To make the measure symmetric, we can multiply the conditional probabilities in both directions, as shown in (4.10). This measures the degree to which a pair of predicates imply each other, and it will be used in the experiments in §6.2.1.

$$\text{logical-sim}(a, b) = \mathbb{P}\left(t_{b,X} \mid t_{a,X}\right) \mathbb{P}\left(t_{a,X} \mid t_{b,X}\right) \tag{4.10}$$

This measure casts similarity in terms of logical inference, and can be contrasted with similarity in terms of feature overlap. Intuitively, if we know that a is true of one individual, and that b is true of another, we can ask whether those individuals are similar. In other words, if we have a similarity measure for pixies, this induces a similarity measure for predicates. Given a prior over pixies, each semantic function defines a posterior over pixies. This is similar to the occasion meanings we saw earlier in this chapter, with the only difference being that we are conditioning on a truth value, rather than an observed predicate. Given two semantic functions, we can define two distributions over pixies, as shown in Fig. 4.3. We can find the expected similarity between pixies from those distributions, as shown in (4.11), where x is drawn from the posterior for a, y is drawn from the posterior for b, and sim is some similarity measure over $\mathcal{X}$.

$$\text{featural-sim}(a, b) = \mathbb{E}_{x,y}\left[\text{sim}(x, y)\right] \tag{4.11}$$

These two similarity measures allow us to distinguish cases where predicates imply each another, and cases where referents are similar. For example, no cat is a dog, and no dog is a cat, but it makes sense to say that cats and dogs are similar, because they share many features. The "logical similarity" of *cat* and *dog* is low, but their "featural similarity" is high.

However, while a functional framework can draw this distinction in theory, learning such a distinction is difficult when learning from distributional data. As Copestake and Herbelot (2012) note, distinguishing logical similarity (which they term "substitutability") from featural similarity (which they term "contextual similarity") requires knowing extensions. However, in distributional semantics, such information is not overtly available. In principle, it could be

[7] Except for contraditions (b is the negation of a) and tautologies (b is a).

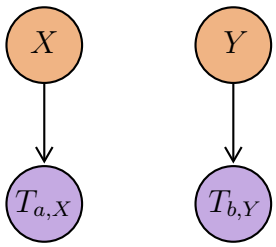


Figure 4.3: Similarity in terms of feature overlap. X and Y are independent and identically distributed random variables. By conditioning X on the truth of a, and Y on the truth of b, and then measuring the similarity of the pixies, we get a similarity measure for the predicates.

possible to use a coreference resolution system to determine when predicates share a referent. This would make it possible to observe, for example, that *cat* and *dog* never share a referent, even when they occur close together in the same text. If two predicates can both be true of the same individual, we would expect to observe coreference. The lack of coreference should lead us to deduce that the predicates should be true in disjoint regions of the semantic space. However, implementing a system that uses coreference information is beyond the scope of this thesis. Furthermore, similarity of such pairs of predicates are inconsistently annotated in the available test data (see §6.2.1), which would make it difficult to evaluate if the system has learnt such distinctions correctly. Nonetheless, the ability to make such a distinction is a clear advantage over vector space models. Exploiting the distinction will be a task for future work.

Finally, in the general case, there are multiple individuals in a situation, as illustrated in Fig. 4.4. This opens up the possibility of inferring what is true of one individual, given what is true of another. For example, if we know that a person is cutting grass, we could ask how likely it is that the person is also a gardener (likely), an artist (less likely), or a flowerpot (very unlikely). As before, we can answer this question by calculating a conditional probability: $\mathbb{P}\left(t_{d,X} \mid t_{person,X}, t_{cut,Y}, t_{grass,Z}\right)$, where d is *gardener*, *artist*, or *flowerpot*.

As with the one-pixie case considered in Fig. 4.2, the truth values are conditionally independent given the latent pixies, but they are not independent. The latent pixies share a joint distribution, and because the truth values are connected via the latent pixies, the truth of one predicate depends on all the other predicates. Intuitively, knowing one truth value tells us something about that predicate's latent pixie, which in turn tells us something about the other pixies in the situation, which in turn tells us something about the other truth values. This dependence is similar to the dependence that we saw for occasion meanings.

Calculating the above conditional probability requires marginalising out all the latent pixies from the joint distribution, and as with the one-pixie case, doing this in practice requires approximate inference algorithms, as will be discussed in §5.4.2. This algorithm will make the connection between logical inference and context dependence even clearer – to calculate the probability of a predicate being true of an individual, given a DMRS graph that describes the situation, we first calculate the (approximate) occasion meanings of the nodes in the DMRS

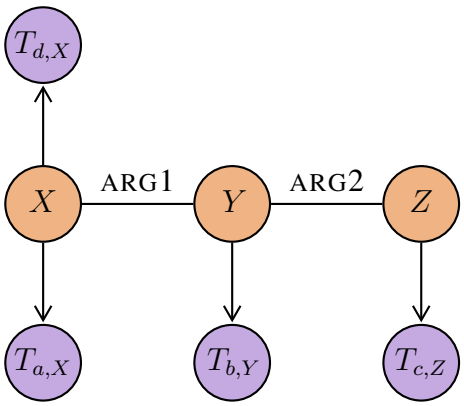


Figure 4.4: Logical inference for three pixies and four predicates: we know that a, b, c are true of X, Y, Z, respectively, and we would like to infer whether d is true of X, which can be calculated as $\mathbb{P}\left(t_{d,X} \mid t_{a,X}, t_{b,Y}, t_{c,Z}\right)$. The distribution for $T_{d,X}$ depends on all the other truth values, because it is indirectly connected to them via the latent pixies.

graph, and then apply the predicate's semantic function to the corresponding occasion meaning.

As with the one-pixie case, there is an equivalence between logical propositions and statements about probabilities. For example, we can say (4.12) and (4.13) are equivalent. Note that ARG1 in the logical proposition does not correspond to a random variable in the conditional probability – it is instead represented in the structure of the graphical model (the edges between orange nodes in Fig. 4.4). As before, this conditional probability will never be exactly 0 or 1, but it is nonetheless a useful quantity, as we will see in the experiments in §6.2.2 and §6.2.3.

$$\exists x, y \in \mathcal{X},\ a(y) \wedge b(x) \wedge \textsc{arg1}(y, x) \tag{4.12}$$

$$\mathbb{P}\left(t_{b,X} \mid t_{a,Y}\right) > 0 \tag{4.13}$$

4.5.1 Proof of Equivalence

In this section, I prove the equivalence between logical propositions and constraints on probabilities. This section can be safely skipped by readers not interested in a formal proof.

4.5.1.1 Proof of General Case

Syllogisms are classically expressed in set-theoretic terms. A quantified proposition of the form *Q a's are b's*, where *Q* is some quantifier, gives constraints on the sizes of sets. Writing A for the extension of *a*, and B for the extension of *b*, a quantified proposition constrains the sizes of the sets $A \cap B$ and $A \setminus B$, and says nothing about the size of B.

For the existential quantifier $\exists$, we have:

$$|A \cap B| > 0$$

For the universal quantifier $\forall$, we have the following, where the second constraint assumes existential import:

$$|A \setminus B| = 0$$
$$|A \cap B| > 0$$

From these definitions, we can use standard set theory to prove all and only the valid syllogisms. To show equivalence with our probabilistic framework, we first note a measure-theoretic correspondence – sizes of sets form a measure (the **counting measure**), and probabilities also form a measure. The above conditions are all constraints on sizes of sets being zero or nonzero, so it suffices to show that the sizes and probabilities are **measure-theoretically equivalent**: they agree on which sets have measure zero.

Given a classical model structure with extensions for *a* and *b*, and assuming that we have a semantic space that is fine-grained enough to represent each individual by a distinct pixie, we can define a prior over the semantic space to be a uniform distribution over the pixies corresponding to the individuals.[8] Finally, a classical truth-theoretic function is a special case of a semantic function, where all conditional probabilities are either 0 or 1.[9] So we have now constructed a probabilistic model following Fig. 4.2.

Because the semantic space is fine-grained enough to embed all individuals as distinct pixies, the sets of individuals A and B have corresponding sets of pixies. As we have defined the prior for X so that its value always corresponds to an individual, A and B also have corresponding events $X \in A$ and $X \in B$. In what follows, I will write $\mathbb{P}(A)$ to denote $\mathbb{P}(X \in A)$, and so on. (The use of upper case A and B follows the set-theoretic convention, rather than the random-variable convention.)

First, we note that $\mathbb{P}(B \mid A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)}$ is defined only when $\mathbb{P}(A) > 0$. This will give us existential import. We need to prove that the statements about conditional probabilities are equivalent to the statements about which events have zero probability.

For $\exists$, if $\mathbb{P}(B \mid A) > 0$, then:

$$\mathbb{P}(B \mid A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)} > 0$$
$$\mathbb{P}(A \cap B) > 0$$

We can say nothing further about the probability $\mathbb{P}(A \setminus B) = \mathbb{P}(A) - \mathbb{P}(A \cap B)$, which may be zero or nonzero, just as in the classical case.

Conversely, if $\mathbb{P}(A \cap B) > 0$, then $\mathbb{P}(B \mid A) > 0$.

[8] Technically, we can choose any distribution measure-theoretically equivalent to this one (assigning nonzero probability precisely to those pixies). I have suggested a uniform distribution for concreteness.

[9] For pixies that do not correspond to any individual, we can set the conditional probability to be 0, for completeness. However, it doesn't actually matter, because these pixies have zero probability. In classical logic, we don't need to worry about generalisation to new situations.

For $\forall$, if $\mathbb{P}(B \mid A) = 1$, then:

$$\begin{aligned}
\frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)} &= 1 \\
\mathbb{P}(A \cap B) &= \mathbb{P}(A) \\
\mathbb{P}(A \cap B) &= \mathbb{P}(A \cap B) + \mathbb{P}(A \setminus B) \\
\mathbb{P}(A \setminus B) &= 0
\end{aligned}$$

And we also have:

$$\begin{aligned}
\mathbb{P}(A \setminus B) + \mathbb{P}(A \cap B) &= \mathbb{P}(A) > 0 \\
\mathbb{P}(A \cap B) &> 0
\end{aligned}$$

Conversely, if $\mathbb{P}(A \setminus B) = 0$ and $\mathbb{P}(A \cap B) > 0$, then:

$$\begin{aligned}
\mathbb{P}(A) &= \mathbb{P}(A \cap B) + \mathbb{P}(A \setminus B) \\
&= \mathbb{P}(A \cap B)
\end{aligned}$$

$$\implies \mathbb{P}(B \mid A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)} = 1$$

This demonstrates the equivalence.

4.5.1.2 Example – Barbara

We can prove the Barbara syllogism as follows:

$$\begin{aligned}
\mathbb{P}(B \mid A) = 1 \implies \mathbb{P}(A \setminus B) &= 0, \\
\mathbb{P}(A) &> 0
\end{aligned}$$

$$\mathbb{P}(C \mid B) = 1 \implies \mathbb{P}(B \setminus C) = 0$$

$$\begin{aligned}
\mathbb{P}(A \setminus C) &= \mathbb{P}(A \cap B \setminus C) + \mathbb{P}(A \setminus B \setminus C) \\
&\leq \mathbb{P}(B \setminus C) + \mathbb{P}(A \setminus B) \\
&= 0
\end{aligned}$$

$$\begin{aligned}
\mathbb{P}(A \cap C) &= \mathbb{P}(A) - \mathbb{P}(A \setminus C) \\
&= \mathbb{P}(A)
\end{aligned}$$

$$\implies \mathbb{P}(C \mid A) = \frac{\mathbb{P}(A \cap C)}{\mathbb{P}(A)} = 1$$

Chapter 5

Implementation and Inference Algorithms

In Chapter 3, I described a general probabilistic framework, which extends model-theoretic semantics, and which can be used for distributional semantics. However, a probabilistic graphical model only gives constraints on a distribution, rather defining a specific distribution. Here I explicitly construct a distribution that implements this graphical model. As I have already noted, the large number of latent variables present a learning challenge, so I will try to keep the network architecture as simple as possible, so as not to introduce additional challenges. Experiments using this implementation will be presented in Chapter 6.

The most crucial part of this chapter is §5.1, which gives details of the probabilistic model. I discuss how to train the model in §5.2, and I discuss approximate inference algorithms in §5.3 and §5.4. Longer derivations have been set aside as subsections, to make it easier for a reader to skip them, if they are not interested in such details.

A Note on Notation

Although matrix-vector notation is common in NLP, it is often cumbersome, with a frequent need to use matrix transposition. Furthermore, it is always unclear for higher-order tensors.

I will use **index notation** (also called **suffix notation**), where tensors (including vectors) are written with subscripts that range over dimensions of the vector space. An n^{th}-order tensor has n subscripts. For example, a_i is a vector, a_{ij} is a matrix, a_{ijk} is a third-order tensor, and so on. I will also use the **Einstein summation convention**, which means an index which is repeated is summed over. For example, $a_i b_i$ represents a dot product between vectors a and b, and $a_{ij} b_{jk}$ represents a matrix multiplication of matrices a and b.

Indices which are not dimensions of a vector space will be written as superscripts in brackets. For example, a truth value which was written as $T_{r,X}$ in the previous two chapters will be written as $T^{(r,X)}$, because predicates and random variables are not dimensions of a space. In contrast, the semantic space will be a vector space, so pixies can be written as x_i.

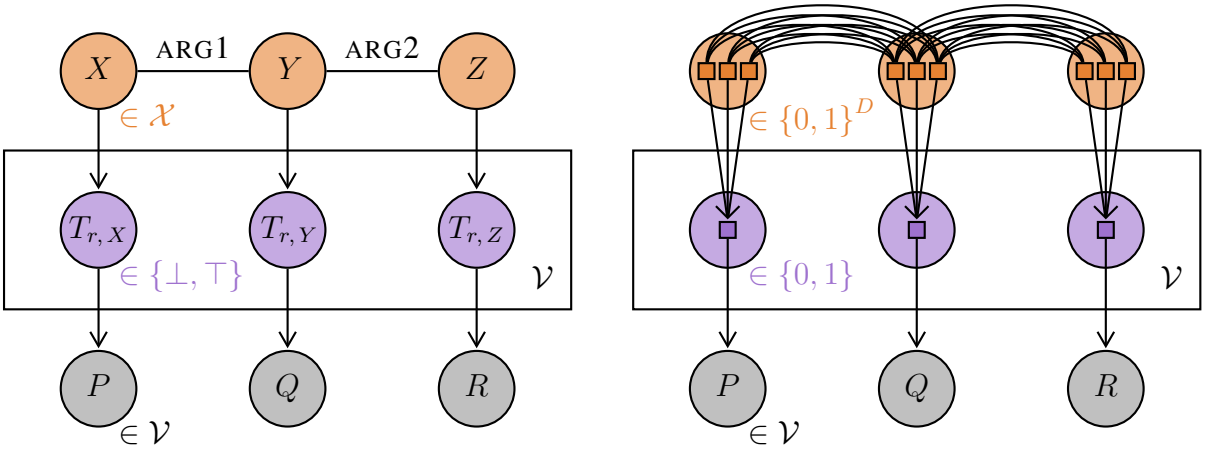


Figure 5.1: The neural network on the right implements the probabilistic graphical model on the left (repeated from Fig. 3.10). Each square represents a binary unit (with D units per pixie). **Top row:** pixies are binary-valued vectors, forming a CaRBM. For each semantic role, there are connections determining how likely it is that pairs of units are active at the same time. **Middle row:** each semantic function is a one-layer feedforward network, with a single output in the range $[0, 1]$, which is used to generate a truth value. **Bottom row:** true predicates are weighted by frequency and one is generated.

5.1 Network Architecture

The neural network architecture is shown in Fig. 5.1, alongside the graphical model defined in §3.6. Because of the undirected edges in the graphical model, there are no existing models used in NLP that can easily be adapted for this purpose. I take the semantic space $\mathcal{X}$ to consist of D-dimensional binary-valued vectors. Furthermore, I assume that these vectors are **sparse** – for any particular vector, most dimensions have value 0. In particular, I assume that exactly C of these dimensions are 1, as shown in (5.1). Intuitively, each dimension represents a different feature. Sparse representations have been shown to be beneficial in NLP, both for applications and for interpretability of features (for example: Murphy et al., 2012; Faruqui et al., 2015). Binary-valued vectors both encourage sparsity and make it easier to define a joint distribution over several pixies, because the space is finite (although still very large, for reasonable D).

$$\mathcal{X} = \left\{ x \in \{0,1\}^D \;\middle|\; \sum_i x_i = C \right\} \tag{5.1}$$

The rest of the architecture has been chosen to match this semantic space. We need to specify how the edges corresponding to semantic roles determine a joint distribution for the pixie nodes, and we need to specify how semantic functions determine probabilities of truth.

To define the joint distribution for pixie nodes, the basic idea is that we want to specify how the features of one pixie node should co-occur with the features of other nodes that are linked to it. I will refer to one dimension of a pixie node as a **unit**. If a unit's value is 1, I will call it

active or **on**, and if its value is 0, I will call it **off**. I will denote a semantic role l from x to y as $x \xrightarrow{l} y$. As explained in §3.5, the semantic role is directed, while the edge in the graphical model is undirected but asymmetric. I define the distribution over situations using a **Restricted Boltzmann Machine (RBM)** (Smolensky, 1986; Hinton et al., 2006; Hinton, 2010), but rather than having connections between "hidden" and "visible" units (as in a normal RBM), we have connections between units of pixies, whenever two pixies are linked by a semantic role. This is illustrated in the top row of Fig. 5.1. A variant of the RBM which is suitable for sparse vectors is the **Cardinality RBM (CaRBM)**, introduced by Swersky et al. (2012). This introduces a constraint fixing the total number of active units, which matches the space defined by (5.1).

Let S be a random variable for a situation, distributed according to this RBM. The probability of S taking the particular value s depends on the **energy** $E(s)$, as shown in equations (5.2) and (5.3). A high energy denotes an unlikely situation. Conversely, a large negative energy denotes a likely situation.[1] The normalization constant Z ensures that the total probability across all situations sums to 1.

The energy of a situation depends on the connections of the RBM, plus bias terms, as shown in (5.4). Each semantic role l has a corresponding parameter matrix $w_{ij}^{(l)}$, which determines the strength of association between different features of the linked pixies. Indices i and j vary over dimensions of the space $\mathcal{X}$. The first term in (5.4) sums the contributions over all semantic roles $x \xrightarrow{l} y$ between pixie nodes. Because I am using sparse representations, I assume that all link parameters are non-negative – this means that we only have a nonzero parameter when a pair of units are likely to be active together (and *not* when they are *unlikely* to be active together). This is similar to the use of *positive* PMI in vector space models. I also introduce a bias vector b_i, to control how likely each dimension is to be active, regardless of the semantic roles. The second term in (5.4) sums the biases over all pixies x in the situation. The parameter tensors $w_{ij}^{(l)}$ and b_i together define the distribution over situations.

$$\mathbb{P}\left(S=s\right) = \frac{1}{Z}\exp\left(-E(s)\right) \tag{5.2}$$

$$Z = \sum_{s'}\exp\left(-E(s')\right) \tag{5.3}$$

$$-E(s) = \sum_{x \xrightarrow{l} y \text{ in } s} w_{ij}^{(l)} x_i y_j - \sum_{x \text{ in } s} b_i x_i \tag{5.4}$$

I now turn to the semantic functions $t^{(r)}$, which map from pixies to probabilities of truth. As previously explained, the semantic functions determine the distributions for the truth value nodes, as shown in (5.5). I implement each semantic function as a **feedforward network**. Unlike the RBM, where the connections go in both directions, a feedforward network maps from an input to an output. In this case, the output will be a value in $[0, 1]$, interpreted as the

[1] The minus sign may seem unfortunate in machine learning, but the term "energy" has been inherited from statistical physics. A physical system is more likely to be in a low energy state.

probability of truth. For simplicity, each semantic function is only a single-layer network, as given in (5.6) and (5.7) and shown in Fig. 5.1. Each predicate r has a parameter vector $v_i^{(r)}$, which determines the strength of association with each dimension of the semantic space. As with the semantic role parameters, I assume that these parameters are all non-negative, so that only positive associations are stored in the model. I also include a bias term $a^{(r)}$, which controls how likely the predicate is to be true or false in general. These together define a score $F(x, r)$, as shown in (5.7).[2] This is passed through the sigmoid function to give a value in the range $[0, 1]$, as shown in (5.6). The parameter vector $v_i^{(r)}$ looks just like a set of feature weights as proposed in the prototype theory of concepts (Rosch, 1975, 1978).

$$\mathbb{P}\left(T^{(r,X)} = \top \,\middle|\, X = x\right) = t^{(r)}(x) \tag{5.5}$$

$$t^{(r)}(x) = \frac{1}{1 + \exp\left(F(x, r)\right)} \tag{5.6}$$

$$-F(x, r) = v_i^{(r)} x_i - a^{(r)} \tag{5.7}$$

Given the semantic functions, choosing a predicate for a pixie can be hard-coded, for simplicity, as was discussed briefly in §3.6. I will write $R^{(X)}$ for the predicate node corresponding to a pixie node X. The probability that $R^{(X)}$ takes the value r depends on whether it is true of the pixie, and depends on the **frequency** $f^{(r)}$ of the predicate, where frequencies are defined as a proportion of observed tokens, so that $\sum_r f^{(r)} = 1$. This distribution over predicates is shown in (5.8) and (5.9), where $Z(x)$ normalises the distribution.[3]

$$\mathbb{P}\left(R^{(X)} = r \,\middle|\, X = x\right) = \frac{1}{Z(x)} f^{(r)} T^{(r,X)} \tag{5.8}$$

$$Z(x) = \sum_{r'} f^{(r')} T^{(r',X)} \tag{5.9}$$

However, the trouble with using this definition is that we cannot calculate the probability of generating a predicate without sampling a truth value for every predicate in the vocabulary, because $Z(x)$ is a random variable that depends on all truth values for that pixie. A slightly more tractable alternative is to use a **mean-field** approximation, where we consider the expected value of each truth value node, rather than sampling a value. In other words, we use probabilities of truth, rather than sampling truth values. We can then calculate a distribution based on these expected values, as shown in (5.10) and (5.11), which approximate the result of (5.8) and (5.9). Now, the probability of generating a predicate r depends on the predicate's frequency $f^{(r)}$ and the value of its semantic function $t^{(r)}(x)$. We still need to consider the whole vocabulary to

[2] The minus sign in the definition of F is to show the similarity with E. We can view $F(x, r)$ as the energy associated with r being true of x, where falsehood always has an energy of 0.

[3] If it is usually the case that many predicates are true at the same time, this implicitly assumes that all predicates are true equally often, but frequent predicates are more likely to be generated. An alternative would be to use some function of the frequency, such as taking it to some power in the range $[0, 1]$.

calculate the normalisation constant $Z(x)$, but it is at least no longer a random variable.

$$\mathbb{P}\left(R^{(X)}=r \,\middle|\, X=x\right) = \frac{1}{Z(x)} f^{(r)} t^{(r)}(x) \tag{5.10}$$

$$Z(x) = \sum_{r'} f^{(r')} t^{(r')}(x) \tag{5.11}$$

5.1.1 Summary of Architecture

To ensure clarity in the above description, I used the verbose notation that distinguishes random variables, values, and events of a random variable taking a value. In the following sections, I will use the more succinct notation explained in Chapter 4, where $\mathbb{P}\left(X=x\right)$ is written as $\mathbb{P}\left(x\right)$, and $\mathbb{P}\left(T^{(a,X)}=\top\right)$ is written as $\mathbb{P}\left(t^{(a,X)}\right)$.

Below is a summary of the above model, using the succinct notation. To further simplify the equations, I have given unnormalised probabilities, without the normalisation constants (writing $\propto$ to indicate this). Equation (5.12) shows that the distribution over situations is implemented by an RBM, with connections between pixies that are linked by a semantic role (plus bias terms that make some parts of the semantic space more likely). Equation (5.13) shows that a semantic function is implemented by a one-layer feedforward network. I have further simplified the equation by writing σ for the sigmoid activation function, $\sigma(x) = (1+\exp(-x))^{-1}$. Finally, equation (5.14) shows that predicates are generated according to their frequency and the value of the semantic function. These three equations correspond to the three rows of Fig. 5.1, from top to bottom.

$$\mathbb{P}\left(s\right) \propto \exp\left(\sum_{x \xrightarrow{l} y \text{ in } s} w_{ij}^{(l)} x_i y_j - \sum_{x \text{ in } s} b_i x_i\right) \tag{5.12}$$

$$t^{(r)}(x) = \sigma\left(v_i^{(r)} x_i - a^{(r)}\right) \tag{5.13}$$

$$\mathbb{P}\left(r \,|\, x\right) \propto f^{(r)} t^{(r)}(x) \tag{5.14}$$

5.1.2 Soft Constraints

It should be noted that this model only implements soft constraints on semantics – indeed, it would be difficult to learn hard constraints from corpus data alone. This means that, all distributions over pixies have a nonzero probability for every pixie, and all semantic functions assign a nonzero probability of truth to every pixie. By analogy with a traditional model structure, we might want to have zero values, to indicate that a certain pixie or situation is impossible, or that a certain predicate is definitely false of a certain pixie. However, from a Bayesian point of view, a zero probability is problematic – it would imply that, no matter what new evidence an agent observes, they cannot change their mind.

In practice, some probabilities will be vanishingly small. In fact, to make interesting predictions, this is *necessary* – for high-dimensional spaces, an interesting subspace (perhaps representing a domain, like rock climbing or ballroom dancing) may be small compared to the whole space. For example, suppose we have 1000 dimensions, with only 40 active at once. This gives 10^{72} pixies. A subspace only using 200 dimensions has 10^{42} pixies, or one part in 10^{30} of the whole space! To define a distribution $\mathbb{P}(x)$ over the space, with most probability mass in this subspace, pixies in the subspace must be at least 10^{30} times more likely than pixies outside.

Suppose a predicate is probably true of pixies in this subspace, and probably false of pixies outside. Intuitively, knowing that the predicate is true should restrict our attention to the subspace. More formally, given a single pixie node, with a uniform prior over the space, and given a truth value node which is observed to be true, we might expect the pixie node's posterior to assign most probability mass to the subspace. For this to happen, the probability $\mathbb{P}(t \mid x)$ of the predicate being true must be 10^{30} times larger for pixies in the subspace than for pixies outside. So, for a semantic function to be useful, it must be close to a step function. This makes it look more like a traditional truth-conditional function with only 0 and 1 as values.

5.2 Gradient Descent

To train the above architecture, we need to determine values for its parameters – the semantic role parameters $w_{ij}^{(l)}, b_i$, and the semantic function parameters $v_i^{(r)}, a^{(r)}$. We are aiming to optimise these parameters to maximise the probability of observing the training data. As described in §3.6, each data point is a DMRS graph observed in a parsed corpus.

When the probability of the training data is viewed as a function of the model parameters, it is called the **likelihood** of the parameters. For the family of optimisation algorithms based on **gradient descent**, we need to know the **gradient** (or **derivative**) of the likelihood with respect to the model parameters. The basic idea is to update each parameter in the direction that increases the likelihood. The gradient of the likelihood with respect to a parameter θ is given in (5.15), which decomposes into four terms: the first two are for the prior distribution over situations, and the last two are for the semantic functions. In both cases, one term is positive and conditioned on the data, while the other term is negative and represents the predictions of the model – the model **converges** (the parameter updates go to zero) when the predictions of the model exactly match the data, so the two terms cancel out.

In (5.15), s is a latent situation and g is an observed DMRS graph, (corresponding to the top and bottom rows of Fig. 5.1, respectively). The gradient is of the *log*-likelihood, for mathematical convenience, because multiplying probabilities corresponds to summing log-probabilities, and it's easier to differentiate a sum than a product. Maximising the log-likelihood is equivalent to maximising the likelihood. In the last two lines, the sum is over the pixies x in the latent situation, and r is the observed predicate corresponding to x (which could also be written as $r^{(X)}$

to indicate this). In the last line, r' denotes a predicate generated from the latent pixie x, and g' denotes a DMRS graph generated from the latent situation s – both of these are supposing we fix the situation s, and imagine we haven't observed the predicates. The subscripts on the expectations denote which random variable is marginalised out, and which random variables are conditioned on (if any). A full derivation is given in §5.2.1 below.

$$\begin{aligned}\frac{\partial}{\partial\theta}\log\mathbb{P}\left(g\right) = \;& \mathbb{E}_{s|g}\left[\frac{\partial}{\partial\theta}\left(-E(s)\right)\right] \\ & -\mathbb{E}_{s}\left[\frac{\partial}{\partial\theta}\left(-E(s)\right)\right] \\ & +\mathbb{E}_{s|g}\left[\sum_{x\text{ in }s}\left(1-t^{(r)}(x)\right)\frac{\partial}{\partial\theta}\left(-F(x,r)\right)\right] \\ & -\mathbb{E}_{s|g}\left[\mathbb{E}_{g'|s}\left[\sum_{x\text{ in }s}\left(1-t^{(r')}(x)\right)\frac{\partial}{\partial\theta}\left(-F(x,r')\right)\right]\right]\end{aligned} \tag{5.15}$$

We can now expand (5.15) for each of the parameter tensors, as shown in (5.16) to (5.19). For the semantic role parameters $w_{ij}^{(l)}$ and b_i, we only get contributions from the first two terms of (5.15), while for the semantic function parameters $v_i^{(r)}$ and $a^{(r)}$, we only get contributions from the last two terms of (5.15). The gradient for $w_{ij}^{(l)}$ is given in (5.16), showing that we reinforce connections between units we expect to be active for the observed graph, and we weaken connections between units that we expect to be active in general (not conditioned on the observed graph). The gradient goes to zero when each RBM connection is used equally often when explaining the data and when generating new situations. The gradient for b_i is given in (5.17), where we similarly weaken bias against units we expect to be active for the observed graph, and we strengthen bias against units we expect to be active in general.

$$\frac{\partial}{\partial w_{ij}^{(l)}}\log\mathbb{P}\left(g\right) = \left(\mathbb{E}_{s|g}-\mathbb{E}_{s}\right)\left[\sum_{x\xrightarrow{l}y\text{ in }s}x_i y_j\right] \tag{5.16}$$

$$\frac{\partial}{\partial b_i}\log\mathbb{P}\left(g\right) = \left(\mathbb{E}_{s|g}-\mathbb{E}_{s}\right)\left[-\sum_{x\text{ in }s}x_i\right] \tag{5.17}$$

The gradient for $v_i^{(r)}$ is given in (5.18), where, for the observed predicate, we reinforce weights for the units we expect to be active – and if other predicates are likely to be generated for this pixie, we weaken their weights. The different behaviour for the observed predicate and other predicates is indicated by $\mathbb{1}_{r'=r}$, which is 1 when $r'=r$, and 0 otherwise. The gradient goes to zero when the predicted predicates for latent situations match the observed predicates. Finally, the gradient for $a^{(r)}$ is given in (5.18), where we similarly weaken the bias against the

observed predicate, and strengthen the bias against other likely predicates.

$$\frac{\partial}{\partial v_i^{(r')}} \log \mathbb{P}(g) = \mathbb{E}_{s|g}\left[\sum_{x \text{ in } s}\left(\mathbb{1}_{r'=r} - \mathbb{P}(r' \mid x)\right)\left(\left(1-t^{(r')}(x)\right)x_i\right)\right] \tag{5.18}$$

$$\frac{\partial}{\partial a^{(r')}} \log \mathbb{P}(g) = \mathbb{E}_{s|g}\left[\sum_{x \text{ in } s}\left(\mathbb{1}_{r'=r} - \mathbb{P}(r' \mid x)\right)\left(-\left(1-t^{(r')}(x)\right)\right)\right] \tag{5.19}$$

Calculating the above expectations exactly is infeasible, as this requires summing over all possible situations, and for high-dimensional spaces, there are simply too many to sum over. In the following two sections, I introduce approximate calculations for these expectations: in §5.3, I introduce a Markov Chain Monte Carlo method, which approximates the expectation by summing over a number of samples, and in §5.4, I introduce a Variational Inference method, which approximates the distribution we are summing with respect to.

5.2.1 Derivation of Gradient

In this section, I derive (5.15). We are aiming to optimise the log-likelihood $\log \mathbb{P}(g)$, with respect to the model parameters. The important idea that will lead us to the relatively intuitive form of (5.15) is that the model is a combination of two distributions in the **exponential family** – probabilities are proportional to the exponential of a negative energy. Both the prior distribution over situations, and the conditional distribution over predicates given a situation, can be expressed in this form.[4] This means that each half of the model gives a pair of terms in the gradient: one positive, conditioned on the data; and one negative, generated by the model.

First, we apply the chain rule to $\log \mathbb{P}(g)$. Although using a log simplifies the gradient, the simplification is not immediate, because the model generates the graph based on the latent situation, so calculating $\log \mathbb{P}(g)$ requires summing over all s. We then expand $\mathbb{P}(s, g)$ into a product of terms. Here, each pixie x in s corresponds to a predicate r in g. Recall that $t^{(r)}$ denotes the semantic function for r, and $f^{(r)}$ denotes the frequency of r, which is a constant.

$$\begin{aligned}
\frac{\partial}{\partial \theta} \log \mathbb{P}(g) &= \frac{1}{\mathbb{P}(g)} \frac{\partial}{\partial \theta} \mathbb{P}(g) \\
&= \frac{1}{\mathbb{P}(g)} \frac{\partial}{\partial \theta} \sum_s \mathbb{P}(s, g) \\
&= \frac{1}{\mathbb{P}(g)} \frac{\partial}{\partial \theta} \sum_s \frac{1}{Z} \exp\left(-E(s)\right) \prod_{x \text{ in } s} \frac{1}{Z(x)} f^{(r)} t^{(r)}(x)
\end{aligned}$$

[4] It is tempting to try to reparametrise the model so the whole thing is in the exponential family. Indeed, directed and undirected graphical models are equivalent – given a model of one form, there is a model of the other form defining the same distribution. However, I have introduced a *family* of graphical models with different situation structures, but with shared parameters. Conversion to an undirected graph loses parameter sharing between semantic functions, and conversion to a directed graph loses parameter sharing between semantic roles.

As the summand is product of several terms, applying the product rule will give a sum of several terms. However, we can make use of the fact that all these terms are exponentials, so their derivatives of are multiples of the original term:

$$\begin{aligned}
\frac{\partial}{\partial\theta}e^{-E(s)} &= e^{-E(s)}\frac{\partial}{\partial\theta}\left(-E(s)\right)\\
\frac{\partial}{\partial\theta}t^{(r)}(x) &= t^{(r)}(x)\left(1-t^{(r)}(x)\right)\frac{\partial}{\partial\theta}\left(-F(x,r)\right)\\
\frac{\partial}{\partial\theta}\frac{1}{Z(x)} &= -\frac{1}{Z(x)^2}\frac{\partial}{\partial\theta}Z(x)\\
\frac{\partial}{\partial\theta}\frac{1}{Z} &= -\frac{1}{Z^2}\frac{\partial}{\partial\theta}Z
\end{aligned}$$

This allows us to derive:

$$\begin{aligned}
\frac{\partial}{\partial\theta}\log\mathbb{P}\left(g\right) = \frac{1}{\mathbb{P}\left(g\right)}\sum_{s}\mathbb{P}\left(s,g\right)\Bigg[&\frac{\partial}{\partial\theta}\left(-E(s)\right)\\
&+\sum_{x\text{ in }s}\left(1-t^{(r)}(x)\right)\frac{\partial}{\partial\theta}\left(-F(x,r)\right)\\
&-\sum_{x\text{ in }s}\frac{1}{Z(x)}\frac{\partial}{\partial\theta}Z(x)\\
&-\frac{1}{Z}\frac{\partial}{\partial\theta}Z\Bigg]
\end{aligned}$$

The final summand does not depend on s, so the sum simplifies: $\sum_s \mathbb{P}\left(s,g\right) = \mathbb{P}\left(g\right)$, which cancels with $1/\mathbb{P}\left(g\right)$. For the other three terms, we can simplify using $\mathbb{P}\left(s,g\right)/\mathbb{P}\left(g\right) = \mathbb{P}\left(s\,|\,g\right)$. Then:

$$\begin{aligned}
\frac{\partial}{\partial\theta}\log\mathbb{P}\left(g\right) = \sum_{s}\mathbb{P}\left(s\,|\,g\right)\Bigg[&\frac{\partial}{\partial\theta}\left(-E(s)\right)\\
&+\sum_{x\text{ in }s}\left(1-t^{(r)}(x)\right)\frac{\partial}{\partial\theta}\left(-F(x,r)\right)\\
&-\sum_{x\text{ in }s}\frac{1}{Z(x)}\frac{\partial}{\partial\theta}Z(x)\Bigg]\\
&-\frac{1}{Z}\frac{\partial}{\partial\theta}Z
\end{aligned}$$

We now expand the derivatives of the normalisation constants. As these are sums of the first

two terms, they give analogous derivatives, except summed over all values:

$$\begin{aligned}\frac{1}{Z}\frac{\partial}{\partial\theta}Z &= \frac{1}{Z}\frac{\partial}{\partial\theta}\sum_{x}\exp\big(-E(x)\big)\\ &= \sum_{s}\frac{\exp\big(-E(x)\big)}{Z}\frac{\partial}{\partial\theta}\big(-E(s)\big)\\ &= \sum_{s}\mathbb{P}\left(s\right)\frac{\partial}{\partial\theta}\big(-E(s)\big)\end{aligned}$$

$$\begin{aligned}\frac{1}{Z(x)}\frac{\partial}{\partial\theta}Z(x) &= \frac{1}{Z(x)}\frac{\partial}{\partial\theta}\sum_{r'}f^{(r')}t^{(r')}(x)\\ &= \sum_{r'}\frac{f^{(r')}t^{(r')}(x)}{Z(x)}\big(1-t^{(r')}(x)\big)\frac{\partial}{\partial\theta}\big(-F(x,r')\big)\\ &= \sum_{r'}\mathbb{P}\left(r'\mid x\right)\big(1-t^{(r')}(x)\big)\frac{\partial}{\partial\theta}\big(-F(x,r')\big)\end{aligned}$$

Putting this all together, we get:

$$\begin{aligned}\frac{\partial}{\partial\theta}\log\mathbb{P}\left(g\right) = \sum_{s}\mathbb{P}\left(s\mid g\right)\Bigg[&\frac{\partial}{\partial\theta}\big(-E(s)\big)\\ &+\sum_{x\text{ in }s}\big(1-t^{(r)}(x)\big)\frac{\partial}{\partial\theta}\big(-F(x,r)\big)\\ &-\sum_{x\text{ in }s}\sum_{r'}\mathbb{P}\left(r'\mid x\right)\big(1-t^{(r')}(x)\big)\frac{\partial}{\partial\theta}\big(-F(x,r')\big)\Bigg]\\ -\sum_{s}\mathbb{P}\left(s\right)\frac{\partial}{\partial\theta}\big(-E(s)\big)&\end{aligned}$$

Finally, we write expectations instead of sums of probabilities:

$$\begin{aligned}\frac{\partial}{\partial\theta}\log\mathbb{P}\left(g\right) = \mathbb{E}_{s|g}\Bigg[&\frac{\partial}{\partial\theta}\big(-E(s)\big)\\ &+\sum_{x\text{ in }s}\big(1-t^{(r)}(x)\big)\frac{\partial}{\partial\theta}\big(-F(x,r)\big)\\ &-\sum_{x\text{ in }s}\mathbb{E}_{r'|x}\left[\big(1-t^{(r')}(x)\big)\frac{\partial}{\partial\theta}\big(-F(x,r')\big)\right]\Bigg]\\ -\mathbb{E}_{s}\left[\frac{\partial}{\partial\theta}\big(-E(s)\big)\right]&\end{aligned}$$

5.3 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) methods approximate expectations by sampling a small number of points, that should be somehow representative of the whole distribution. For example, this kind of inference is often used for LDA (Griffiths and Steyvers, 2004) and for RBMs (Hinton, 2002, 2010). For the random variable that we need to marginalise out, we start from some random initial value, and define a probabilistic update rule. This rule is chosen so that, as we keep applying it, the distribution of values tends towards the distribution we want. After applying the rule many times, we can use the resulting value as a sample. The updates can be seen as defining a **Markov chain** (the distribution for the next value only depends on the previous value) and the sampling as a **Monte Carlo** method (we take samples to approximate the full distribution) hence the name.

In (5.15), there are expectations with respect to three different distributions, and so we need to sample values for each of these three distributions. For the first, third, and fourth terms, we need to sample a situation conditioned on the observed DMRS graph, while for the second term, we need to sample a situation without conditioning on the data. The sampling of situations both with and without conditioning on the data is analogous to the training of normal RBMs. For the fourth term, we also need to sample a predicate for each sampled pixie, which we use to approximate the fourth term – this sampled predicate is analogous to the negative samples used by Mikolov et al. (2013) in a Skip-gram model.

For all three sets of samples, we can initialise values randomly, but we need to know update rules so that the sampled values tend towards the correct distributions. As we will see, the form of the model means that sampling is more difficult than for LDA or normal RBMs.

The prior distribution over situations is the easiest to calculate, because we can exactly calculate the conditional distribution of one pixie, given the other pixies in the situation. This is by virtue of the design of a CaRBM – Swersky et al. (2012) show they can be calculated using **belief propagation**[5] (for an introduction, see: Yedidia et al., 2003). Intuitively, if we didn't have the cardinality constraint, we could switch each unit on or off independently. With the constraint, we can first consider units independently, and then rule out cases where the total number of active units is wrong. This can be done efficiently, because we can recursively calculate how many units are active for the first i units, and build up to the whole vector.

We first fix an (arbitrary) order of the units, and calculate the probabilities p_i that each unit is on, ignoring the cardinality constraint, as shown in (5.20). The probability for each unit depends on its connections to other pixies in the situation, as well as its bias.

$$p_i = \sigma\left(\sum_{y \xrightarrow{l} x} w_{ji}^{(l)} y_j + \sum_{y \xleftarrow{l} x} w_{ij}^{(l)} y_j - b_i\right) \tag{5.20}$$

[5] Also known as the "sum-product algorithm".

We can then sample a pixie using two passes over the units. In the first pass, we calculate the probabilities $m(i, n)$ that, out of the first i units, exactly n of them are active, supposing we don't have a cardinality constraint, as shown in (5.21). These are calculated recursively over i, starting from $m(1, 1) = p_1$ and $m(1, 0) = 1-p_1$. We can let n range from 0 to C, discarding the probability mass making the total is above C. We know that exactly C must be active overall, so once we have finished the first pass, we can discard all of the probability mass making the total below C. In the second pass, we go back along the units, probabilistically deciding whether to switch each unit on or off. At each unit i, we know how many more units need to be on – let this number be k_i. For the last unit, we simply have $k_D = C$. We can find the probability that the current unit is on, using the unnormalised probabilities calculated in the first pass – in particular, the probabilities that, out of the remaining units, exactly k_i or k_i-1 are on, as shown in (5.22) and (5.23). After probabilistically choosing a value, we can calculate k_{i-1} based on k_i, subtracting 1 if we turned unit i on.

$$m(i, n) = p_i\, m(i-1, n-1) + (1-p_i)\, m(i-1, n) \tag{5.21}$$

$$\mathbb{P}\,(x_i=1) \propto m(i-1, k_i-1) \tag{5.22}$$

$$\mathbb{P}\,(x_i=0) \propto m(i-1, k_i) \tag{5.23}$$

I now turn to the second distribution we need to sample from, the distribution over situations s conditioned on an observed DMRS graph g. We cannot calculate the conditional distribution $\mathbb{P}\,(s \mid g)$, and unlike for the prior $\mathbb{P}\,(s)$, we also cannot calculate the conditional distribution for each pixie node, given the graph and all other pixies. For the prior, it was possible to use belief propagation, but this relied on decomposing the cardinality constraint into a series of recursive steps. This is unfortunately not possible when conditioning on a predicate. I will denote the conditional distribution for one pixie node as $\mathbb{P}\,(x \mid g, s_{-X})$, where s_{-X} represents the values for all other pixie nodes in the situation.

However, if we compare two particular values x and x', for a single latent pixie variable, the normalisation constant cancels out in the ratio $\mathbb{P}\,(x' \mid g, s_{-X}) \,/\, \mathbb{P}\,(x \mid g, s_{-X})$, so we can use the **Metropolis-Hastings** algorithm (Metropolis et al., 1953; Hastings, 1970). Each update step involves first considering switching to a different value, and then probabilistically deciding to either switch to the new value or stay with the current value. For sampling a sparse binary-valued vector, given the current value x, we can uniformly at random choose one unit to switch on, and one to switch off, to get a proposal x'. If the ratio of probabilities shown in (5.24) is above 1, we automatically switch to x'; while if the ratio is below 1, it is used as the probability of switching to x'. Here, r is the observed predicate corresponding to the latent pixie.

$$\frac{\mathbb{P}\,(x' \mid g, s_{-X})}{\mathbb{P}\,(x \mid g, s_{-X})} = \frac{\exp\big(-E(s')\big)\,\frac{1}{Z(x')}t^{(r)}(x')}{\exp\big(-E(s)\big)\,\frac{1}{Z(x)}t^{(r)}(x)} \tag{5.24}$$

Although Metropolis-Hastings avoids the need to calculate the normalisation constant Z of the prior distribution, the model was defined with two normalisation steps (unlike most applications of the algorithm, where there is just one) and so we still have the normalisation constant $Z(x)$ for choosing a predicate given a pixie. This constant represents the proportion of the vocabulary that is true of the pixie (weighting predicates by frequency). Intuitively, we prefer to sample a pixie which few predicates are true of, rather than a pixie which many predicates are true of. Calculating this requires summing over all the whole vocabulary, which is expensive.

To avoid this problem, we can introduce an approximation. Rather than averaging the outputs of the semantic functions, we can instead average the functions themselves. This means that we can calculate a single average, which we can use for all pixies, rather than having to calculate a different average for each pixie. Weighting predicates by frequency, we can calculate the average predicate weights $\overline{v}_i$ and biases $\overline{b}$, which define an average semantic function $\overline{t}$. One option for approximating the ratio of normalisation constants is to apply this average semantic function, as shown in (5.25). However, we have no guarantee that this is a good approximation. An alternative is to consider the average predicate weights for the dimensions that are being turned on and off. Intuitively, if the average weight is higher for one dimension, more predicates may be true when that dimension is active. A heuristic approximation is given in (5.26), where x and x' differ in dimensions i and i' only: x_i and $x'_{i'}$ are on, while $x_{i'}$ and x'_i are off. Using an exponential means that we can simply add this term to the energy. The constant k can be freely chosen to improve the accuracy of this approximation.

$$\frac{Z(x)}{Z(x')} \approx \frac{\overline{t}(x)}{\overline{t}(x')} \tag{5.25}$$

OR

$$\frac{Z(x)}{Z(x')} \approx \exp\left(k\left(\overline{v}_i - \overline{v}_{i'}\right)\right) \tag{5.26}$$

Finally, we have the distribution over predicates, given a latent pixie. This can be done straightforwardly using the Metropolis-Hastings algorithm, according to the ratio shown in (5.27), where x is the value of the latent pixie, r is the current value of the sampled predicate, and r' is the proposed new value of the sampled predicate.

$$\frac{\mathbb{P}\left(r' \mid x\right)}{\mathbb{P}\left(r \mid x\right)} = \frac{f^{(r')}t^{(r')}(x)}{f^{(r)}t^{(r)}(x)} \tag{5.27}$$

One optimisation is worth noting. After each update to the model parameters, all of the above distributions will change. Rather than recalculating the samples from scratch after each parameter update, we can use **fantasy particles** (also known as **persistent** Markov chains), which Tieleman (2008) found effective for training RBMs. This simply involves keeping the value from before the parameter update, and using this as the initial value after the update.

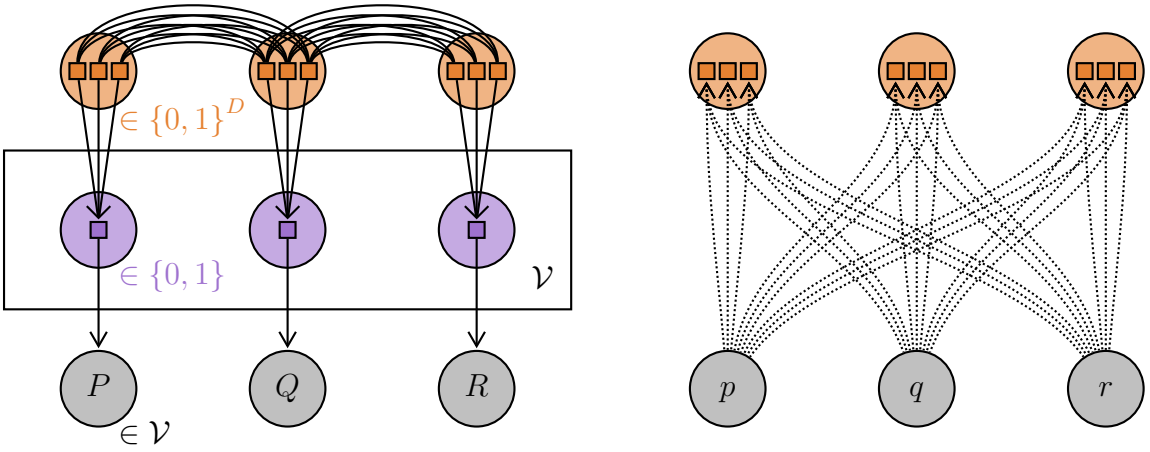


Figure 5.2: The diagram on the right depicts a mean-field approximation, where each unit has an independent probability of being active. These probabilities are optimised so that the joint distribution approximates the conditional distribution of the pixie nodes in the graphical model on the left (repeated from Fig. 5.1), given the observations $P = p$, $Q = q$, and $R = r$. We are not interested in the latent truth values, so they are not explicitly represented.

5.4 Variational Inference

The MCMC algorithms described in the previous section guarantee unbiased estimates once the Markov chain has converged (with the exception of the approximation for $Z(x)$). However, an MCMC algorithm is slow for two reasons. Firstly, many iterations of the Markov chain are required for it to converge. Secondly, even though we are not summing over the entire space, many samples are still needed, because the discrete values lead to high variance. In this section, I introduce a **variational inference** algorithm, which directly approximates the distribution we need to calculate, and then optimises this approximation.

The distribution that is difficult to sample from is $\mathbb{P}(x \mid g, s_{-X})$, the conditional distribution of a pixie node, given the observed DMRS graph g and all other latent pixies s_{-X}. The basic idea is to introduce a simpler distribution $\mathbb{Q}(x)$, and then optimise the parameters for this simpler distribution. In particular, we can use a **mean-field** approximation, where each unit has an independent probability q_i of being active, as shown in (5.28). This is simpler than the true distribution, because each unit is independent. Furthermore, we will optimise each of these probabilities based on the average activation of all other units (in other words, the *mean-field* activation).

$$\mathbb{P}(x \mid g, s_{-X}) \approx \mathbb{Q}(x) = \prod_{i|x_i=1} q_i \prod_{i|x_i=0} (1 - q_i) \tag{5.28}$$

The $\mathbb{Q}$ distribution therefore has one parameter for each unit, and the values of these parameters are jointly optimised across all pixies, so that the overall distribution is close to the true distribution $\mathbb{P}$. Because the parameters are jointly optimised, each parameter depends on all the observed pixies. This is illustrated in Fig. 5.2.

For $\mathbb{Q}$ to be a good approximation, it needs to be close to $\mathbb{P}$. We can measure this using the **Kullback-Leibler (KL) divergence** from $\mathbb{Q}$ to $\mathbb{P}$, which measures the number of extra bits we need to encode a sample from $\mathbb{P}$ using a code designed for $\mathbb{Q}$. Minimising this quantity is also done in the **Expectation Propagation** algorithm (Minka, 2001). However, a semantic function model is not in the exponential family, which means we cannot apply Expectation Propagation. In contrast, **Variational Bayes** minimises the KL-divergence in the opposite direction, from $\mathbb{P}$ to $\mathbb{Q}$. However, for the above approximation, this is infinite: for any pixie where the number of active units is not equal to the fixed cardinality, $\mathbb{P}(x) = 0$ but $\mathbb{Q}(x) \neq 0$, giving infinite $\mathbb{Q}(x) \log \mathbb{P}(x \mid g, s_{-X})$. Furthermore, while Variational Bayes prefers "high precision" approximations (areas of high $\mathbb{Q}$ are accurate), optimising the opposite KL-divergence leads to "high recall" approximations (areas of high $\mathbb{P}$ are accurate). This is appropriate for two reasons. Firstly, one way that $\mathbb{Q}$ has low precision is predicting pixies with the wrong number of active units. However, we can avoid these areas by using the belief propagation algorithm explained in the previous section – we simply replace p_i with q_i in (5.20). Secondly, in areas where the number of active units is correct, $\mathbb{Q}$ will be much higher than $\mathbb{P}$ only if there is a dependence between dimensions that $\mathbb{Q}$ cannot capture, such as if $\mathbb{P}$ has multiple local maxima. Because of the definition of an RBM, such a dependence is impossible within one pixie. Between pixies, this kind of dependence is possible, as was discussed in §4.3, but it is reasonable to expect such cases to be rare, since they need a fine balance between the prior over situations and the semantic functions.

To optimise $\mathbb{Q}$, we can use gradient descent on the parameters q_i. For simplicity, we can begin by considering a situation composed of a single pixie x, with an observed predicate r. Exactly calculating the gradient turns out to be difficult, and additional approximations are necessary. In particular, given a mean-field distribution, what is the expected output of a semantic function applied to a vector drawn from the distribution? The mean field vector is not in $\mathcal{X}$, because each component lies in the range $[0, 1]$, rather than the binary set $\{0, 1\}$. Each value represents how much we expect the pixie to have a particular feature, given the observed predicates. However, we can still apply semantic functions to these mean-field vectors, since they have been implemented as feedforward neural nets – each parameter in a neural net can be multiplied by a value in the range $[0, 1]$ just as easily as it can be multiplied by 0 or 1. Since a mean-field vector defines a distribution over pixies, applying a semantic function to a mean-field vector lets us approximately calculate the probability that a predicate is true of a pixie drawn from this distribution.

Differentiating the KL-divergence with respect to q_i, and using the above idea that we can apply semantic functions to mean-field vectors, we can derive the update rule given in (5.29). A full derivation is given in §5.4.3. This updates $\mathbb{Q}$ one parameter q_i at a time, while holding the other parameters fixed. The rule looks at the probability of generating the predicate r when the unit x_i is on, and when it is off. If r is more likely to be generated when x_i is on, q_i will

be high. If r is more likely to be generated when x_i is off, q_i will be low. If there is no difference, q_i will be C/D, the expected probability if all dimensions are equally likely. I have written $x^{(+i)}$ for the mean-field vector where unit i is fixed to be on, and $x^{(-i)}$ for the mean-field vector where unit i is fixed to be off. Details are given in §5.4.3. Optimising $\mathbb{Q}$ can then be done by repeatedly applying this update rule across all dimensions. To approximate the ratio of normalisation constants $Z(x)$, we can use (5.25).

$$q_i = \left(1 + \frac{D-C}{C} \frac{t^{(r)}\left(x^{(-i)}\right)}{t^{(r)}\left(x^{(+i)}\right)} \frac{Z\left(x^{(+i)}\right)}{Z\left(x^{(-i)}\right)}\right)^{-1} \tag{5.29}$$

For multiple pixies, as shown in Fig. 5.2, the process is similar. We have one mean-field vector for each pixie node, and we optimise these together. The only difference to the update rule is that, as well as considering how activating one unit changes the probability of a predicate being generated, we also have to consider how likely this dimension is to be active, given the other pixies in the situation. This leads to an extra term in the update rule, as shown in (5.30), where we sum over incoming links $y \xrightarrow{l} x$ and outgoing links $y \xleftarrow{l} x$. This rule also includes the bias terms, which were neglected in the update rule above.

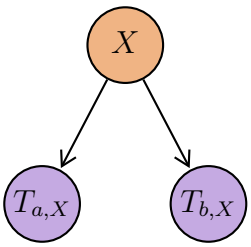
$$q_i = \left(1 + \frac{D-C}{C} \frac{t^{(r)}\left(x^{(-i)}\right)}{t^{(r)}\left(x^{(+i)}\right)} \frac{Z\left(x^{(+i)}\right)}{Z\left(x^{(-i)}\right)} \exp\left(b_i - \sum_{y \xrightarrow{l} x} w_{ji}^{(l)} y_j - \sum_{y \xleftarrow{l} x} w_{ij}^{(l)} y_j\right)\right)^{-1} \tag{5.30}$$

Intuitively, we assign high probabilities to a unit for two possible reasons: either it's strongly connected to highly probable units in other pixies, or activating this unit makes it much more likely for an observed predicate to be generated. If neither of these facts hold, we will assign a low probability – because we are enforcing sparsity on the pixie vectors, the dimensions are effectively competing with each other.

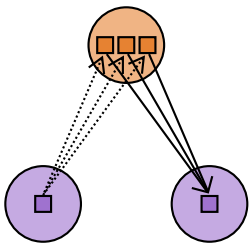
5.4.1 Variational Inference for Context Dependence

In §4.2, I proposed representing the occasion meaning of a predicate as the posterior distribution of its corresponding pixie node. Because the above mean-field vectors approximate posterior distributions of pixie nodes, we can see a mean-field vector as an approximate occasion meaning. Indeed, the right-hand side of Fig. 5.2 clearly shows how each mean-field vector depends on the whole context.

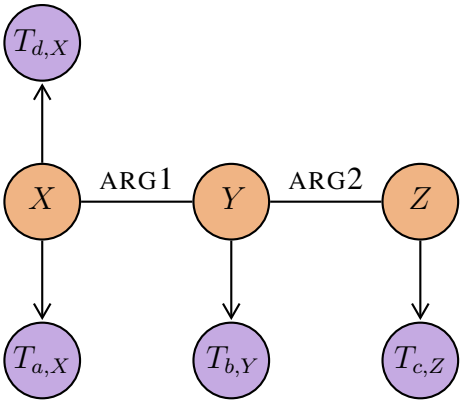
The update rule given in (5.30) applies specifically to linguistic context (observing a DMRS graph), but the general principle behind it could be applied to other kinds of context. As previously argued, a probabilistic graphical model can easily be extended to include extralinguistic nodes, and variational inference can similarly be extended to approximate other kinds of posterior distribution, and hence other kinds of occasion meaning.



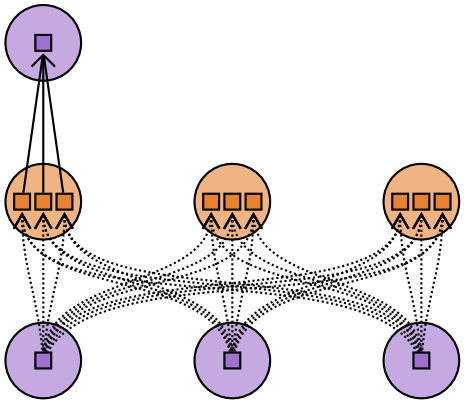
(a) Logical inference for a single pixie X, repeated from Fig. 4.2. We would like to know if b is true of X, given that a is true of X, which can be cast as the conditional probability $\mathbb{P}(t_{b,X} \mid t_{a,X})$.



(b) Variational inference for Fig. 5.3a. The dotted lines indicate mean-field variational inference, and the solid lines indicate approximate inference using the mean-field pixie vector.



(c) Logical inference for three pixies, repeated from Fig. 4.4. We would like to know if d is true of X, given that a, b, c are true of X, Y, Z, respectively, which can be cast as the conditional probability $\mathbb{P}(t_{d,X} \mid t_{a,X}, t_{b,Y}, t_{c,Z})$.



(d) Variational inference for Fig. 5.3c. The ARG1 and ARG2 roles are not represented, but the mean-field probabilities are optimised to approximate the joint distribution induced by the links, and so each unit depends on all observed truth values.

Figure 5.3: Each variational distribution on the right allows us to approximately calculate a conditional probability (which represents a logical inference) defined by the graph on the left.

5.4.2 Variational Inference for Logical Inference

In §4.5, I cast logical inference as Bayesian inference between truth value nodes. The above variational inference algorithm now gives us an efficient way to calculate these conditional probabilities, as illustrated in Fig. 5.3. The only difference with the previous discussion is that we are now conditioning on a predicate being true, rather than a predicate being generated. This changes just one detail in the derivation in §5.4.3 – we can drop the $Z(x)$ terms, because we are just considering the truth of one predicate, rather than choosing to generate a predicate from the vocabulary. This means that conditioning on truth is actually slightly easier than conditioning on observing a predicate, because we don't need to approximate $Z(x)$.

To use the mean-field approximation for logical inference, we first find the mean-field vectors for all pixie nodes in the situation, conditioning on the observed truth values. Note that we have to construct a vector for *every* pixie node, because they are *jointly* optimised to approximate the joint posterior distribution for the pixie nodes. We then take the semantic function for predicate of interest, and apply it to the relevant mean-field pixie.

5.4.3 Derivation of Update Rule

In this section, I derive (5.29) and (5.30). A few additional approximations are necessary, beyond the mean-field approximation, which will be introduced in the course of the derivation. We are trying to optimise $\mathbb{Q}$ to minimise the KL-divergence from $\mathbb{Q}(x)$ to $\mathbb{P}(x \mid g, s_{-X})$:

$$\begin{aligned} D_{\mathrm{KL}}(\mathbb{P}||\mathbb{Q}) &= \sum_x \mathbb{P}(x \mid g, s_{-X}) \log \frac{\mathbb{P}(x \mid g, s_{-X})}{\mathbb{Q}(x)} \\ &= \sum_x \mathbb{P}(x \mid g, s_{-X}) \left(\log \mathbb{P}(x \mid g, s_{-X}) - \left(\sum_{j|x_j=1} \log(q_j) + \sum_{j|x_j=0} \log(1-q_j) \right) \right) \end{aligned}$$

To optimise $\mathbb{Q}$, we take the derivative with respect to a parameter q_i. Note that the first term above is independent of $\mathbb{Q}$, which gives us the first line below. The independence assumption of $\mathbb{Q}$ also means that $\partial q_j / \partial q_i = 0$ for all $j \neq i$. This gives us the second line below, where the sum over x has been split according to whether x_i is on or off.

$$\begin{aligned} \frac{\partial}{\partial q_i} D_{\mathrm{KL}}(\mathbb{P}||\mathbb{Q}) &= -\frac{\partial}{\partial q_i} \sum_x \mathbb{P}(x \mid g, s_{-X}) \left(\sum_{j|x_j=1} \log(q_j) + \sum_{j|x_j=0} \log(1-q_j) \right) \\ &= -\sum_{x|x_i=1} \mathbb{P}(x \mid g, s_{-X}) \frac{1}{q_i} + \sum_{x|x_i=0} \mathbb{P}(x \mid g, s_{-X}) \frac{1}{1-q_i} \end{aligned}$$

Now we can rewrite $\mathbb{P}(x \mid g, s_{-X})$ as the following. Let r be the observed predicate corresponding to x. For simplicity, we will first assume that r is the only predicate in g, and so x is the only pixie in s. We will also assume a uniform prior over x, which is equivalent to neglecting the bias terms b_i. For D dimensions, of which C are active, there are $\binom{D}{C}$ different vectors, which are each equally likely.

$$\begin{aligned} \mathbb{P}(x \mid g, s_{-X}) = \mathbb{P}(x \mid r) &= \frac{\mathbb{P}(x)\,\mathbb{P}(r \mid x)}{\mathbb{P}(r)} \\ &= \frac{f^{(r)} t^{(r)}(x)}{\binom{D}{C} \mathbb{P}(r)\, Z(x)} \end{aligned}$$

Note that $f^{(r)} / \binom{D}{C} \mathbb{P}(r)$ is constant in x, so the only part that depends on x is $t^{(r)}(x)/Z(x)$. Setting the derivative of the KL-divergence to 0, and substituting in above the expression for $\mathbb{P}(x \mid g, s_{-X})$, the parts constant in x cancel and we are left with:

$$\sum_{x|x_i=1} \frac{t^{(r)}(x)}{Z(x)} \frac{1}{q_i} = \sum_{x|x_i=0} \frac{t^{(r)}(x)}{Z(x)} \frac{1}{1-q_i}$$

It is intractable to sum over all these x, but now we can use the mean-field approximation.

Let $x^{(+i)}$ denote the mean-field vector where x_i is on and $C-1$ other units are on, and let $x^{(-i)}$ denote the mean-field vector where x_i is off, and C other units are on. We can define these mean-field vectors so that the value for each dimension $j \neq i$ is the marginal distribution for that unit, given the value for x_i, the distribution $\mathbb{Q}$, and the cardinality constraint. This can be calculated using the belief propagation algorithm explained in §5.3, modified so that we record each marginal probability, rather than sampling a value. The sums k_i are now real numbers, recursively calculated as $k_{i-1} = k_i - x_i$. However, this is prohibitively expensive when making many updates. A cheaper alternative is to linearly scale $\mathbb{Q}$ so that the sum of the components is correct (but preventing values from going above 1). For $j \neq i$, we have:

$$x_j^{(+i)} = \min\left\{1,\ \frac{C-1}{\sum_{k\neq i} q_k} q_j\right\}$$
$$x_j^{(-i)} = \min\left\{1,\ \frac{C}{\sum_{k\neq i} q_k} q_j\right\}$$

Now we can use the second approximation mentioned in the main text above – rather than applying $t^{(r)}$ to many values of x, we can apply it to the mean-field vector, and simply count how many different values of x would have had to consider. The approximations for $Z(x)$ that was given in (5.25) can similarly be used for mean-field vectors. We then have:

$$\binom{D-1}{C-1} \frac{t^{(r)}\left(x^{(+i)}\right)}{Z\left(x^{(+i)}\right)} \frac{1}{q_i} \approx \binom{D-1}{C} \frac{t^{(r)}\left(x^{(-i)}\right)}{Z\left(x^{(-i)}\right)} \frac{1}{1-q_i}$$
$$\frac{t^{(r)}\left(x^{(+i)}\right)}{Z\left(x^{(+i)}\right)} \frac{1}{q_i} \approx \frac{D-C}{C} \frac{t^{(r)}\left(x^{(-i)}\right)}{Z\left(x^{(-i)}\right)} \frac{1}{1-q_i}$$

Re-arranging for q_i yields the following, which is the optimal value for q_i, given the other dimensions q_j, and given the above approximations:

$$q_i \approx \left(1 + \frac{D-C}{C} \frac{t^{(r)}\left(x^{(-i)}\right)}{t^{(r)}\left(x^{(+i)}\right)} \frac{Z\left(x^{(+i)}\right)}{Z\left(x^{(-i)}\right)}\right)^{-1}$$

In the above derivation, we assumed a uniform prior over x, which meant that we had $\mathbb{P}(x \mid g, s_{-X}) \propto t^{(r)}(x)/Z(x)$. If there are bias terms, or if there links between pixies, then this no longer holds, and we instead have the prior $\mathbb{P}(x)$ being determined by the RBM parameters,

which gives the following, where we sum over incoming links $y \xrightarrow{l} x$ and outgoing links $y \xleftarrow{l} x$.

$$\begin{aligned}\mathbb{P}\left(x \mid g, s_{-X}\right) &\propto \frac{t^{(r)}(x)}{Z(x)} \exp\left(\sum_{y \xrightarrow{l} x} w_{kj}^{(l)} x_j y_k + \sum_{y \xleftarrow{l} x} w_{jk}^{(l)} x_j y_k - b_j x_j\right) \\ &= \frac{t^{(r)}(x)}{Z(x)} \exp\left(\left(\sum_{y \xrightarrow{l} x} w_{kj}^{(l)} y_k + \sum_{y \xleftarrow{l} x} w_{jk}^{(l)} y_k - b_j\right) x_j\right)\end{aligned}$$

So to amend the update rule, we replace $t^{(r)}(x)/Z(x)$ with the above expression, which gives the following:

$$q_i \approx \left(1 + \frac{D-C}{C} \frac{t^{(r)}\left(x^{(-i)}\right) Z\left(x^{(+i)}\right) \exp\left(\left(\sum_{y \xrightarrow{l} x} w_{kj}^{(l)} y_k + \sum_{y \xleftarrow{l} x} w_{jk}^{(l)} y_k + b_j\right) x_j^{(-i)}\right)}{t^{(r)}\left(x^{(+i)}\right) Z\left(x^{(-i)}\right) \exp\left(\left(\sum_{y \xrightarrow{l} x} w_{kj}^{(l)} y_k + \sum_{y \xleftarrow{l} x} w_{jk}^{(l)} y_k + b_j\right) x_j^{(+i)}\right)}\right)^{-1}$$

Now note that this ratio of exponentials can be rewritten as:

$$\exp\left(\left(\sum_{y \xrightarrow{l} x} w_{kj}^{(l)} y_k + \sum_{y \xleftarrow{l} x} w_{jk}^{(l)} y_k - b_j\right)\left(x_j^{(-i)} - x_j^{(+i)}\right)\right)$$

For dimensions $j \neq i$, the difference between the two mean-field vectors will be small, so if each of the terms $w_{kj}^{(l)} y_k$, $w_{jk}^{(l)} y_k$, and b_j are on average close to zero, the above expression will be dominated by the value at $j = i$. In fact, we don't need to *assume* that $w_{jk}^{(l)} y_k$ is on average close to zero, but we can exploit a **gauge symmetry** – adding a constant value to every component of $w_{jk}^{(l)}$ does not affect the model, because it adds the same amount of energy to every situation. This is because the number of active units in each pixie is fixed, and so the number of active connections between linked pixies is also fixed. This means we can add a constant to $w_{jk}^{(l)}$, so that the average energy associated with each link is zero. We can similarly add a constant to b_j so that the average bias energy of each pixie is zero. This means that, when using an appropriate gauge, we can approximate the above ratio of exponentials as:

$$\exp\left(b_j - \sum_{y \xrightarrow{l} x} w_{ki}^{(l)} y_k - \sum_{y \xleftarrow{l} x} w_{ik}^{(l)} y_k\right)$$

This yields the update rule given in (5.30):

$$q_i \approx \left(1 + \frac{D-C}{C} \frac{t^{(r)}\left(x^{(-i)}\right)}{t^{(r)}\left(x^{(+i)}\right)} \frac{Z\left(x^{(+i)}\right)}{Z\left(x^{(-i)}\right)} \exp\left(b_i - \sum_{y \xrightarrow{l} x} w_{ki}^{(l)} y_k - \sum_{y \xleftarrow{l} x} w_{ik}^{(l)} y_k\right)\right)^{-1}$$

Chapter 6

Experiments

In this chapter, I bring together the work of the previous chapters, and I show how the framework can be used in practice. In §6.1, I first describe how I trained a model, using the neural network architecture presented in Chapter 5. Then in §6.2, I present results on three tasks, illustrating the usefulness of the model for distinguishing similarity from relatedness (see §6.2.1), for representing occasion meanings of verbs (see §6.2.2), and for composing relative clauses (see §6.2.3). These experiments apply the ideas presented in Chapter 4. The source code for my experiments is available online.[1]

6.1 Training

In §6.1.1, I begin by describing the dataset I used, and in §6.1.2, I describe how I trained a model on this dataset. Because training a model based on a random initialisation leads to long training times, I present a simple method for parameter initialisation in §6.1.3, which adapts an existing method for producing sparse count vectors.

6.1.1 Training Data

WikiWoods[2] is an annotated corpus providing DMRS graphs for 55m sentences of English (900m tokens). It was produced by Flickinger et al. (2010) and Solberg (2012) from the July 2008 dump of the full English Wikipedia, using the English Resource Grammar (ERG) (Flickinger, 2000, 2011) and the PET parser (Callmeier, 2001; Toutanova et al., 2005), with parse ranking trained on the manually treebanked subcorpus WeScience (Ytrestøl et al., 2009). It is updated with each release of the ERG, and I have used the version of WikiWoods based on the 1212 version of the ERG. The corpus is distributed by DELPH-IN.

[1] https://github.com/guyemerson/sem-func
[2] http://moin.delph-in.net/WikiWoods

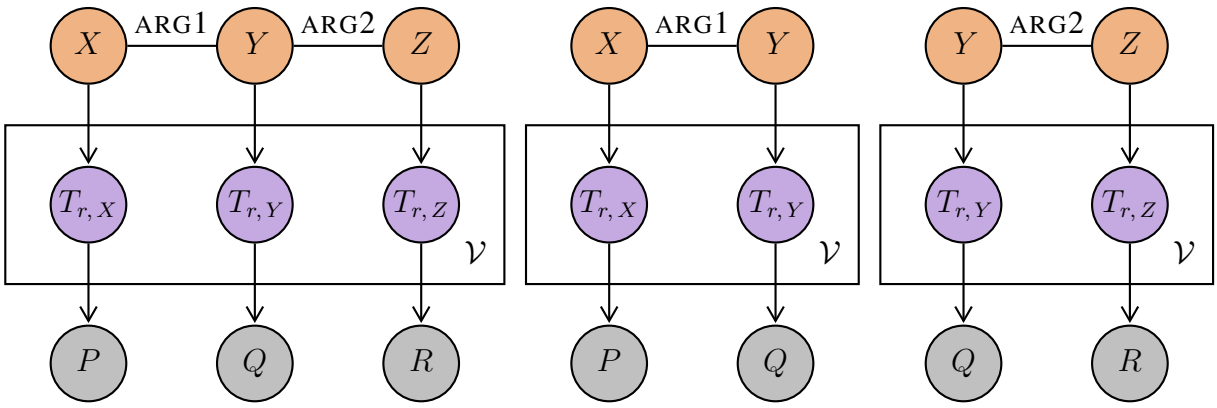


Figure 6.1: Graphical models for the three DMRS topologies extracted from WikiWoods.

DMRS topology	No. instances
Both arguments	10,091,234
ARG1 only	6,301,280
ARG2 only	14,868,213
Total	31,260,727

Table 6.1: Size of the training data, without transforming copula clauses.

To preprocess the corpus, I used the Python packages pydelphin[3] (developed by Michael Goodman), and pydmrs[4] (Copestake et al., 2016). For simplicity, I restricted attention to **subject-verb-object (SVO)** triples, although I should stress that this is not an inherent limitation of the framework, which could be applied to arbitrary graphs. The term "SVO" is a slight abuse of terminology, as DMRS graphs are semantic, not syntactic – but it is nonetheless a concise and informative term.

In the ERG, there are two kinds of predicates: **surface** predicates[5] (which correspond to words), and **abstract** predicates[6] (which correspond to grammatical constructions). In this work, I restricted attention to surface predicates, and I ignored properties such as number on nouns and tense on verbs. Surface predicates have a three part structure, consisting of a **lemma**, a **part of speech**, and a **sense**, written in that order, and separated by underscores. Note that senses are only distinguished on the basis of syntax, not semantics. For example, homonymous words like *bank* and *bass* only have a single sense in the ERG, because the homonymous senses cannot be distinguished syntactically. In contrast, *rest* has two predicates: `_rest_n_of`, a count noun taking an optional PP-complement; and `_rest_n_1`, a mass noun.

I searched for all verbal predicates in the WikiWoods treebank (identified by the part of speech `v`), excluding modal verbs such as *can* and *may* (identified by the sense `modal`), that had either an ARG1 or an ARG2, or both. For simplicity, I ignored arguments involving coor-

[3] https://github.com/delph-in/pydelphin
[4] https://github.com/delph-in/pydmrs
[5] Also called "real predicates", or "realpreds".
[6] Also called "grammar predicates", or "gpreds".

dinations (such as: *promoted events and parties*). I kept all instances whose arguments were nominal (identified by the part of speech `n`), which avoids verbs taking clausal or adjectival complements, and which also avoids pronouns and proper nouns.

The ARG1-only graphs typically correspond to intransitive verbs, but also include cases where I ignored a coordinated ARG2. The ARG2-only graphs include passives – and in particular, bare passive adjuncts (such as: *a related function*, *a self-titled album*), which are relatively common. They also includes cases where I ignored a coordinated ARG1.

As a result of this process, all data is of the form (*verb-predicate*, ARG1-*noun-predicate*, ARG2-*noun-predicate*), where one but not both of the arguments may be missing. Using predicates rather than surface forms (for example, `_write_v_to` instead of *write*, *writes*, *writing*, *written*, *wrote*) makes sense model-theoretically, and also reduces data sparsity.

However, the ERG does not automatically convert out-of-vocabulary items from their surface form to lemmatised predicates, because this cannot always be done deterministically – for example, given a past tense verb ending in *-tted*, does the stem end with *-t*, *-tt*, or *-tte*? All three such spellings exist in English (*chat*, *boycott*, *piroutte*), and determining the stem cannot be done without a lexicon. For out-of-vocabulary items, the ERG simply records the surface form. To find lemmas for these items, I applied WordNet's morphological processor Morphy (Fellbaum, 1998), as available in NLTK (Bird et al., 2009). Finally, I filtered out triples including rare predicates, so that every predicate appears at least five times in the dataset.

The number of instances of each DMRS topology is given in Table 6.1. In total, the dataset contains 72m tokens, with 88,526 distinct predicates. Graphical models for each topology are shown in Fig. 6.1.

Over 7% of the SVO triples involve the **copula** (the verb *be*). While the English copula is also used as an auxiliary verb (*it is raining*) and with adjectives (*the sky is bright*), these uses are not represented as a DMRS node. However, when the copula is used to link two noun phrases, the ERG represents it with the predicate `_be_v_id`, with the two noun phrases as its ARG1 and ARG2.

In the general case, it could be argued that the copula links distinct referents (*every tree was once a seed*), and it is certainly the case that they may differ in features such person, number, and gender. However, in encyclopaedic text, the copula is often used to equate its two arguments. In these cases, we could consider them as having the same referent, and hence there should only be one individual in the model structure. To do this, we can transform DMRS graphs involving `_be_v_id`, so that we have a single pixie node, but two observed predicates, as shown in Fig. 6.2. I applied this transformation, discarding instances where the two arguments were the same – these make sense given the full context (*a tourism region is a geographical region that has been designated...*), but are useless without it (*a region is a region*). As before, I filtered the dataset so that every predicate occurs at least 5 times. A summary of the transformed dataset is given in Table 6.2. In total, it contains 69m tokens, with 87,862 distinct predicates.

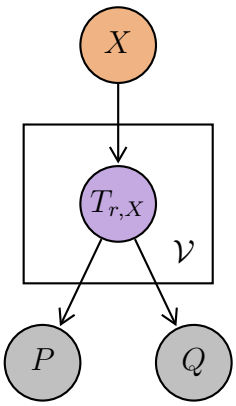


Figure 6.2: Graphical model for copula clauses, after transformation. The situation comprises a single pixie X. Two predicates P and Q are generated for this pixie, with independent and identical distributions, based on the same set of truth values.

DMRS topology	No. instances
Both arguments	8,231,139
ARG1 only	6,079,769
ARG2 only	14,557,212
Copula	1,692,945
Total	30,561,065

Table 6.2: Size of the training data, after transforming copula clauses.

For the graphical model in Fig. 6.2, the gradients for the model parameters can still be calculated as explained in §5.2. The only difference is that we have observed two samples from the distribution over predicates given a latent pixie. This means we have just two changes. Firstly, we have to sum over the two observed predicates in the last two lines of (5.15), and therefore also in (5.18) and (5.19). Secondly, when calculating the expectation $\mathbb{E}_{s|g}$ over situations given an observed DMRS graph, we simply multiply the contributions $t^{(r)}(x)/Z(x)$ from each predicate – for Metropolis-Hastings, these contributions can be seen in (5.24), and for variational inference, they can be seen in (5.29) and (5.30).

6.1.2 Training Algorithm

The model parameters can be trained using gradient descent (see §5.2). However, there are also a number of **hyperparameters** – quantities that must be set before training begins. Some hyperparameters involve decisions about the model architecture (see §5.1) – in particular, to define the semantic space, we need to set the dimensionality D, and the cardinality C.

Other hyperparameters involve decisions about the training algorithm. Since the gradient cannot be calculated exactly (see §5.2), we first have to decide how to approximate it, such as using the MCMC method presented in §5.3, or the variational inference method presetned in §5.4. This first decision can be seen as a discrete hyperparameter. With both of these algo-

rithms, we have to decide on the number of update steps. With the MCMC algorithm, we also have to decide on the number of samples.

While the gradient tells us whether to increase or decrease each parameter, it does not tell us *how much* we should change it. Once we have calculated gradients of all model parameters, we still need to choose a **step size**. We may also want to use a different step size for each parameter.

Many schemes for choosing step sizes have been proposed. I used a version of AdaGrad (Duchi et al., 2011),[7] which compares the size of the gradient with previously calculated gradients, allowing it to respond when gradients become larger or smaller. At each update step, we find the square of the gradient (componentwise), and keep a sum of the squares of previous gradients. The current gradient is compared to the square root of this sum, to determine the step size, as shown in (6.1), where θ_t is the value of parameter θ at time t, $\mathcal{L}$ is the objective function we are aiming to optimise, G_t is the sum of the square gradients, and η is a hyperparameter, called the **learning rate**.

However, if the initial weights are far from the optimum, they will need to change drastically during training, and AdaGrad may reduce the step size too much. I therefore used a modified version of AdaGrad, called RMSProp (Tieleman, 2012), with an exponential **decay** of this sum, as shown in (6.2), where $\alpha \in [0, 1]$ is another hyperparameter. The two hyperparameters η and α together determine the general step size, and how far back we look at previous gradients.

$$\theta_{t+1} = \theta_t + \frac{\eta}{\sqrt{G_t}} \frac{\partial \mathcal{L}}{\partial \theta} (\theta_t) \tag{6.1}$$

$$G_t = \alpha G_{t-1} + \left(\frac{\partial \mathcal{L}}{\partial \theta} (\theta_t) \right)^2 \tag{6.2}$$

Gradients will vary between DMRS graphs, and updating parameters based on one graph at a time will lead to high variance in the updates. On the other hand, averaging gradients across many graphs (or even the whole dataset) leads to slow training, because a large amount of computation is required to make a single update. I trained the model using **minibatches** of DMRS graphs, where the gradient is summed for a small number of graphs. The size of the minibatch is an additional hyperparameter. As each minibatch can be processed independently, minibatches can be processed in parallel, with an update made whenever a minibatch is finished.

Finally, we may have expectations about likely parameter values. For example, we might believe that very large values would indicate overfitting. We can use **regularisation** to enforce a prior over model parameters. Rather than directly maximising the log-likelihood $\log \mathbb{P}(g)$ (the log-probability of generating a DMRS graph g, for the given parameter values), we optimise an objective function $\mathcal{L}$ that includes additional terms penalising certain parameter values. I used L1 and L2 regularisation, which penalise the absolute value of a parameter, and the square value of a parameter, respectively. This is shown in (6.3), where λ_1 and λ_2 are hyperparameters.

[7] I also experimented with other update schemes, such as Adam (Kingma and Ba, 2015) and AdaDelta (Zeiler, 2012), but I did not find a noticeable difference between them.

Intuitively, these terms mean that parameters can only become large if they are particularly useful for modelling the data, which reduces overfitting. L1 regularisation encourages sparsity (setting parameters to 0), while L2 regularisation more strongly penalises large values.

$$\mathcal{L} = \log \mathbb{P}(g) - \lambda_1|\theta| - \lambda_2|\theta|^2 \tag{6.3}$$

Some of the training hyperparameters can also be set differently for different sets of parameters. For example, it may make sense to set $\eta, \alpha, \lambda_1, \lambda_2$ differently for the semantic role parameters and the semantic function parameters, since they control distinct parts of the model.

6.1.3 Parameter Initialisation

Although it is possible to initialise the model parameters randomly at the start of training, I found that this leads to a long training time, due to slow convergence. I suspect that this is because the co-occurrence of predicates is mediated via at least two latent vectors, which leads to mixing of semantic classes in each dimension, particularly in the early stages of training. Such behaviour can similarly happen with complicated topic models – for example, Ó Séaghdha (2010) observed this for their "Dual Topic" model. Carefully initialising the model parameters allows a drastic reduction in training time.

Firstly, we can note that nominal and verbal pixies naturally lie in separate subspaces – that is, their active units are almost always in different dimensions. This is because I have restricted attention to SVO triples, with a verbal predicate for Q in Fig. 6.1, and nominal predicates for P and R. We must therefore minimise the probability of generating nominal predicates for Q, and verbal predicates for P and R. As these predicates are generated from latent pixies, this forces the pixies to have different active dimensions, and hence the semantic functions to have different nonzero values. I have observed this separation of dimensions when starting from a completely random initialisation. To avoid this initial training time, I simply assigned half the dimensions to nominal predicates and half to verbal predicates.[8]

To further improve parameter initialisation, I used a simple method for producing sparse count vectors. In particular, I used a simplified version of Random Positive-only Projections, a random-indexing technique proposed by QasemiZadeh and Kallmeyer (2016). Each context is randomly assigned to a dimension. Many contexts will therefore be assigned to the same dimension. This random assignment of dimensions means that information is lost, but using a small number of dimensions (compared to the vocabulary size) allows us to construct vectors quickly and efficiently.

I defined contexts using semantic roles – each pair (*predicate*, ARG-n) defines a context, randomly assigned to a dimension. We first count how many times each dimension occurs with

[8] For a dataset with more varied DMRS topologies (for example, with verbs taking clausal complements, and nouns also taking complements), it would probably not be optimal to partition dimensions in this way. For example, a verb might be observed with both nominal and clausal complements.

each target predicate. This gives us counts n_{ri}, where r ranges over predicates, and i ranges over dimensions. Based on these counts, we can calculate frequencies, as shown in (6.4). This process can be carried out separately for nouns and verbs, to keep them in distinct subspaces.

$$f_{ri} = \frac{n_{ri}}{\sum_{r,i} n_{ri}} \qquad f_{r\cdot} = \sum_i f_{ri} \qquad f_{\cdot i} = \sum_r f_{ri} \tag{6.4}$$

We can then calculate PPMI vectors, as shown in (6.5). This compares the observed frequency f_{ri} with the expected frequency if contexts were completely random, $f_{r\cdot} f_{\cdot i}$.

$$v_i^{(r)} = \max\left\{0,\, \log \frac{f_{ri}}{f_{r\cdot} f_{\cdot i}}\right\} \tag{6.5}$$

Because this method uses PPMI, we can use the same hyperparameters discussed by Levy et al. (2015a). In particular, we can **smooth** the dimension frequencies by taking them to the power α, as shown in (6.6), and we can add a **negative offset** $\log k$ to the PMI scores, as shown in (6.7). Levy et al. find that these parameters are important, recommending $\alpha = 0.75$ and $k = 5$. However, because we are not using the vectors in the same way, we should not necessarily expect the optimal hyperparameters to be the same. Indeed, in the experiments reported in §6.2, I did not find these hyperparameters to be useful.

$$f_i \propto \left(\sum_r n_i^{(r)}\right)^{\alpha} \tag{6.6}$$

$$v_i^{(r)} = \max\left\{0,\, \log \frac{f_i^{(r)}}{f^{(r)} f_i} - \log k\right\} \tag{6.7}$$

We can also introduce a **scaling** hyperparameter – since the vectors are used as parameters in a feedforward neural net, and not to calculate cosine similarity, the magnitude of the vector matters. So, we can multiply the PMI score in (6.5) by some factor. In fact, for the tasks considered in §6.2, I found empirically that a factor close to 1 is optimal.

After initialising the parameter vectors $v_i^{(r)}$ using the above technique, we also need to initialise the bias terms $a^{(r)}$. If we keep a parameter vector fixed, varying the bias changes how likely the predicate is to be true in general, while maintaining the relative probabilities of truth for different pixies. So, we can initialise each bias so that the expected frequency of a predicate matches the observed frequency $f^{(r)}$. The frequency is given by (6.9).

$$f^{(r)} = \sum_x \mathbb{P}(x)\mathbb{P}(r|x) \tag{6.8}$$

$$= \mathbb{E}_x\left[\frac{1}{Z(x)} f^{(r)} t^{(r)}(x)\right] \tag{6.9}$$

However, the above equation cannot be calculated exactly. If we assume that the total weight of other predicates being true can be approximated as a constant value $Z_\star$, then when r is true of x, we have $Z(x) \approx Z_\star + f^{(r)}$. The hyperparameter $Z_\star$ lets us control the proportion of the vocabulary that should typically be true at the same time. We can approximate the expectation by applying $t^{(r)}$ to a mean-field vector $x^\star$, which has value $2C/D$ across all dimensions in the subspace (either the noun subspace or verb subspace, as appropriate). This approximation is shown in (6.10) and (6.11), and leads to the formula in (6.12) for initialising the biases.

$$f^{(r)} \approx \frac{f^{(r)}}{Z_\star + f^{(r)}} t^{(r)}(x^\star) \tag{6.10}$$

$$= \frac{f^{(r)}}{Z_\star + f^{(r)}} \left(1 + \exp\left(-v_i^{(r)} x_i^\star + a^{(r)}\right)\right)^{-1} \tag{6.11}$$

$$\implies \quad a^{(r)} \approx v_i^{(r)} x_i^\star + \log\left(\left(f^{(r)} + Z_\star\right)^{-1} - 1\right) \tag{6.12}$$

Once the semantic function parameters have been initialised, the semantic role parameters (in the CaRBM) can be initialised based on mean-field vectors. Each semantic function defines a mean-field vector for a single-pixie situation, as described in §5.4 and illustrated in Fig. 5.3b. For each SVO triple in the training data, we can take the mean-field vectors for the observed predicates, and for each semantic role, we can calculate the mean-field activation of each pair of dimensions of the linked pixies. This is simply the outer product of the mean-field vectors – if we have a link $x \xrightarrow{l} y$, and vectors x_i and y_j, then we have an activation $x_i y_j$ for the link.

We can average these mean-field activations across the whole training set, to get an observed average activation $m_{ij}^{(l)}$ for each semantic role l. We can then initialise the parameters $w_{ij}^{(l)}$ using the PPMI of these activations, as shown in (6.13).[9] It makes sense that $w_{ij}^{(l)}$ increases as the logarithm of $m_{ij}^{(l)}$, because the probability of a unit being active increases as the exponential of $w_{ij}^{(l)}$. The PPMI compares the observed activation $m_{ij}^{(l)}$ with the expected activation for completely random vectors, which is $(C/D)^2$.

$$w_{ij}^{(l)} = \max\left\{0,\, \log\left(m_{ij}^{(l)}\right) - 2\log\left(\frac{C}{D}\right)\right\} \tag{6.13}$$

As with the PPMI vectors for semantic functions, we can use the hyperparameters discussed by Levy et al., as well as a scaling hyperparameter. In the experiments reported in §6.2, I found the negative offset and scaling hyperparameters to be useful.

Finally, we can initialise the biases b_i, so that each dimension has a expected probability of C/D of being active, before the cardinality constraint is applied. Adding a constant to all biases has no effect on the normalised probabilities, because it adds a constant energy to every

[9] As mentioned in §5.4.3, it may be better to choose a gauge so that the average energy of a link is 0. This can be achieved by replacing the 0 in (6.13) with an appropriate negative value.

situation. In physics, this is known as a **gauge symmetry**. However, a judicious choice of bias can make calculations more numerically stable, by avoiding underflow errors (when quantities are extremely close to 0).

$$b_i = -\log\left(\frac{C}{D}\right) \tag{6.14}$$

6.2 Experimental Results

Finding a good evaluation task is far from obvious. Simple similarity tasks do not require structured semantic representations like dependency graphs, while tasks like textual entailment require a level of coverage beyond the scope of this thesis. As well as evaluating on lexical similarity (see §6.2.1), I also chose to consider the SVO similarity and RELPRON datasets, described in §6.2.2 and §6.2.3, because they provide restricted tasks which allow us to explore approaches to semantic composition. A brief discussion of future work is given in §6.2.4.

I compare my model to two vector baselines. The first is a standard Skip-gram model (Mikolov et al., 2013), trained on the plaintext version of the WikiWoods corpus. The second is the same Skip-gram model, but trained on the SVO triples I used to train my model – contexts are therefore defined as occurence in the same triple. In both cases, I used the implementation in Gensim (Řehůřek and Sojka, 2010), with default hyperparameter settings.

In the experiments reported in the following sections, I initialised the model as described in §6.1.3, without further training. These results were reported in Emerson and Copestake, 2017b. The earlier results reported in Emerson and Copestake, 2016 used a random initialisation and MCMC gradient descent. The long training time meant that it was difficult to tune hyperparameters, and results were considerably worse than those reported below. In future work, I intend to perform gradient descent on the carefully initialised model. For all three tasks considered below, I used the variational inference algorithm presented in §5.4.2 to approximately calculate conditional probabilities.

6.2.1 Lexical Similarity

Lexical similarity datasets measure similarity between pairs of words, as judged by human annotators. In evaluating on these datasets, I have two aims. Firstly, I aim to show that the performance of my model is competitive with state-of-the-art vector space models. Secondly I aim to show that my model can specifically target **similarity** rather than **relatedness**. For example, the predicates for *painter* and *painting* are related (since a painter paints paintings), but they are unlikely to be true of the same individuals, and the individuals they are true of are unlikely to share features. In contrast, the predicates for *painter* and *artist* are similar, because they are likely to be true of the same individuals, and the individuals they are true of are likely to share features. Vector space models tend to conflate these two notions.

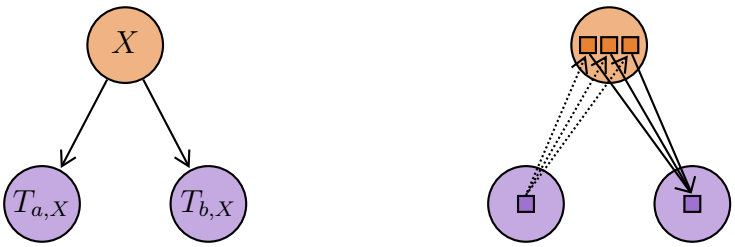


Figure 6.3: Lexical similarity as logical inference (on the left), calculated using mean-field variational inference (on the right). Repeated from Figs. 5.3a and 5.3b.

I evaluated my model on four datasets. SimLex-999 (Hill et al., 2015) and SimVerb-3500 (Gerz et al., 2016) are two datasets that aim to measure similarity, not relatedness. MEN (Bruni et al., 2014) and WordSim-353 (Finkelstein et al., 2001) primarily measure relatedness, rather than similarity. However, Agirre et al. (2009) split WordSim-353 into similarity and relatedness subsets – although the annotations were not changed, so the similarity subset is not as targeted as SimLex-999 and SimVerb-3500, where annotator instructions explicitly target similarity.

SimLex-999 contains 666 noun pairs and 222 verb pairs; I ignored the 111 adjective pairs as I did not include adjectives in my training data. SimVerb-3500 contains 3500 verb pairs, split into a development set (500 pairs) and a test set (3000 pairs). MEN contains 3000 word pairs; of these, I used the 2005 noun pairs. It also includes 29 verb pairs, but this set is too small to be useful on its own. I also ignored the 96 adjective pairs, and the 870 pairs with mixed parts of speech. Finally, WordSim-353 contains 252 pairs in the relatedness subset, and 203 pairs in the similarity subset (with some overlap between the two). Of all four datasets, SimVerb-3500 is the largest and hence most statistically significant.

For predicates which are true of similar but disjoint sets of individuals, annotations in these datasets are not completely consistent. For example, SimLex-999 gives a low score of 1.8 (out of 10) to the pair (*dog*, *cat*), but a high score of 7.8 to (*rat*, *mouse*). This kind of inconsistency means that, although these datasets can be used to get a rough idea of how a model measures similarity and relatedness, it is difficult to interpret the results in model-theoretic terms (such as overlap of extensions, or similarity of features of individuals).

To calculate a similarity score in a semantic function model, we can recast similarity as inference: we can use the conditional probability of one predicate being true, given that another predicate is true, as shown in Fig. 6.3. To make this into a symmetric score, we can multiply the conditional probabilities in both directions. As discussed in §4.5, using a conditional probability should in principle measure the overlap between predicates' extensions, rather than similarity of features – but disjointness of extensions is difficult to learn from distributional data alone. Furthermore, even with training data that would allow the model to learn predicates with disjoint extensions but similar features, the conditional probabilities might also encode the probability of making mistakes – for example, a speaker might know that rats and mice are disjoint, but still mistake one for the other.

Model	SL Noun	SL Verb	SimVerb	MEN	WS Sim	WS Rel
Skip-gram	.40	.23	.21	**.62**	**.69**	.46
SVO Skip-gram	.44	.18	.23	.60	.61	.24
Semantic Functions	**.46**	**.25**	**.26**	.52	.60	**.16**

Table 6.3: Spearman rank correlation with average annotator judgements, for SimLex-999 (SL) noun and verb subsets, SimVerb-3500, MEN, and WordSim-353 (WS) similarity and relatedness subsets. Note that we would like to have a *low* score for WS Rel, which measures relatedness, rather than similarity.

Results are shown in Table 6.3.[10] We can see that the semantic function model is competitive with Skip-gram, but has qualitatively different behaviour, as it has very low correlation for the relatedness subset of WordSim-353. Vector space models tend to have high performance on both subsets – although we can see that SVO Skip-gram has intermediate performance on WS Rel. Compared to Skip-gram, the semantic function model has lower performance on MEN and the similarity subset of WordSim-353, but these two datasets were not annotated to specifically target similarity, in the sense given above. For SimLex-999 and SimVerb-3500, which do target similarity, performance is higher than Skip-gram.

It is also interesting to note that SVO Skip-gram performs notably better than normal Skip-gram, despite being trained on a much smaller amount of data – only 72m tokens, rather than 900m. This is also despite the fact that the default hyperparameter settings have been tuned for the normal use of Skip-gram. Tuning hyperparameters for SVO Skip-gram might further increase the gap between the two.

For my model, hyperparameters for each dataset were tuned on the remaining datasets, except for SimVerb-3500, which has its own development set. Five different random seeds were used (for the random indexing), and the results averaged for each hyperparameter setting. I found that the optimal settings for nouns and verbs differ considerably. For example, if the settings for SimVerb-3500 and the verb subset of SimLex-999 are chosen based on the performance on the remaining datasets (which are all noun-based), performance is considerably worse, and below that of Skip-gram.

For the smoothing and negative offset hyperparameters discussed by Levy et al. (2015a), the best settings differ from those suggested in that paper. In particular, I found that, unlike for normal word vectors, it was unhelpful to use a negative offset for PPMI scores. The negative offset k encourages sparsity, by setting values to zero if the PMI is only slightly positive, so that only the strongest features remain. This loses information, but also removes noise in the data. However, when using vectors as parameters for a semantic function, sparsity is already enforced in the pixie vectors. This means that small parameters have little effect on the mean-field vectors, which effectively provides a different way to remove noise in the data. In

[10] Performance of Skip-gram on SimLex-999 is higher than reported by Hill et al. (2015). Despite correspondence with the authors, it is not clear why their figures are so low.

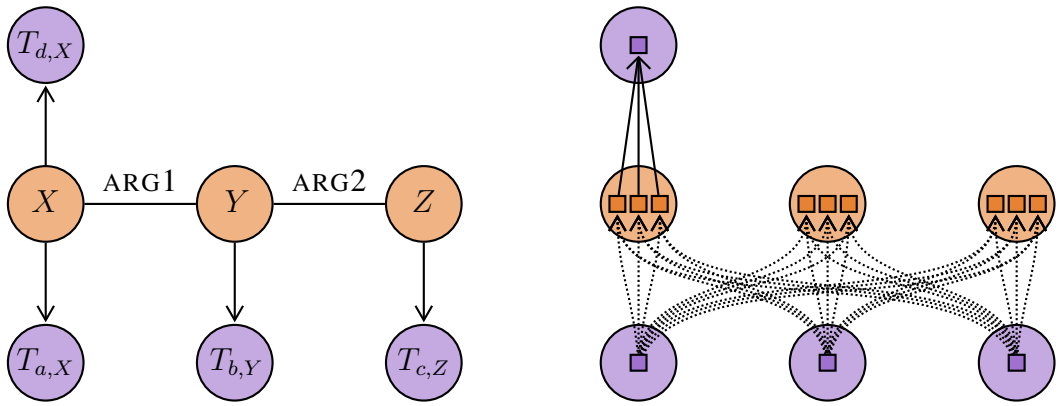


Figure 6.4: Contextual logical inference (on the left), calculated using mean-field variational inference (on the right). Repeated from Figs. 5.3c and 5.3d. After jointly calculating mean-field vectors for all pixie nodes, we apply a semantic function to one mean-field vector. For the RELPRON dataset (see §6.2.3), the function is applied to either X (as shown) or Z, depending on whether the property has a subject or object relative clause. For the GS2011 dataset (see §6.2.2), the function is applied to Y.

this setting, it appears that the loss of information caused by the negative offset is harmful for performance. For the smoothing hyperparameter α, results were less conclusive, but I did not observe the increased performance reported by Levy et al. for vector space models. In the absence of evidence suggesting that we should choose $\alpha < 1$, it seems reasonable to set $\alpha = 1$, as this removes the need to tune this hyperparameter.

6.2.2 Similarity in Context

Grefenstette and Sadrzadeh (2011) produced a dataset of pairs of SVO triples, where only the verb varies in the pair. Each pair was annotated for similarity. For example, annotators had to judge the similarity of the triples (*table*, *show*, *result*) and (*table*, *express*, *result*). In line with lexical similarity datasets, a system can be evaluated using the Spearman rank correlation between the system's scores and the average annotations.

For each triple, I calculated the mean-field vector for the verb, conditioned on all three predicates. I then calculated the probability that the other verb's predicate is true of this mean-field vector, as shown in Fig. 6.4 (except that we are interested in Y rather than X). To get a symmetric score, I multiplied the probabilities in both directions. For the vector space models, I simply summed the vectors for the three words, and then calculated the cosine similarity of the two sums of vectors.

Semantic function hyperparameters were tuned based on average performance across the lexical similarity datasets (see §6.2.1), and semantic role hyperparameters were tuned based on the RELPRON development set (see §6.2.3).

Results are given in Table 6.4. The performance of my model (.25) matches the best model Grefenstette and Sadrzadeh consider. I also include results for an ensemble, which combines

Model	GS2011
Skip-gram, Addition	.12
SVO Skip-gram, Addition	.30
Semantic Functions	.25
SVO Skip-gram and Sem-Func Ensemble	**.32**

Table 6.4: Spearman rank correlation with average annotator judgements, on the GS2011 dataset for similarity in context.

Model	Dev	Test
Skip-gram, Addition	.50	.47
Semantic Functions	.20	.16
Skip-gram and Sem-Func Ensemble	**.53**	**.49**

Table 6.5: Mean average precision on the RELPRON development and test sets. The Skip-gram model was trained on a larger training set (a more recent version of Wikipedia), to allow a direct comparison with Rimell et al.'s results.

the scores produced by my model and by the SVO Skip-gram model. The performance of this ensemble (.32) matches the improved model of Grefenstette et al. (2013), despite using less training data. Furthermore, the fact that the ensemble outperforms both the semantic function model and the vector space model shows that the two models have learnt different kinds of information. If they made the same kinds of mistakes, combining the models would not give an improvement. This improvement is also not due to the combined model having a larger capacity – increasing the dimensionality of the individual models did not give this improvement.

6.2.3 Composition of Relative Clauses

Rimell et al. (2016) produced the RELPRON dataset, which aims to evaluate how well a model can perform semantic composition – in particular, composition of relative clauses. It consists of a set of **terms**, each paired with up to ten **properties**. Each property is a short phrase, consisting of a hyperonym of the term, modified by a relative clause with a transitive verb. For example, a *telescope* is a *device that astronomers use*, and a *saw* is a *device that cuts wood*. The task is to identify the properties which apply to each term, viewed as a retrieval task: given a single term, and the full set of properties, the aim is to rank the properties, with the correct properties at the top of the list. There are 65 terms and 518 properties in the development set, and 73 terms and 569 properties in the test set. Unlike the other datasets considered in this chapter, the human ceiling on this dataset is near 100%, far higher than state-of-the-art performance with vector space models.

Every property follows one of only two syntactic patterns – a noun modified by either a subject relative clause (*that cuts wood*) or an object relative clause (*that astronomers use*). This dataset therefore lets us focus on evaluating semantics, rather than parsing. A model

that uses relatedness can perform fairly well on this dataset – for example, *astronomer* can predict *telescope*, without knowing what relation there is between them. However, the dataset also includes lexical **confounders** – for example, a *document that has a balance* is a financial *account*, not the quality of *balance* (not falling over). The lexical overlap means that a vector addition model is easily fooled by such confounders, and indeed the best three models that Rimell et al. tested all ranked this confounding property at the top, when retrieving properties for the term *balance*.

We can represent each property as a situation of three pixies, as shown in Fig. 6.4. Although the properties are syntactically noun phrases, we have the same set of semantic roles as in a transitive clause. For each property, I calculated the contextual mean-field vectors, conditioned on all three predicates. To find the probability that the term's predicate is true, we can apply the term's semantic function to the head noun's mean-field vector. The difference between subject and object relative clauses is captured by whether this vector corresponds to the ARG1 pixie X, or the ARG2 pixie Z.

For the vector space models, I took a weighted sum of the vectors for the three words in the property, and calculated cosine similarity with the term. Since we may not want to weight each word equally, there are two additional hyperparameters (three minus one, because the vector magnitude does not matter for cosine similarity). These hyperparameters were tuned on the development set. For my model, I first tuned the semantic function hyperparameters based on average performance across the lexical similarity datasets, and then tuned the remaining hyperparameters on the development set.

Results are given in Table 6.5. My model performs worse than vector addition, perhaps as expected, since it does not capture relatedness, (as explained in §6.2.1), but many properties can be predicted based on relatedness. As in §6.2.2, I also give results for an ensemble combining my model with the vector space model, re-tuning hyperparameters for the component models. There is an additional hyperparameter controlling the importance of each of the two models. The ensemble performs better than either model alone – just as argued in §6.2.2, this shows that my model has learnt different information from the vector space model.

In particular, we can inspect the tuned weights for the vector space model's weighted sum, as I tuned these separately when using the vector space model on its own and as part of the ensemble. When part of the ensemble, it has a much lower weight for the head noun, which shows that the semantic function model has effectively taken over responsibility for deciding if the head noun is a hyperonym of the term, while the vector space model can better detect relatedness between the other noun and the term.

Finally, the ensemble also improves performance on the lexical confounders, of which there are 27 in the test set. The vector space model places 17 of them in the top rank, and all of them in the top 4 ranks. The ensemble model, however, succeeds in moving 9 confounders out of the top 10 ranks. There is clearly further progress to be made, since two thirds of the confounds

are still in the top 10 ranks. However, to my knowledge, this is the first system that manages to improve both overall performance as well as performance on the confounders.

6.2.4 Future Work

In future work, I plan to use the datasets produced by Herbelot and Vecchi (2016) and Herbelot (2013), where pairs of "concepts" (such as *tricycle*) and "features" (such as *is small*) have been annotated with suitable quantifiers (out of these options: *all*, *most*, *some*, *few*, *no*). This would allow an experimental evaluation of the approach to quantification presented in §4.5 and further developed in Chapter 7. One challenge posed by these datasets is the syntactic variation in the features, such as *has 3 wheels* and *lives on coasts*. These datasets can therefore be seen as a further stepping stone between the datasets considered above and general textual entailment, since they are more varied than the above datasets, but more targeted than textual entailment datasets.

Chapter 7

Quantifiers and First-Order Logic

Model theory requires quantifiers to give truth values to propositions (see §1.3), but they resist integration into vector space approaches, as discussed in §2.3.2. In this chapter, I give an account of how quantifiers can be interpreted in my framework. I first provide a background on generalised quantifiers in §7.1, and then explain in §7.2 how quantifier scope is underspecified in (D)MRS. These two sections clarify the classical (non-probabilistic) view which I aim to emulate. The main contribution of this chapter is in §7.3, where I propose a probabilistic version of quantification. In §7.4, I discuss how a probabilistic approach provides a better account of vague quantifiers, which are challenging for classical model theory. In §7.5, I explain how to combine precise quantifiers with vague predicates, and in §7.6, I relate this chapter to the simpler account given in §4.5. Although this chapter is relatively programmatic, and although I do not have any experimental results to report, this work constitutes significant progress towards using distributional semantics in an expressive logic like first-order predicate calculus.

7.1 Generalised Quantifiers

Partee (2012) recounts how quantifiers have played an important role in the development of model-theoretic semantics, seeing a major breakthrough with Montague (1973)'s work, and culminating in the theory of **generalised quantifiers** (Barwise and Cooper, 1981; Van Benthem, 1984), which I will summarise in this section.

We can calculate the truth of a proposition using a **scope tree**, which is a convenient way to represent a logical proposition, as illustrated in Fig. 7.2. The basic idea is to calculate the truth value for the whole proposition by working bottom-up through the tree. The leaves of the tree are expressions including variables. They can be assigned truth values, if each variable is fixed as a specific individual in the model structure. To assign a truth value to the whole proposition, we work up through the tree, quantifying the variables one at at time. Once we reach the root, all variables have been quantified, and we are left with a truth value.

$$\forall x\ \textit{picture}(x) \rightarrow \exists z \exists y\ \textit{tell}(y) \wedge \textit{story}(z) \wedge \text{ARG1}(y, x) \wedge \text{ARG2}(y, z)$$

Figure 7.1: A first-order logical proposition, repeated from Fig. 1.2a.

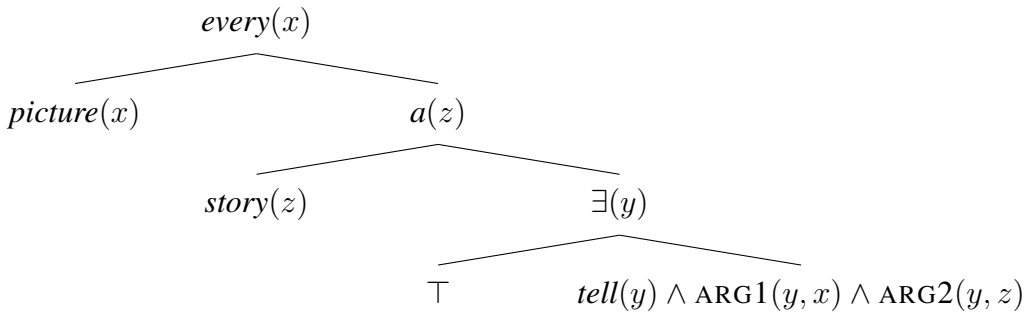


Figure 7.2: A scope tree, equivalent to Fig. 7.1 above. Each non-terminal node is a quantifier, with its bound variable in brackets. Its left child is its restriction, and its right child its body.

Each **quantifier** is a non-terminal node in the tree, with two children – its **restriction** (on the left) and its **body** (on the right). It also quantifies exactly one variable, called its **bound variable**. Each node in the tree also has a certain number of **free variables**. For each leaf, its free variables are exactly the variables appearing in the logical expression. For each quantifier, its free variables are the union of the free variables of its restriction and body, minus its own bound variable. For a well-formed scope tree, the root of the tree has no free variables. Each node in the tree defines a truth value, given a fixed value for each of its free variables.

To define the truth value for a quantifier node, we look at its restriction and body. Given fixed values for the quantifier's free variables, the restriction and body only depend on the quantifier's bound variable. This means we can work out whether the restriction and body are true, for different values of the bound variable. The restriction and body therefore each define a set of individuals in the model structure – the individuals for which the restriction is true, and the individuals for which the body is true. I will write these as $\mathcal{R}(v)$ and $\mathcal{B}(v)$, respectively, where v denotes the fixed values for all free variables.

Generalised quantifier theory says that to know whether a quantified proposition is true, we only need to know two quantities: the cardinality of the restriction set $|\mathcal{R}(v)|$, and the cardinality of the intersection of the restriction and body sets $|\mathcal{R}(v) \cap \mathcal{B}(v)|$. Natural language quantifiers can all be expressed in this way, with examples given in Table 7.1.

Quantifier	Condition
some	$\lvert\mathcal{R}(v) \cap \mathcal{B}(v)\rvert > 0$
every	$\lvert\mathcal{R}(v) \cap \mathcal{B}(v)\rvert = \lvert\mathcal{R}(v)\rvert$
no	$\lvert\mathcal{R}(v) \cap \mathcal{B}(v)\rvert = 0$
most	$\lvert\mathcal{R}(v) \cap \mathcal{B}(v)\rvert > \frac{1}{2}\lvert\mathcal{R}(v)\rvert$

Table 7.1: Classical truth conditions for precise quantifiers.

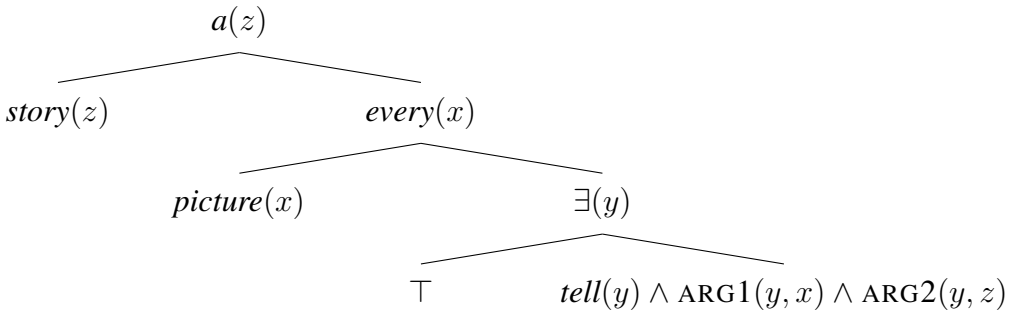


Figure 7.3: An alternative scope tree to Fig. 7.2, for the sentence *every picture tells a story*.

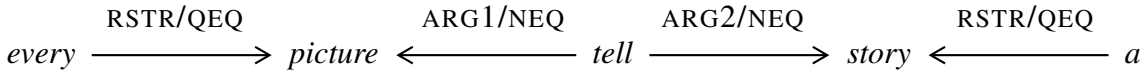


Figure 7.4: DMRS graph, underspecifying between Figs. 7.2 and 7.3. Repeated from Fig. 1.2c.

7.2 Quantifier Scope in Minimal Recursion Semantics

So far in this thesis, I have not used scope trees, and they may seem quite different from DMRS graphs. However, as mentioned in §1.3.3, (D)MRS underspecifies scope (Copestake et al., 2005). For example, while Fig. 7.2 shows the more likely reading of *every picture tells a story*, it also has another reading, that there is one single story which all pictures tell, as shown in Fig. 7.3. The preference for one scope tree or another depends on the particular words in the sentence, as well as the surrounding context.

The graph in Fig. 7.4 is underspecified between these two scopes, simply giving constraints on a possible scope tree. It specifies that *picture* is in the restriction of *every* and that *story* is in the restriction of *a*. In this case, *picture* and *story* are the immediate children of the respective quantifiers, but in the general case, there could be nodes in between the two – a RSTR/QEQ link specifies **dominance** in the scope tree, but not **immediate dominance**.[1]

The ERG does not introduce quantifiers for event variables, as shown in Fig. 7.4. I assume that event variables are existentially quantified with no constraints on the restriction, and these quantifiers scope low (immediately above the event variable), as shown in Figs. 7.2 and 7.3.

Efficiently determining the set of scope trees that correspond to a given (D)MRS is an interesting computational challenge when there is a large number of quantifiers (for discussion, see: Koller and Thater, 2005, 2006), but the details of such an algorithm are not important here. The aim of this section was to explain how scope trees relate to the DMRS graphs used so far in this thesis. In the rest of this chapter, I will work exclusively with scope trees.

[1] Technically, a QEQ constraint means that *only quantifiers* can intervene. However, for (D)MRS representations derived from a grammar compliant with the composition algebra, this detail is unnecessary (Copestake et al., 2005, footnote 11). In any case, I only discuss quantifiers here, and not other sources of scope, such as modals (*it might rain*), scopal adverbs (*it probably rained*), and attitude report verbs (*the dog knew it rained*).

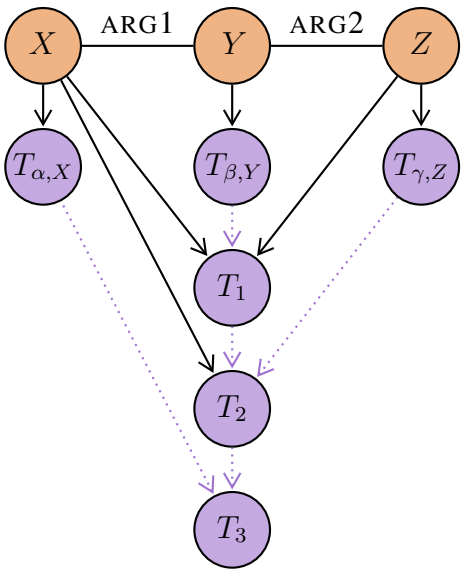


Figure 7.5: A probabilistic scope tree. T_1, T_2, T_3 correspond to non-terminal nodes in Fig. 7.2, going up through the tree. Solid lines indicate conditional dependence (on the *value* of the parent node), and dotted lines indicate the scope true (using the *distribution* of the parent node). One random variable is marginalised out at a time, until T_3 is no longer dependent on any variables. For the scope tree in Fig. 7.3, T_2 would be conditionally dependent on Z, not X.

7.3 Probabilistic Scope Trees

With a probabilistic model structure, we can adapt the definition of generalised quantifiers, by replacing sets of individuals with distributions over pixies. The basic process is the same as in the classical case – we have a scope tree, and we work bottom up through the tree. However, we will replace binary truth values with probabilities of truth. This extends the the account of quantifiers given in §4.5, which only considered a single quantifier and a single variable. We will now consider multiple quantifiers and multiple variables, thereby moving from syllogistic logic to first-order predicate logic. However, the core insight from §4.5 still holds – quantification corresponds to a conditional probability, where a random variable has been marginalised out. This correspondence follows from the more basic correspondence between sizes of sets and probabilities of sets, which was exploited in the proof of equivalence given in §4.5.1.

We begin with a probabilistic model structure, in the form of a probabilistic graphical model, as introduced in §3.5. The pixie nodes together define a distribution over situations. We also have a scope tree, which represents a first-order proposition. In the classical case, given a set of situations, a scope tree can be assigned a truth value. In the probabilistic case, given a distribution over situations, a scope tree can be assigned a probability of truth. Note that this probability is not a function of one situation, but rather of the whole distribution.

For each predicate in the scope tree, the graphical model includes one pixie node and one truth value node (this is from the one-to-one correspondence between predicates and variables in neo-Davidsonian semantics). We now introduce an additional binary-valued random variable

for each quantifier node in the scope tree, as illustrated in Fig. 7.5. Each of these random variables is conditionally dependent on all of the free variables for its corresponding scope tree node. The distributions for these random variables will be defined bottom-up through the tree.

For each leaf, the conditional distribution of the random variable is determined by the semantic function(s).[2] For each non-terminal, we have to define a distribution to mimic the classical case. For precise quantifiers, these distributions are degenerate – they are either true with probability 1, or false with probability 1. This matches the interpretations of the universal and existential quantifiers in §4.5, where we had precise constraints on conditional probabilities. However, when we come to vague quantifiers in §7.4, we will need intermediate probabilities.

For a classical scope tree, the truth of a quantifier node depends on its free variables, and is defined in terms of the extensions of its restriction and body, in a way that removes the bound variable. For a probabilistic scope tree, the distribution for a quantifier node will be conditionally dependent on its free variables, and will be defined in terms of the distributions for its restriction and body, marginalising out the bound variable.

Let the pixie node for the bound variable be X, let the set of pixie nodes for the free variables be V, and let the probabilistic truth values for the restriction and body be R and B, respectively. In the classical case, given fixed values for all free variables, the restriction and body each define a set of individuals, which we can write as $\mathcal{R}(v)$ and $\mathcal{B}(v)$. In the probabilistic case, given fixed values for all free variables, conditioning on r or b defines distributions for X. These conditional distributions directly correspond to the classical sets, as shown in (7.1) and (7.2).[3] As explained in §4.5, I write $\mathbb{P}(r)$, $\mathbb{P}(b)$, $\mathbb{P}(v)$ for $\mathbb{P}(R{=}\top)$, $\mathbb{P}(B{=}\top)$, $\mathbb{P}(V{=}v)$, respectively.

$$\mathbb{P}(r \mid v) = \mathbb{P}(X \in \mathcal{R}(v) \mid v) \tag{7.1}$$

$$\mathbb{P}(b \mid v) = \mathbb{P}(X \in \mathcal{B}(v) \mid v) \tag{7.2}$$

For classical generalised quantifiers, we only need to consider the cardinalities $|\mathcal{R}(v)|$ and $|\mathcal{R}(v) \cap \mathcal{B}(v)|$. In a probabilistic model structure, these correspond to the probabilities $\mathbb{P}(r \mid v)$ and $\mathbb{P}(r, b \mid v)$. It therefore makes sense to consider the conditional probability $\mathbb{P}(b \mid r, v)$, because this uses both of the classical sets, as shown in (7.3). Intuitively, this makes sense – the truth of a quantified expression depends on how likely B is to be true, given that R is true.

$$\mathbb{P}(b \mid r, v) = \frac{\mathbb{P}(r, b \mid v)}{\mathbb{P}(r \mid v)} = \frac{\mathbb{P}(X \in \mathcal{R}(v) \cap \mathcal{B}(v) \mid v)}{\mathbb{P}(X \in \mathcal{R}(v) \mid v)} \tag{7.3}$$

[2] The conditional dependence of the leaf node including the verb highlights the assumption that DMRS topology is isomorphic to situation structure (see §3.1 and §3.6). The leaf has three free variables, so its truth value should depend on all three pixie nodes. However, the verb's semantic function depends only on the event. This mismatch is because the ARG1 and ARG2 roles are trivially "true", having been built into the situation structure. Removing the isomorphism would mean that this leaf node is conditonally dependent on all three pixie nodes.

[3] Given a classical model structure, we can construct a probabilistic model structure where this holds, as done in §4.5.1 for the special case of situations containing one individual. If there are symmetries in the situation structure (permuting individuals without affecting semantic role labels), each permutation needs nonzero probability.

Quantifier	Condition
some	$\mathbb{P}(b \mid r, v) > 0$
every	$\mathbb{P}(b \mid r, v) = 1$
no	$\mathbb{P}(b \mid r, v) = 0$
most	$\mathbb{P}(b \mid r, v) > \frac{1}{2}$

Table 7.2: Truth conditions for precise quantifiers, in terms of the conditional probability of the body given the restriction (and given all free variables). These conditions mirror Table 7.1.

Truth conditions for quantifiers can be defined in terms of $\mathbb{P}(b \mid r, v)$, as shown in Table 7.2. In the special case where there are no free variables, and the body and restriction are each a single predicate, the conditions for *every* and *some* reduce to the conditions given in §4.5 for the universal and existential quantifiers. This account does not cover so-called **cardinal quantifiers** like *one* and *two*. However, the ERG represents numbers not as quantifiers, but as additional predicates (like adjectives). This is compatible with Link (1983)'s lattice-theoretic approach, which allows reference to plural individuals without quantification.

To work through a specific example, we can consider the scope tree in Fig. 7.2, and the corresponding graphical model in Fig. 7.5 (if we set α, β, γ to be *picture*, *tell*, *story*). The distributions for the nodes $T_{\alpha,X}$, $T_{\beta,Y}$, $T_{\gamma,Z}$ are determined by semantic functions. We have three quantifier nodes in the scope tree, and hence three additional truth value nodes in the graphical model. We first define a distribution for T_1, which represents the $\exists(y)$ quantifier, and which depends on the free variables x and z. It is true if, out of situations involving the fixed pixies x and z, there is nonzero probability that they are the ARG1 and ARG2 of a telling-event pixie.[4] Next, we define a distribution for T_2, which represents the $a(z)$ quantifier, and depends on the free variable x. It is true if, out of situations involving the fixed pixie x and a story pixie z, there is nonzero probability that they are the ARG1 and ARG2 of a telling-event pixie. Finally, we define a distribution for T_3, which represents the *every*(x) quantifier, and has no free variables. It is true if, for situations involving a picture pixie x, we are certain to have nonzero probability that x is the ARG1 of a telling-event pixie, which has a story pixie ARG2.

To sum up, this approach allows us to directly interpret quantifiers in the distributional model, unlike the previous work discussed in §2.3.2, which either built a hybrid system (for example: Lewis and Steedman, 2013; Beltagy et al., 2016), or was not as general as first-order logic (for example: Grefenstette, 2013; Herbelot and Vecchi, 2015). The above approach also goes further than Cooper et al. (2015), who discuss quantifiers in probabilistic TTR, but only non-vague quantifiers. By expressing generalised quantifiers in terms of conditional probabilities, it is possible to capture vague quantifiers as well, as discussed in the following section.

[4] This shows this approach does not rely on neo-Davidsonian event semantics. After quantifying out an event variable with n semantic roles, we get a function from n pixies to a probability of truth. We can see this as an n-ary semantic function, and we could choose to directly represent n-ary predicates in this way. However, this requires accounting for the number of arguments – for example, representing both transitive and intransitive *eat*.

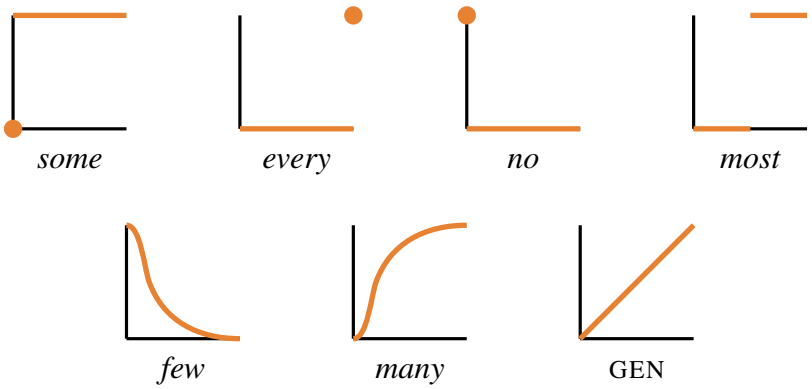


Figure 7.6: Probabilities of truth for various quantifiers. In each graph, the x-axis is the value of $\mathbb{P}(b \mid r, v)$, and the y-axis is the probability of the quantifier node being true. All axes range from 0 to 1. Quantifiers in the bottom row are vague, and GEN is the generic quantifier.

7.4 Vague Quantifiers

While *some*, *every*, *no*, and *most* can be given precise truth conditions, other natural language quantifiers are vague. In particular, we can consider the terms *few* and *many*, which don't have precise truth conditions. Partee (1988) surveys work suggesting that *few* and *many* are ambiguous between a vague cardinal reading and a vague proportional reading. As mentioned in the previous section, we can treat cardinals like adjectives, so I will only discuss the proportional readings here.

Under a classical account, *many* means that $\mathcal{R}(v) \cap \mathcal{B}(v)$ is large compared to $\mathcal{R}(v)$, but how large is underspecified. Similarly, *few* means that $\mathcal{R}(v) \cap \mathcal{B}(v)$ is small compared to $\mathcal{R}(v)$, but how small is underspecified. Note that, for the proportional reading of *few*, it is also true when the proportion is zero. For example, if someone said, *few people would argue that the most nutritious food is depleted uranium crumble*, they might believe it's almost certain that no one would argue such a thing, but they are leaving open the possibility that someone would. It would be absurd to argue back by saying, *you're wrong, no one would argue that!*.[5]

The underspecification of the proportion that makes *few* and *many* true can naturally be represented as a distribution over thresholds, or equivalently as a function from proportions to probabilities of truth (see §3.4.1). So, we can define the meaning of a vague generalised quantifier to be a function from $\mathbb{P}(b \mid r, v)$ to a probability of truth. This is illustrated in Fig. 7.6, where the non-vague quantifiers are shown in the first row, and the vague quantifiers are shown in the second. For the non-vague quantifiers, the value on y-axis is always 0 or 1, but for the vague quantifiers, we have intermediate values.

Finally, another challenging case of natural language quantification involves **generic** sentences, such as *dogs bark*, *ducks lay eggs*, and *mosquitoes carry malaria*. Generics are ubiquitous in natural language, but they are challenging for classical models, because the truth condi-

[5] This argument is due to Mary Karavaggelis (p.c.).

tions are hard to pin down, and seem to depend heavily on the content of the proposition, and on the context of use (for discussion, see: Carlson, 1977; Carlson and Pelletier, 1995; Leslie, 2008; Herbelot, 2010).

As we saw in Chapter 4, a probabilistic model can provide a natural account of context dependence, so it is reasonable to ask if the same technique can be applied to generic sentences. Indeed, Tessler and Goodman (2016) analyse generic sentences using a Bayesian approach to pragmatics, as formalised in the framework of Rational Speech Acts (RSA) (Frank and Goodman, 2012; Goodman and Frank, 2016). In this framework, literal truth values are separated from pragmatic meaning. Given a standing meaning of an utterance, and a prior over situations, a **literal listener** can construct an occasion meaning for that utterance (a posterior distribution over situations). A **pragmatic speaker** who directly observes a situation can then choose an utterance which is informative for a literal listener – in particular, they can choose the utterance which maximises a literal listener's posterior probability for the observed situation. In the general case, a pragmatic speaker may not deterministically generate an utterance, but instead probabilistically generate one, preferentially choosing informative utterances.

In terms of my framework, we can see a literal listener as defining distributions for truth value nodes, and a pragmatic speaker as defining distributions for predicate nodes (the third row of the graphical model in Fig. 3.10). I have not mentioned predicate nodes so far in this chapter, because I have been discussing literal meaning. However, given the literal meanings defined by the above probabilistic quantifiers, we can define a pragmatic speaker. This would give a more interesting distribution over DMRS graphs than the simple generative process suggested in §3.6, which simply involved choosing a predicate at random out of the true predicates. In the RSA framework, this simple generative process would be called a **literal speaker**.

In §4.1, I discussed Bayesian inference over a literal speaker, and showed how this leads to a natural account of context dependence. The crux of the RSA framework is to perform Bayesian inference over a *pragmatic* speaker. Given an observed utterance from a pragmatic speaker and given a prior over situations, a **pragmatic listener** can construct a pragmatic occasion meaning for the utterance (a posterior distribution over situations). Tessler and Goodman's insight is that this Bayesian inference of *pragmatic* meanings can account for the challenging behaviour of generic sentences. The literal meaning of a generic quantifier can be very simple (it is more likely to be true as the proportion increases), but the pragmatic meaning can have a rich dependence on the world knowledge encoded in the prior over situations. For example, the utterance *mosquitoes carry malaria* does not mean that all mosquitoes do (in fact, many do not) but it can inform a pragmatic listener to update their distribution over situations so that they have an increased expectation of mosquitoes carrying malaria.

In Fig. 7.6, I have followed Tessler and Goodman and represented the meaning of the generic quantifier as simply the identity function. At first sight, this may seem too simple to model generic sentences, but the complexity comes from pragmatic inference, and not literal meaning.

7.5 Quantification with Soft Constraints

In the previous sections, I proposed casting quantification in terms of conditional probabilities. For a probabilistic model structure that corresponds to a classical model structure, this probabilistic account gives the same truth values. However, for a probabilistic model structure expressing soft constraints, this account would seem to fail – for a model with soft constraints, conditional probabilities will never be exactly 0 or exactly 1 (except for contradictions and tautologies), as discussed in §4.5 and §5.1.2.

For example, suppose two people are in a room in an art gallery, and one person says, *every picture tells a story*. Whether a picture tells a story has vague truth conditions, but the listener might look around the room and agree that it's relatively clear what story each picture is telling. In other words, they might agree that the utterance is probably true, even though they're not 100% certain that each picture is telling a story. On the other hand, if one of the pictures is somewhat abstract, so the listener isn't quite sure what story it is supposed to be telling, they might be less willing to agree with the utterance.

The account of quantification given in §7.3 would seem to predict that the utterance is certainly false in both of the above cases, because the listener isn't 100% certain that each picture is telling a story. With probabilities between 0 and 1, *every* is always false, and *some* is always true, which would render the logic somewhat pointless. In this section, I explain how this approach to quantification can be adapted to deal with soft constraints, allowing precise quantifiers like *every* and *some* to have intermediate probabilities of truth.

There are two places where the model encodes soft constraints – in the distribution over situations, and in the semantic functions. Just one of these is enough to create a challenge. If the body of a quantifier depends on semantic functions with soft constraints, then $\mathbb{P}(r, b \mid v)$ will be strictly between 0 and 1. On the other hand, if the distribution over situations assigns nonzero probability to every combination of pixies, then we can always find one combination of pixies where the restriction and body are both true, and another combination where the restriction is true but the body false. Hence, $\mathbb{P}(r, b \mid v)$ will be strictly between 0 and 1. For the above account of quantification to give nontrivial results for precise quantifiers, it would seem that we need hard constraints, both on the distribution over situations and on the semantic functions.

How can we usefully define quantification in the face of soft constraints? The basic idea is to view soft constraints as distributions over hard constraints. This was already discussed in §3.4.1, where a vague semantic function can be seen as a distribution over precise regions (given a covariance function). As for the distribution over situations, we can consider a finite number of situations (as done in the example given in §3.3). We can see the collection of situations as itself being a situation – a **supersituation** consisting of multiple **subsituations**.[6] While the distribution over supersituations may encode soft constraints, the conditional distribution over

[6] Kratzer (2017) uses the terms "topic situation" and "resource situation", respectively.

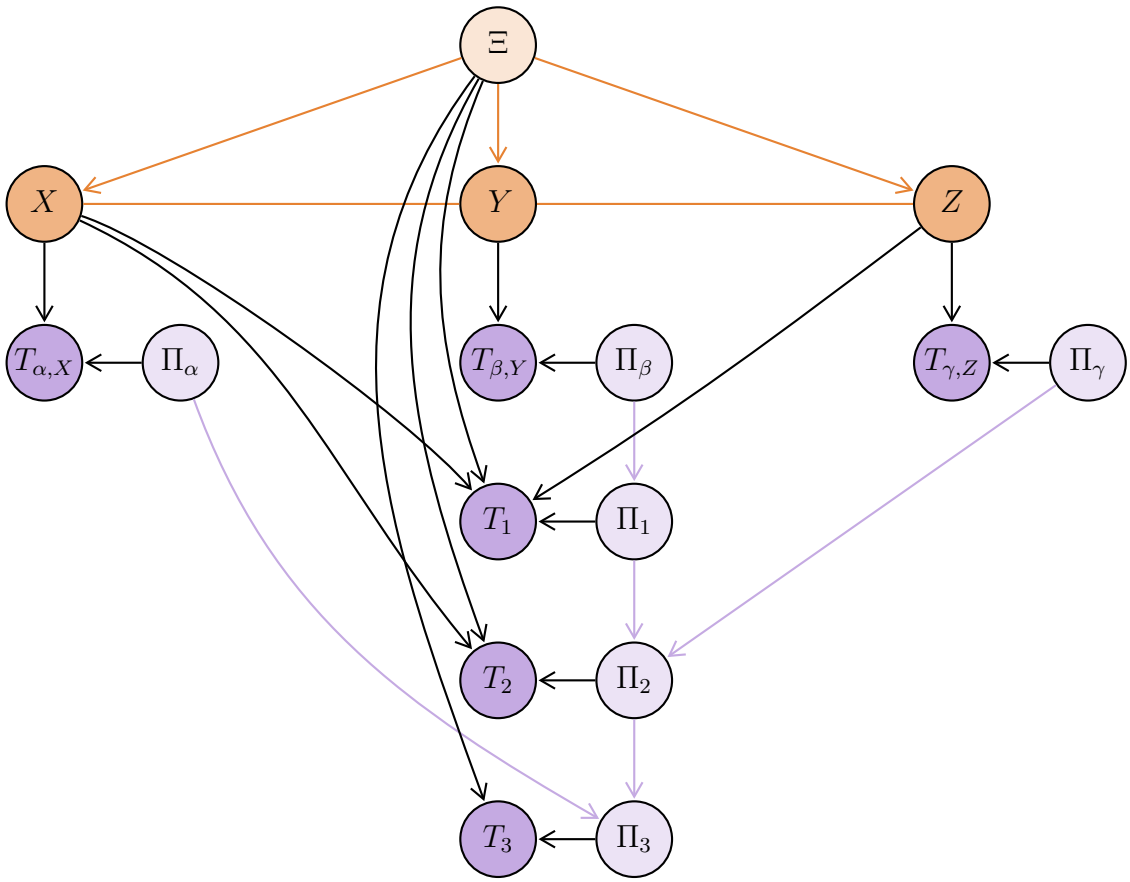


Figure 7.7: A probabilistic scope tree, accounting for soft constraints. A supersituation ξ consists of a finite number of subsituations (x, y, z). Each truth value depends on its free variables, and for non-leaf nodes, also depends on the supersituation. Given these variables, each truth value is determined by a precise function π. These functions are composed according to the scope tree. All edges indicate conditional dependence, but have been colour-coded for clarity: the orange edges denote drawing a subsituation from the supersituation; the black edges denote determining a truth value; the purple edges denote semantic composition via quantification.

subsituations (given a fixed supersituation) has nonzero probability in a finite number of cases.

This is illustrated in Fig. 7.7. At the top, we can see how the joint distribution over pixies is now dependent on a supersituation node Ξ. For any particular supersituation ξ, the joint distribution for X, Y, Z is defined by sampling a subsituation from ξ. This sampling is indicated by the orange edges. Fig. 7.5 implicitly fixed a supersituation, but it is explicit in Fig. 7.7.

One way that we might define a distribution for Ξ is to take a finite number of samples from a graphical model for situations, as shown in Fig. 7.8. Here, Ξ is simply the collection of N samples. The graphical model inside the plate encodes soft constraints for (sub)situations, but by taking a finite number of samples, we can define a distribution with hard constraints.[7]

Turning to the probabilities of truth, we can start with semantic functions (which define

[7] Technically, the conditional dependence structure inside the plate does not induce the same conditional dependence in a set of samples. For example, if we have two samples (x_1, y, z_1) and (x_2, y, z_2), there is now a conditional dependence between X and Z. So, we would either need to draw an extra undirected edge between X and Z in Fig. 7.7, or modify Fig. 7.8 so that Ξ records pairs (x, y) and (y, z), rather than triples (x, y, z).

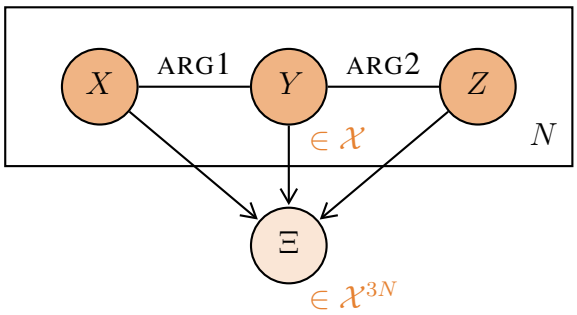


Figure 7.8: A distribution over supersituations, defined by taking N samples from an undirected graphical model for (sub)situations, so that $\Xi = \big((X_1, Y_1, Z_1), \ldots, (X_N, Y_N, Z_N)\big)$.

the leaves of the scope tree) and then consider quantification (which defines the rest of the scope tree). As previously mentioned, we can see a vague semantic function as a distribution over precise functions. For a vague function t, with a corresponding distribution over precise functions π, we have (7.4) by definition. Given a pixie and a precise function, we can find the truth value. This can be seen in the third row of Fig. 7.7.

$$t(x) = \mathbb{E}_{\pi}\left[\pi(x)\right] \tag{7.4}$$

For the quantifier nodes in the scope tree, we can view quantification as semantic composition – each scope tree node corresponds to a distribution over functions, and quantification combines the restriction and body functions into a new function. This can be seen in the purple edges in Fig. 7.7, which follow the scope tree.

Because the distribution over subsituations is determined by the supersituation, the truth of each quantifier node depends on the supersituation – as soon as we have quantified out a pixie node, the truth value node expresses information about the whole collection of sibsituations, rather than just a single subsituation. We therefore have an edge from Ξ to each quantifier truth value node. Once we have quantified out all variables, the truth value for the root node depends only on the supersituation – intuitively, a proposition quantifying over subsituations is a proposition about the supersituation. The scope tree allows us to move from semantic functions that apply to individuals to a proposition that applies to (super)situations.

The function π_Q for a quantifier determines the truth value q, given all free variables v, and the supersituation ξ, as shown in (7.5). A vague function t_Q can then be defined by marginalising out the distribution over precise functions, as shown in (7.6).

$$\mathbb{P}\left(q \,|\, v, \xi, \pi_Q\right) = \pi_Q(v, \xi) \tag{7.5}$$

$$t_Q(v, \xi) = \mathbb{P}\left(q \,|\, v, \xi\right) = \mathbb{E}_{\pi_Q}\left[\pi_Q(v, \xi)\right] \tag{7.6}$$

In §7.3, I gave an account of quantification in terms of the conditional probability $\mathbb{P}\left(b \,|\, r, v\right)$. With the supersituation now explicit, this must be amended to $\mathbb{P}\left(b \,|\, r, v, \xi\right)$. With semantic

functions now considered as distributions over precise functions, this must be further amended to $\mathbb{P}(b \mid r, v, \xi, \pi_R, \pi_B)$, where π_R and π_B are functions for the restriction and body. What remains to be shown is that this corresponds to constructing a quantifier function π_Q, from the restriction and body functions π_R and π_B.

As explained in §7.4, the probability of a quantifier being true is given by a function f_Q applied to the above conditional probability, as in (7.7). This can be rewritten as a ratio of probabilities (corresponding to the classical sets), as in (7.8). As the restriction and body depend on the bound variable u, while the quantifier does not, we need to sum over possible values, as in (7.9). This can now be factorised according to the conditional dependence structure (illustrated in Fig. 7.7), as in (7.10). Finally, we can write this expression in terms of the functions for the restriction and body, as in (7.11). Note that these functions take $u \cup v$ as an argument – by definition of a scope tree, if we add the bound variable of a quantifier to its set of free variables, we get the free variables of its restriction and body. I have written $u \cup v$ rather than $\{u\} \cup v$, to leave open the possibility that the quantifier has more than one bound variable.

$$\mathbb{P}(q \mid v, \xi, \pi_R, \pi_B) = f_Q\left(\mathbb{P}(b \mid r, v, \xi, \pi_R, \pi_B)\right) \tag{7.7}$$

$$= f_Q\left(\frac{\mathbb{P}(b, r \mid v, \xi, \pi_R, \pi_B)}{\mathbb{P}(r \mid v, \xi, \pi_R, \pi_B)}\right) \tag{7.8}$$

$$= f_Q\left(\frac{\sum_u \mathbb{P}(b, r, u \mid v, \xi, \pi_R, \pi_B)}{\sum_u \mathbb{P}(r, u \mid v, \xi, \pi_R, \pi_B)}\right) \tag{7.9}$$

$$= f_Q\left(\frac{\sum_u \mathbb{P}(u \mid v, \xi)\, \mathbb{P}(r \mid u, v, \xi, \pi_R)\, \mathbb{P}(b \mid u, v, \xi, \pi_B)}{\sum_u \mathbb{P}(u \mid v, \xi)\, \mathbb{P}(r \mid u, v, \xi, \pi_R)}\right) \tag{7.10}$$

$$= f_Q\left(\frac{\mathbb{E}_{u|v,\xi}\left[\pi_R(u \cup v, \xi)\, \pi_B(u \cup v, \xi)\right]}{\mathbb{E}_{u|v,\xi}\left[\pi_R(u \cup v, \xi)\right]}\right) \tag{7.11}$$

Finally, we must define a distribution for Π_Q that allows both (7.5) and (7.11) to hold. Note that (7.11) defines a vague function, which can be converted to a distribution over precise functions: π_Q compares (7.11) to a threshold value, returning truth iff it's above the threshold. We have a distribution over these functions, using a uniform distribution over thresholds in $[0, 1]$.

We can now recursively define functions for quantifier nodes, working from the leaves to the root. Given distributions over precise semantic functions (in the leaves), we have corresponding distributions over quantifier functions. We can therefore see Fig. 7.5 as an abbreviated notation for Fig. 7.7. The dotted edges do not indicate conditional dependence of truth values on each other, but dependence of *functions* on each other – in other words, dependence of *distributions* of truth values on each other. Distilling this main idea, we can write the abbreviated equation in (7.12). A quantifier function depends on the restriction and body functions, marginalising out the bound variable.

$$\pi_Q = f_Q\left(\frac{\mathbb{E}_u\left[\pi_R \pi_B\right]}{\mathbb{E}_u\left[\pi_R\right]}\right) \tag{7.12}$$

7.6 Revisiting Logical Inference

In §4.5, I proposed casting logical inference as Bayesian inference about one truth value node, given another truth value node. I suggested that such a conditional probability would still be useful, even if it is never exactly 0 or exactly 1. In this section, I re-analyse these conditional probabilities using the quantification machinery introduced in this chapter. In particular, I will show that these conditional probabilities can be seen as instances of generic quantification.

As defined in §7.4, the generic quantifier GEN does not transform the conditional probability, but just uses the identity function. This means that the vague function t_Q is given by:

$$t_Q(v,\xi) = \mathbb{E}_{\pi_R,\pi_B}\left[\frac{\mathbb{E}_{u|v,\xi}\left[\pi_R(u \cup v,\xi)\,\pi_B(u \cup v,\xi)\right]}{\mathbb{E}_{u|v,\xi}\left[\pi_R(u \cup v,\xi)\right]}\right] \tag{7.13}$$

Note that the distributions over precise semantic functions are independent of the distribution over situations. In other words, we can reverse the order of expectations between $\mathbb{E}_\pi$ and $\mathbb{E}_{u|v,\xi}$. It is therefore tempting to write (7.14), which would allow us to express the vague function for a generic quantifier in terms of the vague functions for its restriction and body, as shown in (7.15).

$$t_Q(v,\xi) = \frac{\mathbb{E}_{u|v,\xi}\mathbb{E}_{\pi_R,\pi_B}\left[\pi_R(u \cup v,\xi)\,\pi_B(u \cup v,\xi)\right]}{\mathbb{E}_{u|v,\xi}\mathbb{E}_{\pi_R}\left[\pi_R(u \cup v,\xi)\right]} \tag{7.14}$$

$$= \frac{\mathbb{E}_{u|v,\xi}\left[t_R(u \cup v,\xi)\,t_B(u \cup v,\xi)\right]}{\mathbb{E}_{u|v,\xi}\left[t_R(u \cup v,\xi)\right]} \tag{7.15}$$

Because of the fraction, (7.13) and (7.15) are not quite the same. For example, consider the generic proposition *a's are b's*, and consider a uniform distribution over just two situations (with a single pixie node X). In one situation, predicates *a* and *b* are certainly true of X, while in the other situation, *b* has a $\frac{1}{2}$ chance of truth, while *b* is certainly false. Under (7.13), the proposition has a $\frac{3}{4}$ chance of truth, but under (7.15), it has a $\frac{2}{3}$ chance.

Given a distribution over a large number of situations, or given predicates with probabilities of truth close to 0 or 1, the two probabilities will be very similar, and so they make largely the same predictions. Now, the decision to model generics as in (7.13) was driven by the intuition that generics are vague but semantically simple. The alternative definition in (7.15) is arguably even simpler, because it allows us to directly compose vague functions – this is computationally convenient, because we only need to calculate $\mathbb{E}_u$ once in total, rather than once for each possible π_R and π_B. Distilling this idea, we can write the abbreviated equation in (7.16), which can be compared with (7.12). To put it another way, a vague quantifier doesn't need to use precise functions.

$$t_Q = \frac{\mathbb{E}_u\left[t_R t_B\right]}{\mathbb{E}_u\left[t_R\right]} \tag{7.16}$$

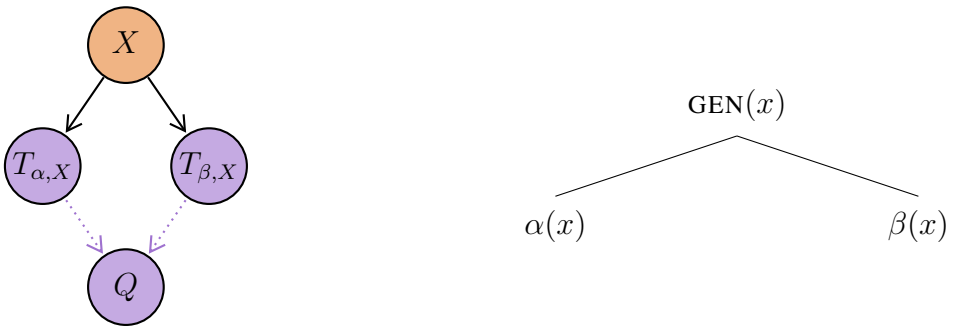


Figure 7.9: Re-analysis of Fig. 4.2 as generic quantification.

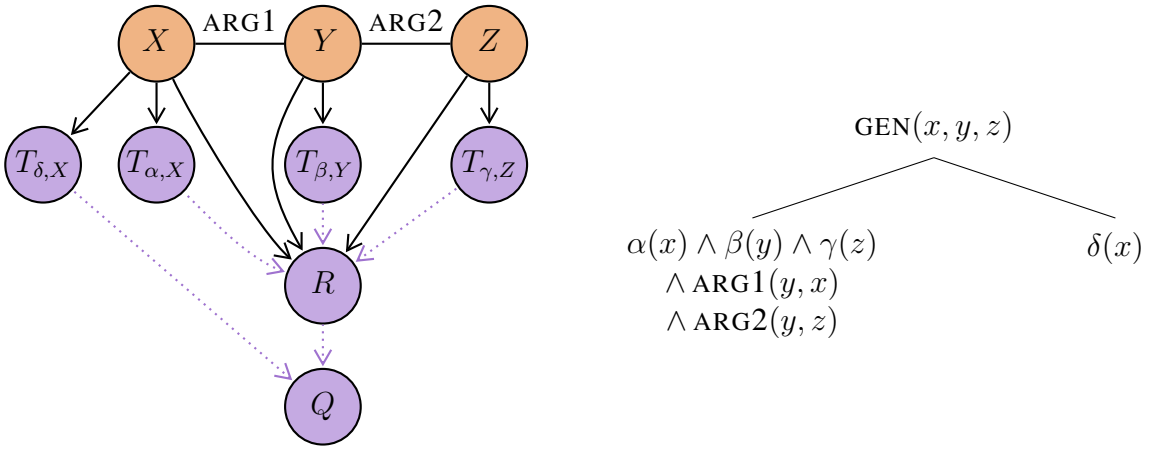


Figure 7.10: Re-analysis of Fig. 4.4 as generic quantification.

Composing vague functions rather than precise functions effectively goes back to the account given in §7.3 and §7.4, before I explicitly introduced precise functions. Indeed, we can rewrite (7.15) as (7.17).

$$t_Q(v, \xi) = \mathbb{P}(b \mid r, v, \xi) \tag{7.17}$$

In the special case where there is only one quantifier in the scope tree, and leaving the supersituation ξ implicit, this reduces to the conditional probability $\mathbb{P}(b \mid r)$. This is simply the probability of one node being true, given that another is true – which is precisely the kind of conditional probability considered in §4.5. In Figs. 7.9 and 7.10, the logical inference shown in Figs. 4.2 and 4.4 has been recast as generic quantification, where there is only one quantifier in the scope tree. Note that in Fig. 7.10, all of the variables are quantified at once – the R node combines the three semantic functions into a function of a whole subsituation, which acts as the restriction of the quantifier node Q. The fact that there is only one quantifier means that we can evaluate the whole scope tree using a single conditional probability.

By viewing these conditional probabilities as generic quantification, we can see that the bottom-up exploration of §4.5 and the top-down approach of this chapter have met in the middle.

Chapter 8

Conclusion

In this thesis, I have presented a novel framework for distributional semantics, and I have used it to tackle both theoretical and practical problems. As this thesis straddles linguistics and machine learning, I will discuss the significance of this work from both perspectives.

8.1 Contribution to Linguistics

Although distributional semantic ideas have been proposed at least since Harris (1954) and Firth (1951), it has been difficult to formalise distributional semantics in a way that is compatible with model-theoretic semantics. While the rise of vector space models has popularised distributional semantics, the vectors produced are not easily interpretable, and hence struggle to account for various aspects of meaning (see Chapter 2).

The main contribution of this thesis is a framework for distributional semantics which is compatible with model theory (see §3.6 and Chapter 5). This allows the framework to combine the strengths of model theory in representing semantic structure, with the strengths of distributional semantics in learning detailed lexical knowledge. As a result, it can meet a range of goals for a theory of semantics (see §3.7).

Developing a theoretical foundation for this distributional semantic framework has required extending classical model theory. In particular, if we take learnability to be an important goal for a semantic theory, then this means taking *generalisation* seriously. This means that it is necessary to make a clear distinction between an *individual* in a model structure and a *pixie*, which represents the features of a possible individual (see §3.2). The notion of a pixie creates a clear distinction between an extension and a truth-conditional function, and makes it easier to discuss generalising a predicate to new situations. Armed with the notion of a pixie, it is also easier to define a probabilistic generalisation of a model structure (see §3.3). While there is other work on probabilistic semantics, the use of a simple graphical model (see §3.5) is what made it possible to use this model for distributional semantics.

One linguistically important component of the model is a lexicon composed of *semantic functions*, which are probabilistic truth-conditional functions (see §3.4). I have examined the nature of these probabilities, connecting them to work in philosophy of language, and arguing that they represent uncertainty about generalising a linguistic convention (see §3.4.1). Furthermore, I have shown how such functions (when equipped with a covariance function) are equivalent to distributions over regions of space, thereby uniting two different views of concepts (see §3.4.1). Finally, implementing a semantic function in a high-dimensional space has brought to light how the boundaries of a concept need to be fairly sharp in practice, in order for the concept to be useful (see §5.1.2).

Furthermore, I have demonstrated how a probabilistic model is more than just a convenience to accommodate learning, and in fact allows a natural account of context dependence (see §4.1 and §4.2). This account is in the spirit of existing work on Bayesian semantics, but applied to more complex situations than previously discussed. It crucially relies on an interaction between conceptual knowledge and world knowledge, a necessary fact highlighted by previous authors. I have also explained how this kind of context dependence relates to disambiguation (see §4.3) and composition (see §4.4), and I have successfully applied it in practice to real-world datasets (see §6.2.2 and §6.2.3).

Designing a graphical model to generalise a classical model structure has enabled it to support a well-defined logic. I have used Bayesian inference over probabilistic truth values to give a generalisation of not only syllogistic logic (see §4.5), but also first-order logic (see §7.3). Furthermore, by representing quantification in terms of conditional probabilities, I have shown how this gives a natural account of vague quantifiers (see §7.4).

8.2 Contribution to Machine Learning

Distributional vector space models have proved effective for a range of NLP tasks, but they are fundamentally limited, because a vector space does not provide an appropriate structure to capture various aspects of meaning (see Chapter 2). One particular concern is that a vector space does not have any logical structure, which limits the use of distributional vectors in any application that requires planning or reasoning.

The main contribution of this thesis is a framework for distributional semantics which is logically interpretable (see §3.5 and §3.6). By defining a probabilistic generative model which incorporates a latent model structure, it is possible to interpret training the model in terms of learning about what situations exist and learning about how situations are described. This interpretability makes the model better suited for capturing semantics than a vector space model (see §3.7). In order to meet model-theoretic demands, the graphical model has both undirected and directed edges, which is unorthodox, but still well-defined.

In particular, I have represented words as *semantic functions*, which are probabilistic bi-

nary classifiers (see §3.4). Such representations have been proposed in other areas of NLP, but not for distributional semantics. I have also explained how such functions can be used to efficiently parametrise a distribution over regions, by combining them with a covariance function (see §3.4.1). Furthermore, I have shown how the logical interpretability extends beyond the word level, by demonstrating that the framework can support not only syllogistic logic (see §4.5), but also first-order logic (see §7.3), and vague quantifiers (see §7.4).

I have given a concrete implementation of this framework, using a combination of Restricted Boltzmann Machines and feedforward networks (see §5.1). Because this is a new kind of model (a combination of undirected and directed graphical models), I have had to adapt existing learning algorithms. While gradient descent can be used (see §5.2), the main practical challenge is the large number of latent variables. I have presented two approximate inference algorithms, a Markov Chain Monte Carlo method (see §5.3), and a Variational Inference method (see §5.4). In both cases, the fact that the model is split into two halves (which is needed for the connection to model theory) meant that existing techniques could not be directly applied, and as a result, I have introduced additional approximations, in order to make the algorithms tractable.

I have trained the model on WikiWoods, a parsed version of the English Wikipedia (see §6.1). Because a random initialisation led to long training times when using MCMC gradient descent, I have adapted a simple method for producing sparse count vectors as a method for parameter initialisation (see §6.1.3). I have optimised the hyperparameters for this method, finding that the optimal settings for producing parameter vectors are quite different from the optimal settings for producing word vectors.

Finally, for several semantic evaluation datasets, I have either matched or pushed forward the state of the art. I have demonstrated that a functional model outperforms Word2Vec on lexical similarity datasets, and furthermore, that it can strongly distinguish between similarity and relatedness (see §6.2.1). I have demonstrated that a functional model outperforms Word2Vec on calculating lexical similarity in context, and is competitive with state-of-the-art tensorial models (see §6.2.2). And lastly, I have demonstrated that a semantic function model can be used to improve the state of the art on RELPRON, a challenging dataset testing semantic composition (see §6.2.3). This last result is particularly exciting, because using semantic functions improved performance on the *confounders* in the dataset – phrases involving lexical overlap, which have been shown to consistently confuse vector space models.

8.3 Looking Forwards

Read narrowly, this dissertation introduces a linguistically interpretable and computationally tractable framework for learning the meanings of words from text. However, this also represents the basis of a larger research project. I may have taken steps towards the goals outlined in Chapter 2, but there is still more work to be done.

As discussed in §2.1 and §3.7.1, all distributional semantic models come up against the symbol grounding problem – if meanings of words are defined in terms of other words, the definitions are circular. Indeed, people do not learn language from text or speech alone, but also connect words with their sensory perception. With its connection to both machine learning and formal semantics, my framework provides a basis for exploring this problem, as state-of-the-art image processing techniques could be directly incorporated into the semantic functions. The functions could be trained in a semi-supervised fashion – where there is grounded information, pixies can be observed directly and the functions trained using direct supervision; where there is only text, pixies can be treated as latent variables, and the functions trained as described in Chapter 5.

The use of multiple data sources could be further extended to include ontologies like WordNet, and structured knowledge bases. A hyponymy relation in an ontology could be treated as a (soft) constraint on semantic functions, using the approach to hyponymy described in §3.7.2. Meanwhile, for a knowledge base that is structured in terms of entities and relations between them, each entity could be treated as a latent pixie, while each relation could be treated as an observed truth value for a predicate and specific latent pixies. These additional data sources would then provide additional supervision signals.

As discussed in §2.3.3 and §3.7.3, compositionality is an important feature of language. However, many phrases cannot be understood by simply combining their parts. Some expression are completely opaque, and might be treated as single lexical items For example, a *red herring* is neither red, nor a herring. However, the particularly challenging cases are *semi-compositional* For example, a *magic carpet* is both magic and a carpet, but is also able to fly. Semi-compositional constructions are poorly understood, both theoretically and computationally (for example: Sag et al., 2002; Reddy et al., 2011; Vincze, 2012). However, the ubiquity of such expressions means they must be properly accounted for. I will briefly take adjective-noun expressions as an example, using the approach given in Chapter 7. The noun and adjective each have a semantic function, which take different arguments. However, after marginalising out the adjective's event variable, we have composed them into a function of a single argument. The framework could be extended to allow additional constraints to be introduced at this stage. This would naturally allow a continuum between fully compositional expressions (where nothing is added), semi-compositional expressions (where some additional meaning is added), and fully idiomatic expressions (where we replace the composed representation with a new one).

A traditional distinction is that, while semantics deals with literal meanings, pragmatics deals with meanings in context. In most of this thesis, I have concerned myself with semantics in this narrow sense. However, to properly account for context dependence, we need to be able to deal with a larger context, which would take us into the traditional domain of pragmatics. As explained in §7.4, my framework is compatible with the framework of Rational Speech Acts, a Bayesian approach to pragmatics. While most work in RSA uses a hand-written semantic model

for a small domain, my framework could provide a semantic model across a large domain. This would both extend my framework to include pragmatic reasoning, and allow us to explore how these pragmatic models behave when they are scaled up.

Finally, to further improve training efficiency, we could use *amortised* variational inference – rather than optimising the variational approximation separately for each observed DMRS graph, we parametrise a function mapping from DMRS graphs to mean-field distributions over pixies, and optimise this function across all observed DMRS graphs in the training data. This function introduces a second level of approximation (since it may map to a suboptimal mean-field distribution), but it can be calculated much more quickly. One method would be to use a graph-convolutional network, which would allow us to make use of the DMRS topology in a natural way.

Reaching all of the goals given in Chapter 2 would be a breakthrough in computational linguistics and artificial intelligence. The framework I have developed in this thesis provides a basis from which we might hope to reach them.

References

Omri Abend and Ari Rappoport. Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, pages 228–238, 2013. Cited on page 24.

Ernest W. Adams. *A Primer of Probability Logic*. Number 68 in CSLI Lecture Notes. Center for the Study of Language and Information (CSLI) Publications, 1998. Cited on page 39.

Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of the 2009 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), Long Papers*, pages 19–27, 2009. Cited on page 112.

Keith Allan. *Natural language semantics*. Blackwell, 2001. Cited on page 19.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 39–48. Institute of Electrical and Electronics Engineers, 2016a. Cited on page 41.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 1545–1554, 2016b. Cited on page 41.

Antoine Arnauld and Pierre Nicole. *La logique ou l'art de penser*. Published by Jean Guignart, Charles Savreux, and Jean de Lavnay, 1662. Widely known as the *Port-Royal Logic*. Cited on page 29.

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404. 1950. Cited on page 19.

Wirote Aroonmanakun. Thoughts on word and sentence segmentation in Thai. In *Proceedings*

of the 7th International Symposium on Natural Language Processing, pages 85–90, 2007. Cited on page 37.

Ben Athiwaratkun and Andrew Gordon Wilson. Multimodal word distributions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, pages 1645–1656, 2017. Cited on pages 34 and 45.

Ben Athiwaratkun and Andrew Gordon Wilson. Hierarchical density order embeddings. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018. Cited on page 36.

John Langshaw Austin. Truth. In *Physical Research, Ethics and Logic*, volume 24 of *Proceedings of the Aristotelian Society, Supplementary Volumes*, pages 111–128. Oxford University Press, 1950. Reprinted in: Austin (1961/1979), *Philosophical Papers*, editors James Opie Urmson and Geoffrey James Warnock, pages 117–133. Cited on page 22.

Emmon Bach. The algebra of events. *Linguistics and Philosophy*, 9(1):5–16. D. Reidel Publishing Company, 1986. Cited on page 20.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL) and 17th International Conference on Computational Linguistics (COLING)*, pages 86–90, 1998. Cited on page 21.

Esma Balkır. *Using density matrices in a compositional distributional model of meaning*. Master's thesis, University of Oxford, 2014. Cited on pages 32, 34, and 36.

Esma Balkır, Mehrnoosh Sadrzadeh, and Bob Coecke. Distributional sentence entailment using density matrices. In *Proceedings of the 1st International Conference on Topics in Theoretical Computer Science (TTCS)*, pages 1–22. International Federation for Information Processing (IFIP), 2015. Cited on page 36.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, page 178–186, 2013. Cited on page 24.

Oren Barkan. Bayesian neural word embedding. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 3135–3143. Association for the Advancement of Artificial Intelligence, 2017. Cited on pages 32 and 45.

Marco Baroni and Roberto Zamparelli. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 15th Conference*

on Empirical Methods in Natural Language Processing (EMNLP), pages 1183–1193. Association for Computational Linguistics, 2010. Cited on page 38.

Marco Baroni, Raffaela Bernardi, and Roberto Zamparelli. Frege in space: A program of compositional distributional semantics. *Linguistic Issues in Language Technology (LiLT)*, 9. Center for the Study of Language and Information (CSLI) Publications, 2014. Cited on pages 19 and 47.

Lawrence W Barsalou, Léo Dutriaux, and Christoph Scheepers. Moving beyond the distinction between concrete and abstract concepts. *Philosophical Transactions of the Royal Society B*, 373(1752):20170144. The Royal Society, 2018. Cited on page 26.

Jon Barwise and Robin Cooper. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4(2):159–219. D. Reidel Publishing Company, 1981. Cited on page 119.

Jon Barwise and John Etchemendy. *The liar: An essay on truth and circularity*. Oxford University Press, 1987. Cited on page 22.

Jon Barwise and John Perry. *Situations and Attitudes*. Massachusetts Institute of Technology (MIT) Press, 1983. Cited on pages 22, 26, 28, 59, and 68.

Eduard Bejček, Eva Hajičová, Jan Hajič, Pavlína Jínová, Václava Kettnerová, Veronika Kolářová, Marie Mikulová, Jiří Mírovský, Anna Nedoluzhko, Jarmila Panevová, Lucie Poláková, Magda Ševčíková, Jan Štěpánek, and Šárka Zikánová. Prague Dependency Treebank 3.0. Institute of Formal and Applied Linguistics (ÚFAL), Charles University, Czech Republic, 2013. Cited on page 24.

Islam Beltagy, Stephen Roller, Pengxiang Cheng, Katrin Erk, and Raymond J. Mooney. Representing meaning with a combination of logical and distributional models. *Computational Linguistics*, 42(4):763–808. Massachusetts Institute of Technology (MIT) Press, 2016. Cited on pages 40, 47, and 124.

Emily M. Bender, Dan Flickinger, Stephan Oepen, Woodley Packard, and Ann Copestake. Layers of interpretation: On grammar and compositionality. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS)*, pages 239–249. Association for Computational Linguistics, 2015. Cited on page 24.

Richard Bergmair. *Monte Carlo Semantics: Robust inference and logical pattern processing with Natural Language text*. PhD thesis, University of Cambridge, 2010. Cited on page 32.

Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009. Cited on page 105.

William Blacoe and Mirella Lapata. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing (EMNLP) and Computational Natural Language Learning (CoNLL)*, pages 546–556. Association for Computational Linguistics, 2012. Cited on page 38.

Adam B Blake, Meenely Nazarian, and Alan D Castel. The Apple of the mind's eye: Everyday attention, metamemory, and reconstructive memory for the Apple logo. *The Quarterly Journal of Experimental Psychology*, 68(5):858–865. Taylor & Francis, 2015. Cited on page 55.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022. 2003. Cited on pages 27 and 43.

Alexandre Blondin-Massé, Guillaume Chicoisne, Yassine Gargouri, Stevan Harnad, Olivier Picard, and Odile Marcotte. How is meaning grounded in dictionary definitions? In *Proceedings of the 3rd Textgraphs Workshop on Graph-Based Algorithms for Natural Language Processing*, pages 17–24. 22nd International Conference on Computational Linguistics (COLING), 2008. Cited on page 26.

Reinhard Blutner. Lexical pragmatics. *Journal of Semantics*, 15(2):115–162. Oxford University Press, 1998. Cited on page 42.

Gemma Boleda and Aurélie Herbelot. Formal distributional semantics: Introduction to the special issue. *Computational Linguistics*, 42(4):619–635. Massachusetts Institute of Technology (MIT) Press, 2017. Cited on page 15.

Georgeta Bordea, Els Lefever, and Paul Buitelaar. SemEval-2016 task 13: Taxonomy extraction evaluation (TExEval-2). In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval)*, pages 1081–1091. Association for Computational Linguistics, 2016. Cited on page 35.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 20th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 632–642. Association for Computational Linguistics, 2015. Cited on page 40.

Robert Boyce Brandom. *Articulating Reasons: An Introduction to Inferentialism*. Harvard University Press, 2000. Cited on pages 26 and 39.

Arthur Bražinskas, Serhii Havrylov, and Ivan Titov. Embedding words as distributions with a Bayesian skip-gram model. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, pages 1775–1789. International Committee on Computational Linguistics (ICCL), 2018. Cited on pages 43, 44, and 45.

Elia Bruni, Giang Binh Tran, and Marco Baroni. Distributional semantics from text and images. In *Proceedings of GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 22–32. Association for Computational Linguistics, 2011. Cited on page 27.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *Journal of Artificial Intelligence Research (JAIR)*, 49(2014):1–47. 2014. Cited on pages 27 and 112.

Luana Bulat, Douwe Kiela, and Stephen Clark. Vision and feature norms: Improving automatic feature norm learning through cross-modal maps. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 579–588, 2016. Cited on page 27.

Jan Buys and Phil Blunsom. Robust incremental neural semantic graph parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, pages 1215–1226, 2017. Cited on page 24.

Ulrich Callmeier. *Efficient parsing with large-scale unification grammars*. Master's thesis, Saarland University, 2001. Cited on page 103.

Jose Camacho-Collados, Claudio Delli Bovi, Luis Espinosa Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion. SemEval-2018 task 9: Hypernym discovery. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval)*, pages 712–724. Association for Computational Linguistics, 2018. Cited on page 35.

Ronnie Cann. *Formal semantics: an introduction*. Cambridge University Press, 1993. Cited on page 19.

Gregory N. Carlson. *Reference to kinds in English*. PhD thesis, University of Massachusetts at Amherst, 1977. Cited on page 126.

Gregory N. Carlson and Francis Jeffry Pelletier, editors. *The generic book*. University of Chicago Press, 1995. Cited on page 126.

Rudolf Carnap. *Meaning and Necessity: A Study in Semantics and Modal Logic*. University of Chicago Press, 1947. Cited on page 29.

Baobao Chang, Wenzhe Pei, and Miaohong Chen. Inducing word sense with automatically learned hidden concepts. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pages 355–364. International Committee on Computational Linguistics (ICCL), 2014. Cited on page 43.

Yufei Chen, Weiwei Sun, and Xiaojun Wan. Accurate SHRG-based semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, pages 408–418, 2018. Cited on page 24.

Jianpeng Cheng, Li Dong, and Mirella Lapata. Long Short-Term Memory-networks for machine reading. In *Proceedings of the 21st Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 551–561. Association for Computational Linguistics, 2016. Cited on page 40.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 19th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics, 2014. Cited on page 38.

Kenneth Ward Church. A pendulum swung too far. *Linguistic Issues in Language Technology (LiLT)*, 6(5):1–27. Center for the Study of Language and Information (CSLI) Publications, 2011. Cited on page 15.

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29. Massachusetts Institute of Technology (MIT) Press, 1990. Cited on page 17.

Petr Cintula, Christian G. Fermüller, and Carles Noguera. Fuzzy logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall edition, 2017. Cited on page 40.

Stephen Clark. Vector space models of lexical meaning. In Shalom Lappin and Chris Fox, editors, *The Handbook of Contemporary Semantic Theory*, chapter 16, pages 493–522. Wiley, 2nd edition, 2015. Cited on page 17.

Stephen Clark, Laura Rimell, Tamara Polajnar, and Jean Maillard. The categorial framework for compositional distributional semantics. Unpublished draft, 2016. Cited on page 48.

Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis*, 36, A Festschrift for Joachim Lambek:345–384. 2010. Cited on pages 19 and 47.

Robin Cooper. Austinian truth, attitudes and type theory. *Research on Language and Computation*, 3(2-3):333–362. Springer, 2005. Cited on page 48.

Robin Cooper, Simon Dobnik, Staffan Larsson, and Shalom Lappin. Probabilistic type theory and natural language semantics. *Linguistic Issues in Language Technology (LiLT)*, 10. Center

for the Study of Language and Information (CSLI) Publications, 2015. Cited on pages 48, 68, and 124.

Ann Copestake. Semantic composition with (Robust) Minimal Recursion Semantics. In *Proceedings of the ACL 2007 Workshop on Deep Linguistic Processing*, pages 73–80. Association for Computational Linguistics, 2007. Cited on pages 37 and 74.

Ann Copestake. Slacker semantics: Why superficiality, dependency and avoidance of commitment can be the right way to go. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 1–9, 2009. Cited on pages 21 and 23.

Ann Copestake and Aurélie Herbelot. Lexicalised compositionality. Unpublished draft, 2012. Cited on pages 41, 67, and 78.

Ann Copestake, Alex Lascarides, and Dan Flickinger. An algebra for semantic construction in constraint-based grammars. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 140–147, 2001. Cited on pages 37 and 74.

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A. Sag. Minimal Recursion Semantics: An introduction. *Research on Language and Computation*, 3(2-3):281–332. Springer, 2005. Cited on pages 23 and 121.

Ann Copestake, Guy Emerson, Michael Wayne Goodman, Matic Horvat, Alexander Kuhnle, and Ewa Muszyńska. Resources for building applications with Dependency Minimal Recursion Semantics. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, pages 1240–1247. European Language Resources Association (ELRA), 2016. Cited on page 104.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297. Springer, 1995. Cited on page 19.

Ryan Cotterell, Adam Poliak, Benjamin Van Durme, and Jason Eisner. Explaining and generalizing Skip-Gram through exponential family principal component analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Short Papers*, pages 175–181, 2017. Cited on page 18.

Donald Davidson. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*, chapter 3, pages 81–95. University of Pittsburgh Press, 1967. Reprinted in: Davidson (1980/2001), *Essays on Actions and Events*, Oxford University Press. Cited on page 21.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. Universal Stanford Dependencies: A cross-linguistic typology. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, pages 4585–4592. European Language Resources Association (ELRA), 2014. Cited on page 24.

Ferdinand de Saussure. *Cours de linguistique générale (Course in general linguistics)*. Edited and published by Charles Bally and Albert Sechehaye, University of Geneva, 1916. Cited on page 59.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407. Wiley, 1990. Cited on page 45.

Lorenz Demey, Barteld Kooi, and Joshua Sack. Logic and probability. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring edition, 2013. The summer 2017 edition contains minor corrections. Cited on page 39.

Keith Devlin. Situation theory and situation semantics. In Dov M. Gabbay and John Woods, editors, *Logic and the Modalities in the Twentieth Century*, volume 7 of *Handbook of the History of Logic*, chapter 8, pages 601–664. Elsevier, 2006. Cited on page 22.

Georgiana Dinu and Mirella Lapata. Measuring distributional similarity in context. In *Proceedings of the 15th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1162–1172. Association for Computational Linguistics, 2010. Cited on page 43.

Georgiana Dinu, Stefan Thater, and Sören Laue. A comparison of models of word meaning in context. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 611–615, 2012. Cited on page 43.

David R. Dowty. Thematic proto-roles and argument selection. *Language*, pages 547–619. Linguistic Society of America, 1991. Cited on pages 21 and 49.

David R. Dowty, Robert E. Wall, and Stanley Peters. *Introduction to Montague Semantics*, volume 11 of *Studies in Linguistics and Philosophy*. Kluwer Academic Publishers, 1981. Cited on page 19.

David Dubin. The most influential paper Gerard Salton never wrote. *Library Trends*, 52(4): 748–764. John Hopkins University Press, 2004. Cited on page 18.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159. 2011. Cited on page 107.

Michael Dummett. What is a theory of meaning? (II). In Gareth Evans and John McDowell, editors, *Truth and Meaning: Essays in Semantics*, chapter 4, pages 67–137. Clarendon Press (Oxford), 1976. Reprinted in: Dummett (1993), *Seas of Language*, chapter 2, pages 34–93. Cited on page 30.

Michael Dummett. What do I know when I know a language? Presented at the Centenary Celebrations of Stockholm University, 1978. Reprinted in: Dummett (1993), *Seas of Language*, chapter 3, pages 94–105. Cited on page 30.

Jeffrey L. Elman. On the meaning of words and dinosaur bones: Lexical knowledge without a lexicon. *Cognitive Science*, 33(4):547–582. Wiley, 2009. Cited on pages 42, 70, 71, and 72.

Guy Emerson and Ann Copestake. Lacking integrity: HPSG as a morphosyntactic theory. In *Proceedings of the 22nd Annual Conference on Head-Driven Phrase Structure Grammar*, pages 75–95. Center for the Study of Language and Information (CSLI) Publications, 2015. Cited on page 31.

Guy Emerson and Ann Copestake. Functional Distributional Semantics. In *Proceedings of the 1st Workshop on Representation Learning for NLP (RepL4NLP)*, pages 40–52. Association for Computational Linguistics, 2016. Cited on pages 14, 51, and 111.

Guy Emerson and Ann Copestake. Variational inference for logical inference. In *Proceedings of the Conference on Logic and Machine Learning in Natural Language (LaML)*, pages 53–62. Centre for Linguistic Theory and Studies in Probability (CLASP), 2017a. Cited on page 14.

Guy Emerson and Ann Copestake. Semantic composition via probabilistic model theory. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS)*, pages 62–77. Association for Computational Linguistics, 2017b. Cited on pages 14 and 111.

Katrin Erk. Supporting inferences in semantic space: representing words as regions. In *Proceedings of the 8th International Conference on Computational Semantics (IWCS)*, pages 104–115. Association for Computational Linguistics, 2009a. Cited on pages 30, 36, and 46.

Katrin Erk. Representing words as regions in vector space. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL)*, pages 57–65. Association for Computational Linguistics, 2009b. Cited on pages 30, 36, and 46.

Katrin Erk. What is word meaning, really? (and how can distributional models help us describe it?). In *Proceedings of the 2010 Workshop on Geometrical Models of Natural Language Semantics*, pages 17–26. Association for Computational Linguistics, 2010. Cited on page 33.

Katrin Erk. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653. Blackwell, 2012. Cited on page 17.

Katrin Erk. What do you know about an alligator when you know the company it keeps? *Semantics and Pragmatics*, 9(17):1–63. 2016. Cited on pages 41 and 47.

Katrin Erk and Sebastian Padó. A structured vector space model for word meaning in context. In *Proceedings of the 13th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 897–906. Association for Computational Linguistics, 2008. Cited on page 43.

Katrin Erk and Sebastian Padó. Exemplar-based models for word meaning in context. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL), Short Papers*, pages 92–97, 2010. Cited on page 43.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. Investigations on word senses and word usages. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (IJCNLP), Long Papers*, pages 10–18, 2009. Cited on page 33.

Katrin Erk, Diana McCarthy, and Nicholas Gaylord. Measuring word meaning in context. *Computational Linguistics*, 39(3):511–554. Massachusetts Institute of Technology (MIT) Press, 2013. Cited on page 33.

Hugo Jair Escalante, Carlos A Hernández, Jesus A Gonzalez, Aurelio López-López, Manuel Montes, Eduardo F Morales, L Enrique Sucar, Luis Villaseñor, and Michael Grubinger. The segmented and annotated IAPR TC-12 benchmark. *Computer Vision and Image Understanding*, 114(4):419–428. Elsevier, 2010. Cited on page 44.

Luana Făgărăşan. From distributional semantics to feature norms: grounding semantic models in human perceptual data. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS)*, pages 52–57. Association for Computational Linguistics, 2015. Cited on page 27.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah Smith. Sparse overcomplete word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, pages 1491–1500, 2015. Cited on page 84.

Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database* [Website]. Massachusetts Institute of Technology (MIT) Press, 1998. Cited on pages 36 and 105.

Yansong Feng and Mirella Lapata. Visual information in semantic representation. In *Proceedings of the 2010 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 91–99, 2010. Cited on pages 27 and 28.

Raquel Fernández and Staffan Larsson. Vagueness and learning: A type-theoretic approach. In *Proceedings of the 3rd Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 151–159. Association for Computational Linguistics, 2014. Cited on pages 32 and 68.

Kit Fine. Vagueness, truth and logic. *Synthese*, 30(3-4):265–300. D. Reidel Publishing Company, 1975. Cited on pages 40 and 59.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on the World Wide Web*, pages 406–414. Association for Computing Machinery, 2001. Cited on page 112.

John Rupert Firth. Modes of meaning. *Essays and Studies of the English Association*, 4:118–149. 1951. Reprinted in: Firth (1957), *Papers in Linguistics*, chapter 15, pages 190–215. Cited on pages 16 and 133.

John Rupert Firth. A synopsis of linguistic theory 1930–1955. In John Rupert Firth, editor, *Studies in Linguistic Analysis*, Special volume of the Philological Society, chapter 1, pages 1–32. Blackwell, 1957. Cited on page 16.

Dan Flickinger. On building a more efficient grammar by exploiting types. *Natural Language Engineering*, 6(1):15–28. Cambridge University Press, 2000. Cited on pages 21 and 103.

Dan Flickinger. Accuracy vs. robustness in grammar engineering. In Emily M. Bender and Jennifer E. Arnold, editors, *Language from a Cognitive Perspective: Grammar, Usage, and Processing*, chapter 3, pages 31–50. Center for the Study of Language and Information (CSLI) Publications, 2011. Cited on pages 21 and 103.

Dan Flickinger, Stephan Oepen, and Gisle Ytrestøl. WikiWoods: Syntacto-semantic annotation for English Wikipedia. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*, pages 1665–1671. European Language Resources Association (ELRA), 2010. Cited on pages 21 and 103.

Dan Flickinger, Emily M. Bender, and Stephan Oepen. Towards an encyclopedia of compositional semantics: Documenting the interface of the English Resource Grammar. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, pages 875–881. European Language Resources Association (ELRA), 2014. Cited on page 24.

Michael C. Frank and Noah D. Goodman. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998. American Association for the Advancement of Science, 2012. Cited on page 126.

Gottlob Frege. Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100:25–50. 1892. Usually translated into English under the title *On Sense and Reference*. Cited on page 28.

Daniel Fried, Tamara Polajnar, and Stephen Clark. Low-rank tensors for verbs in compositional distributional semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL), Short Papers*, pages 731–736, 2015. Cited on page 45.

Mohan Ganesalingam and Aurélie Herbelot. Composing distributions: mathematical structures and their linguistic interpretation. Unpublished draft, 2013. Cited on page 38.

Peter Gärdenfors. *Conceptual spaces: The geometry of thought*. Massachusetts Institute of Technology (MIT) Press, 2000. Cited on pages 30, 51, and 58.

Peter Gärdenfors. *Geometry of meaning: Semantics based on conceptual spaces*. Massachusetts Institute of Technology (MIT) Press, 2014. Cited on pages 30, 36, 51, and 58.

Dan Garrette, Katrin Erk, and Raymond Mooney. Integrating logical representations with probabilistic information using Markov logic. In *Proceedings of the 9th International Conference on Computational Semantics (IWCS)*, pages 105–114. Association for Computational Linguistics, 2011. Cited on pages 40 and 41.

Maayan Geffet and Ido Dagan. The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 107–114, 2005. Cited on page 35.

Samuel J. Gershman and David M. Blei. A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12. Elsevier, 2012. Cited on page 44.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. SimVerb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 21st Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2173–2182. Association for Computational Linguistics, 2016. Cited on page 112.

Zoubin Ghahramani. Unsupervised learning. In Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch, editors, *Advanced Lectures on Machine Learning*, number 3176 in Lecture Notes in Computer Science, chapter 5, pages 72–112. Springer, 2004. Cited on page 63.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Massachusetts Institute of Technology (MIT) Press, 2016. Cited on page 39.

Noah D. Goodman and Michael C. Frank. Pragmatic language interpretation as probabilistic inference. *Trends in cognitive sciences*, 20(11):818–829. Elsevier, 2016. Cited on pages 42 and 126.

Noah D. Goodman and Daniel Lassiter. Probabilistic semantics and pragmatics: Uncertainty in language and thought. In Shalom Lappin and Chris Fox, editors, *The Handbook of Contemporary Semantic Theory*, chapter 21, pages 655–686. Wiley, 2nd edition, 2015. Cited on pages 48 and 68.

Edward Grefenstette. Towards a formal distributional semantics: Simulating logical calculi with tensors. In *Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 1–10. Association for Computational Linguistics, 2013. Cited on pages 41 and 124.

Edward Grefenstette and Mehrnoosh Sadrzadeh. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the 16th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1394–1404. Association for Computational Linguistics, 2011. Cited on page 114.

Edward Grefenstette, Georgiana Dinu, Yao-Zhong Zhang, Mehrnoosh Sadrzadeh, and Marco Baroni. Multi-step regression learning for compositional distributional semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS)*, pages 131–142. Association for Computational Linguistics, 2013. Cited on page 115.

Paul Grice. Logic and conversation. William James Lecture, Harvard University, 1967. Reprinted in: Peter Cole and Jerry Morgan, editors (1975), *Syntax and Semantics 3: Speech Acts*, chapter 2, pages 41–58; Donald Davidson and Gilbert Harman, editors (1975), *The Logic of Grammar*, chapter 6, pages 64–74; Grice (1989), *Studies in the Way of Words*, chapter 2, pages 22–40. Cited on page 35.

Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(supplement 1):5228–5235. 2004. Cited on page 93.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), Short Papers*, pages 107–112, 2018. Cited on page 40.

Petr Hájek. *Metamathematics of Fuzzy Logic*. Number 4 in Trends in Logic. Kluwer Academic Publishers, 1998. Cited on page 40.

Petr Hájek, Lluís Godo, and Francesc Esteva. Fuzzy logic and probability. In *Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence*, pages 237–244. Morgan Kaufmann Publishers Inc., 1995. Cited on page 40.

Eva Hajičová. Prague Dependency Treebank: From analytic to tectogrammatical annotations. In *Proceedings of the 1st Workshop on Text, Speech, Dialogue*, pages 45–50, 1998. Cited on page 24.

Patrick Hanks. Do word meanings exist? *Computers and the Humanities*, 34, Special Issue on SENSEVAL: Evaluating Word Sense Disambiguation Programs:205–215. Kluwer Academic Publishers, 2000. Cited on page 33.

Stevan Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42:335–346. Elsevier, 1990. Cited on page 26.

Zellig Sabbetai Harris. Distributional structure. *Word*, 10:146–162. Linguistic Circle of New York, 1954. Reprinted in: Harris (1970), *Papers in Structural and Transformational Linguistics*, chapter 36, pages 775–794; Harris (1981), *Papers on Syntax*, chapter 1, pages 3-22. Cited on pages 16, 19, and 133.

Martin Haspelmath. The indeterminacy of word segmentation and the nature of morphology and syntax. *Folia Linguistica*, 45(1):31–80. Walter de Gruyter, 2011. Cited on page 31.

W. Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109. Biometrika Trust, 1970. Cited on page 94.

Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING)*, pages 539–545. International Committee on Computational Linguistics (ICCL), 1992. Cited on page 35.

Jules Hedges and Mehrnoosh Sadrzadeh. A generalised quantifier theory of natural language in categorical compositional distributional semantics with bialgebras. Unpublished draft, 2017. Cited on page 48.

Aurélie Herbelot. *Underspecified quantification*. PhD thesis, University of Cambridge, 2010. Cited on page 126.

Aurélie Herbelot. What is in a text, what isn't, and what this has to do with lexical semantics. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS)*, pages 321–327. Association for Computational Linguistics, 2013. Cited on page 117.

Aurélie Herbelot. Mr Darcy and Mr Toad, gentlemen: distributional names and their kinds. In *Proceedings of the 11th International Conference on Computational Semantics (IWCS)*, pages 151–161. Association for Computational Linguistics, 2015. Cited on page 43.

Aurélie Herbelot and Mohan Ganesalingam. Measuring semantic content in distributional vectors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL), Short Papers)*, pages 440–445, 2013. Cited on page 35.

Aurélie Herbelot and Eva Maria Vecchi. Building a shared world: mapping distributional to model-theoretic semantic spaces. In *Proceedings of the 20th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 22–32. Association for Computational Linguistics, 2015. Cited on pages 40 and 124.

Aurélie Herbelot and Eva Maria Vecchi. Many speakers, many worlds: Interannotator variations in the quantification of feature norms. *Linguistic Issues in Language Technology (LiLT)*, 13. Center for the Study of Language and Information (CSLI) Publications, 2016. Cited on page 117.

Karl Moritz Hermann and Phil Blunsom. The role of syntax in vector space models of compositional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, pages 894–904, 2013. Cited on page 38.

Annette Herskovits. *Language and spatial cognition: An interdisciplinary study of the prepositions in English*. Cambridge University Press, 1986. Cited on page 74.

Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695. Massachusetts Institute of Technology (MIT) Press, 2015. Cited on pages 112 and 113.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800. Massachusetts Institute of Technology (MIT) Press, 2002. Cited on page 93.

Geoffrey E. Hinton. A practical guide to training Restricted Boltzmann Machines. Technical Report 2010-003, University of Toronto Machine Learning, 2010. Cited on pages 85 and 93.

Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554. Massachusetts Institute of Technology (MIT) Press, 2006. Cited on page 85.

Wilfrid Hodges. Tarski's truth definitions. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall edition, 2014. Cited on page 19.

Manuela Hürlimann and Johan Bos. Combining lexical and spatial knowledge to predict spatial relations between objects in images. In *Proceedings of the 5th Workshop on Vision and Language*, pages 10–18. Association for Computational Linguistics, 2016. Cited on page 44.

Dominic Hyde and Diana Raffman. Sorites paradox. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer edition, 2018. Cited on page 31.

Thomas F. Icard. *The Algorithmic Mind: A Study of Inference in Action*. PhD thesis, Stanford University, 2014. Institute for Logic, Language and Computation (ILLC) Dissertation Series, DS-2014-02. Cited on page 39.

Emily Elizabeth Constance Jones. *A New Law of Thought and its Logical Bearings*. Cambridge University Press, 1911. Cited on page 29.

Gregory V Jones. Misremembering a common object: When left is not right. *Memory & Cognition*, 18(2):174–182. Psychonomic Society, Springer, 1990. Cited on page 55.

Hans Kamp and Barbara Partee. Prototype theory and compositionality. *Cognition*, 57(2): 129–191. Elsevier, 1995. Cited on page 42.

Hans Kamp and Uwe Reyle. *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*, volume 42 of *Studies in Linguistics and Philosophy*. Springer, 2013. Cited on page 19.

David Kaplan. On the logic of demonstratives. *Journal of Philosophical Logic*, 8(1):81–98. Springer, 1979. Cited on pages 42 and 73.

David Kaplan. Demonstratives: An essay on the semantics, logic, metaphysics, and epistemology of demonstratives and other indexicals. In Joseph Almog, John Perry, and Howard Wettstein, editors, *Themes from Kaplan*, chapter 17, pages 481–563. Oxford University Press, 1989. Cited on pages 42 and 73.

Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Stephen Pulman. Separating disambiguation from composition in distributional semantics. In *Proceedings of the 17th Conference on Computational Natural Language Learning (CoNLL)*, pages 114–123. Association for Computational Linguistics, 2013. Cited on page 37.

Martin Kay. Does a computational linguist have to be a linguist? Invited talk at the 25th International Conference on Computational Linguistics (COLING), 2014. Cited on page 15.

Paul Kay, Brent Berlin, Luisa Maffi, and William Merrifield. Color naming across languages. In Clyde Laurence Hardin and Luisa Maffi, editors, *Color Categories in Thought and Language*, chapter 2, pages 21–56. Cambridge University Press, 1997. Cited on page 58.

Rosanna Keefe. *Theories of Vagueness*. Cambridge Studies in Philosophy. Cambridge University Press, 2000. Cited on page 59.

Anthony Kenny. Concepts, brains, and behaviour. *Grazer Philosophische Studien*, 81(1):105–113. 2010. Cited on page 30.

Douwe Kiela and Stephen Clark. Learning neural audio embeddings for grounding semantics in auditory perception. *Journal of Artificial Intelligence Research*, 60:1003–1030. 2017. Cited on page 27.

Douwe Kiela, Luana Bulat, and Stephen Clark. Grounding semantics in olfactory perception. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL), Short Papers*, pages 231–236, 2015. Cited on page 27.

Adam Kilgarriff. I don't believe in word senses. *Computers and the Humanities*, 31(2):91–113. Kluwer Academic Publishers, 1997. Cited on page 33.

Adam Kilgarriff. Word senses. In Eneko Agirre and Philip Edmonds, editors, *Word Sense Disambiguation: Algorithms and Applications*, chapter 2, pages 29–46. Springer, 2007. Cited on page 33.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015. Cited on page 107.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. In *Proceedings of the NIPS 2014 Deep Learning and Representation Learning Workshop*, 2014. Cited on pages 27 and 28.

Alexander Koller. Top-down and bottom-up views on success in semantics. Invited talk at the 5th Joint Conference on Lexical and Computational Semantics (*SEM), 2016. Cited on page 25.

Alexander Koller and Stefan Thater. The evolution of dominance constraint solvers. In *Proceedings of the 2005 ACL Workshop on Software*, pages 65–76. Association for Computational Linguistics, 2005. Cited on page 121.

Alexander Koller and Stefan Thater. An improved redundancy elimination algorithm for underspecified representations. In *Proceedings of the 21st International Conference on Computational Linguistics (COLING) and the 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 409–416, 2006. Cited on page 121.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. Directional distributional similarity for lexical expansion. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (IJCNLP), Short Papers*, pages 69–72, 2009. Cited on page 35.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389. Cambridge University Press, 2010. Cited on page 35.

Angelika Kratzer. Situations in natural language semantics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter edition, 2017. Cited on pages 22 and 127.

Jayant Krishnamurthy and Tom Mitchell. Vector space semantic parsing: A framework for compositional vector space models. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 1–10. Association for Computational Linguistics, 2013. Cited on page 48.

Marco Kuhlmann and Stephan Oepen. Towards a catalogue of linguistic graph banks. *Computational Linguistics*, 42(4):819–827. Massachusetts Institute of Technology (MIT) Press, 2016. Cited on page 24.

William Labov. The boundaries of words and their meanings. In Charles-James Bailey and Roger W. Shuy, editors, *New Ways of Analyzing Variation in English*, chapter 24, pages 340–371. Georgetown University Press, 1973. Reprinted in: Ralph W. Fasold, editor (1983), *Variation in the Form and Use of Language: A Sociolinguistics Reader*, chapter 3, pages 29–62, Georgetown University Press. Cited on page 31.

Ran Lahav. Against compositionality: the case of adjectives. *Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition*, 57(3):261–279. Springer, 1989. Cited on page 42.

Ran Lahav. The combinatorial-connectionist debate and the pragmatics of adjectives. *Pragmatics & Cognition*, 1(1):71–88. John Benjamins Publishing Company, 1993. Cited on page 42.

Thomas K Landauer and Susan T Dumais. A solution to Plato's problem: The Latent Semantic Analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211–240. American Psychological Association, 1997. Cited on page 45.

Gabriella Lapesa and Stefan Evert. A large scale evaluation of distributional semantic models: Parameters, interactions and model selection. *Transactions of the Association for Computational Linguistics (TACL)*, 2:531–545. 2014. Cited on page 17.

Staffan Larsson. Formal semantics for perceptual classification. *Journal of Logic and Computation*, 25(2):335–369. Oxford University Press, 2013. Cited on pages 30 and 68.

Daniel Lassiter. Vagueness as probabilistic linguistic knowledge. In Rick Nouwen, Robert van Rooij, Uli Sauerland, and Hans-Christian Schmitz, editors, *Vagueness in Communication: Revised Selected Papers from the 2009 International Workshop on Vagueness in Communication*, chapter 8, pages 127–150. Springer, 2011. Cited on pages 32 and 59.

Daniel Lassiter. Bayes nets and the dynamics of probabilistic language. In *Proceedings of Sinn and Bedeutung 21*, 2017. Cited on page 53.

Daniel Lassiter and Noah D. Goodman. Adjectival vagueness in a Bayesian model of interpretation. *Synthese*, 194(10):3801–3836. Springer, 2015. Cited on pages 42, 44, and 60.

Rebecca Lawson. The science of cycology: Failures to understand how everyday objects work. *Memory & Cognition*, 34(8):1667–1675. Psychonomic Society, Springer, 2006. Cited on page 55.

Angeliki Lazaridou, Elia Bruni, and Marco Baroni. Is this a wampimuk? cross-modal mapping between distributional semantics and the visual world. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, pages 1403–1414, 2014. Cited on page 27.

Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. Combining language and vision with a multimodal Skip-gram model. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 153–163, 2015. Cited on pages 27 and 28.

Alessandro Lenci. Distributional semantics in linguistic and cognitive research. *Italian Journal of Linguistics*, 20(1):1–31. 2008. Cited on page 16.

Alessandro Lenci. Distributional models of word meaning. *Annual Review of Linguistics*, 4: 151–171. 2018. Cited on page 16.

Sarah-Jane Leslie. Generics: Cognition and acquisition. *Philosophical Review*, 117(1):1–47. Duke University Press, 2008. Cited on page 126.

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 2177–2185, 2014. Cited on page 18.

Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics (TACL)*, 3:211–225. 2015a. Cited on pages 18, 109, 110, 113, and 114.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 970–976, 2015b. Cited on page 36.

David Lewis. General semantics. *Synthese*, 22:18–67. D. Reidel Publishing Company, 1970. Cited on page 26.

Mike Lewis and Mark Steedman. Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics (TACL)*, 1:179–192. 2013. Cited on pages 41, 47, and 124.

Xiang Li, Luke Vilnis, and Andrew McCallum. Improved representation learning for predicting commonsense ontologies. In *Proceedings of the ICML 17 Workshop on Deep Structured Prediction*, 2017. Cited on page 36.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proceedings of the 13th European Conference on Computer Vision, Part V*, pages 740–755. Springer, 2014. Cited on page 44.

Godehard Link. The logical analysis of plurals and mass terms: A lattice-theoretical approach. In Rainer Bäuerle, Christoph Schwarze, and Arnim von Stechow, editors, *Meaning, Use and the Interpretation of Language*, chapter 18, pages 303–323. Walter de Gruyter, 1983. Reprinted in: Paul Portner and Barbara H. Partee, editors (2002), *Formal semantics: The essential readings*, chapter 4, pages 127–146. Cited on page 124.

Marco Lui, Timothy Baldwin, and Diana McCarthy. Unsupervised estimation of word usage similarity. In *Proceedings of the 10th Australasian Language Technology Association Workshop (ALTA)*, pages 33–41, 2012. Cited on page 43.

Claudia Maienborn. On the limits of the Davidsonian approach: The case of copula sentences. *Theoretical Linguistics*, 31(3):275–316. Walter de Gruyter, 2005. Cited on page 20.

Jean Maillard, Stephen Clark, and Edward Grefenstette. A type-driven tensor-based semantics for CCG. In *Proceedings of the EACL 2014 Workshop on Type Theory and Natural Language Semantics (TTNLS)*, pages 46–54. Association for Computational Linguistics, 2014. Cited on page 47.

Eric Margolis and Stephen Laurence. Concepts. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, summer edition, 2011. The spring 2014 edition contains minor corrections. Cited on page 29.

Michael E. McCloskey and Sam Glucksberg. Natural categories: Well defined or fuzzy sets? *Memory & Cognition*, 6(4):462–472. Springer, 1978. Cited on page 31.

Brian McMahan and Matthew Stone. A Bayesian model of grounded color semantics. *Transactions of the Association for Computational Linguistics (TACL)*, 3:103–115. 2015. Cited on pages 30 and 58.

Louise McNally. Modification. In Maria Aloni and Paul Dekker, editors, *The Cambridge Handbook of Formal Semantics*, chapter 15, pages 442–466. 2016a. Cited on page 42.

Louise McNally. Meaning at a Crossroads. [Slides]. Evening lecture at the 2016 European Summer School in Language, Logic, and Information (ESSLLI), 2016b. Cited on page 42.

Louise McNally. Kinds, descriptions of kinds, concepts, and distributions. In Kata Balogh and Wiebke Petersen, editors, *Bridging Formal and Conceptual Semantics: Selected Papers of BRIDGE-14*, chapter 3, pages 39–61. Düsseldorf University Press, 2017. Cited on page 16.

Louise McNally and Gemma Boleda. Conceptual versus referential affordance in concept composition. In *Compositionality and Concepts in Linguistics and Psychology*, number 3 in Language, Cognition, and Mind, chapter 10, pages 245–267. Springer, 2017. Cited on page 75.

Ken McRae, George S. Cree, Mark S. Seidenberg, and Chris McNorgan. Semantic feature production norms for a large set of living and nonliving things. *Behavior research methods*, 37(4):547–559. Springer, 2005. Cited on page 27.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092. American Institute of Physics, 1953. Cited on page 94.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations (ICLR), Workshop Track*, 2013. Cited on pages 17, 18, 27, 32, 93, and 111.

George A. Miller and Walter G. Charles. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28. Lawrence Erlbaum Associates, 1991. Cited on page 16.

Ruth Garrett Millikan. Language conventions made simple. *The Journal of Philosophy*, 95(4): 161–180. 1998. Reprinted in: Millikan (2005), *Language: A Biological Model*, chapter 1, pages 1–23, Oxford University Press. Cited on page 59.

Thomas P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pages 362–369, 2001. Cited on page 97.

Jeff Mitchell and Mirella Lapata. Vector-based models of semantic composition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, pages 236–244, 2008. Cited on page 37.

Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429. Wiley, 2010. Cited on pages 37 and 38.

Tom M. Mitchell, Svetlana V. Shinkareva, Andrew Carlson, Kai-Min Chang, Vicente L. Malave, Robert A. Mason, and Marcel Adam Just. Predicting human brain activity associated with the meanings of nouns. *Science*, 320(5880):1191–1195. American Association for the Advancement of Science, 2008. Cited on page 27.

Richard Montague. The proper treatment of quantification in ordinary English. In K. Jaakko J. Hintikka, Julius M. E. Moravcsik, and Patrick Suppes, editors, *Approaches to Natural Language*, number 49 in Synthese Library, chapter 10, pages 221–242. Kluwer Academic Publishers, 1973. Reprinted in: Paul Portner and Barbara H. Partee, editors (2002), *Formal semantics: The essential readings*, chapter 1, pages 17–34. Cited on pages 19 and 119.

Raymond J. Mooney. Semantic parsing: Past, present, and future. Invited talk at the ACL 2014 Workshop on Semantic Parsing, 2014. Cited on page 38.

Brian Murphy, Partha Pratim Talukdar, and Tom Mitchell. Learning effective and interpretable semantic models using non-negative sparse embedding. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING)*, pages 1933–1950. International Committee on Computational Linguistics (ICCL), 2012. Cited on page 84.

Gregory Murphy. *The big book of concepts*. Massachusetts Institute of Technology (MIT) Press, 2002. Cited on pages 28, 29, and 30.

William E. Nagy, Richard C. Anderson, and Patricia A. Herman. Learning word meanings from context during normal reading. *American Educational Research Journal*, 24(2):237–270. Sage Publications, 1987. Cited on page 16.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 6338–6347, 2017. Cited on page 36.

Raymond S Nickerson and Marilyn Jager Adams. Long-term memory for a common object. *Cognitive Psychology*, 11(3):287–307. Elsevier, 1979. Cited on page 55.

Diarmuid Ó Séaghdha. Latent variable models of selectional preference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, pages 435–444, 2010. Cited on page 108.

Diarmuid Ó Séaghdha and Anna Korhonen. Probabilistic distributional semantics with latent variable models. *Computational Linguistics*, 40(3):587–631. Massachusetts Institute of Technology (MIT) Press, 2014. Cited on page 17.

Stephan Oepen and Jan Tore Lønning. Discriminant-based MRS banking. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 1250–1255. European Language Resources Association (ELRA), 2006. Cited on pages 21 and 24.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Zdenka Uresová. Towards comparability of linguistic graph banks for semantic parsing. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, pages 3991–3995. European Language Resources Association (ELRA), 2016. Cited on page 24.

Charles K. Ogden and Ivor A. Richards. The meaning of meaning: A study of the influence of language upon thought and of the science of symbolism. Harcourt, Brace & World, Inc., 1923. Cited on page 28.

Charles Egerton Osgood. The nature and measurement of meaning. *Psychological Bulletin*, 49 (3):197–237. American Psychological Association, 1952. Cited on page 17.

David D. Palmer. Tokenisation and sentence segmentation. In Robert Dale, Hermann Moisl, and Harold Somer, editors, *Handbook of Natural Language Processing*, chapter 2, pages 11–36. Marcel Dekker, Inc., 2000. Cited on page 37.

Denis Paperno, Nghia The Pham, and Marco Baroni. A practical and linguistically-motivated approach to compositional distributional semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, pages 90–99, 2014. Cited on page 45.

Terence Parsons. Some problems concerning the logic of grammatical modifiers. *Synthese*, 21 (3-4):320–334. Springer, 1970. Reprinted in: Donald Davidson and Gilbert Harman, editors (1972), *Semantics of Natural Language*, chapter 5, pages 127–141. Cited on page 42.

Terence Parsons. *Events in the Semantics of English: A Study in Subatomic Semantics*. Current Studies in Linguistics. Massachusetts Institute of Technology (MIT) Press, 1990. Cited on page 21.

Barbara H. Partee. Montague grammar and transformational grammar. *Linguistic Inquiry*, 6 (2):203–300. Massachusetts Institute of Technology (MIT) Press, 1975. Cited on page 19.

Barbara H. Partee. Many quantifiers. In *Proceedings of the Eastern States Conference on Linguistics (ESCOL)*, pages 383–402. Ohio State University, 1988. Reprinted in: Partee (2004), *Compositionality in Formal Semantics*, pages 241–258, Blackwell. Cited on page 125.

Barbara H. Partee. The starring role of quantifiers in the history of formal semantics. In *The Logica Yearbook 2012*, pages 113–136. College Publications, 2012. Cited on page 119.

Diane Pecher, Inge Boot, and Saskia Van Dantzig. Abstract concepts: Sensory-motor grounding, metaphors, and beyond. In Brian Ross, editor, *The Psychology of Learning and Motivation*, volume 54, chapter 7, pages 217–248. Academic Press, 2011. Cited on page 26.

Charles Sanders Peirce. On a new list of categories. *Proceedings of the American Academy of Arts and Sciences*, 7:287–298. 1867. Cited on page 28.

Robin Piedeleu, Dimitri Kartsaklis, Bob Coecke, and Mehrnoosh Sadrzadeh. Open system categorical quantum semantics in natural language processing. In Lawrence S. Moss and Pawel Sobocinski, editors, *Proceedings of the 6th Conference on Algebra and Coalgebra in Computer Science (CALCO)*, volume 35 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 270–289. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2015. Cited on page 34.

Tamara Polajnar, Luana Făgărăşan, and Stephen Clark. Reducing dimensions of tensors in type-driven distributional semantics. In *Proceedings of the 19th Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1036–1046. Association for Computational Linguistics, 2014a. Cited on page 45.

Tamara Polajnar, Laura Rimell, and Stephen Clark. Evaluation of simple distributional compositional operations on longer texts. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, pages 4440–4443. European Language Resources Association (ELRA), 2014b. Cited on page 38.

Tamara Polajnar, Laura Rimell, and Stephen Clark. An exploration of discourse-based sentence spaces for compositional distributional semantics. In *Proceedings of the EMNLP Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics (LSDSem)*, pages 1–11. Association for Computational Linguistics, 2015. Cited on page 38.

Carl Pollard and Ivan A. Sag. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, 1994. Cited on page 21.

Friedemann Pulvermüller. How neurons make meaning: brain mechanisms for embodied and abstract-symbolic semantics. *Trends in Cognitive Sciences*, 17(9):458–470. Elsevier, 2013. Cited on page 26.

Behrang QasemiZadeh and Laura Kallmeyer. Random positive-only projections: PPMI-enabled incremental semantic space construction. In *Proceedings of the 5th Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 189–198. Association for Computational Linguistics, 2016. Cited on page 108.

Willard Van Orman Quine. *Word and Object*. Massachusetts Institute of Technology (MIT) Press, 1960. Cited on page 42.

Reinhard Rapp. A practical solution to the problem of automatic word sense induction. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL), Interactive Poster and Demonstration Sessions*, pages 26–29, 2004. Cited on page 34.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. Massachusetts Institute of Technology (MIT) Press, 2006. Cited on page 57.

François Recanati. Compositionality, flexibility, and context-dependence. In Wolfram Hinzen, Edouard Machery, and Markus Werning, editors, *Oxford Handbook of Compositionality*, chapter 8, pages 175–191. Oxford University Press, 2012. Cited on pages 42 and 73.

Siva Reddy, Diana McCarthy, and Suresh Manandhar. An empirical study on compositionality in compound nouns. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 210–218. Asian Federation of Natural Language Processing (AFNLP), 2011. Cited on page 136.

Radim Řehůřek and Petr Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 46–50. European Language Resources Association (ELRA), 2010. Cited on page 111.

Marek Rei. *Minimally supervised dependency-based methods for natural language processing*. PhD thesis, University of Cambridge, 2013. Cited on page 35.

Marek Rei and Ted Briscoe. Looking for hyponyms in vector space. In *Proceedings of the 18th Conference on Computational Natural Language Learning (CoNLL)*, pages 68–77. Association for Computational Linguistics, 2014. Cited on page 35.

Marek Rei, Daniela Gerz, and Ivan Vulić. Scoring lexical entailment with a supervised directional similarity network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL), Short Papers*, pages 638–643, 2018. Cited on page 36.

Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62(1): 107–136. Springer, 2006. Cited on page 41.

Laura Rimell. Distributional lexical entailment by topic coherence. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Long Papers*, pages 511–519, 2014. Cited on page 35.

Laura Rimell, Jean Maillard, Tamara Polajnar, and Stephen Clark. RELPRON: A relative clause evaluation dataset for compositional distributional semantics. *Computational Linguistics*, 42

(4):661–701. Massachusetts Institute of Technology (MIT) Press, 2016. Cited on pages 38, 115, and 116.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. Reasoning about entailment with neural attention. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 2016. Cited on page 40.

Eleanor Rosch. Cognitive representations of semantic categories. *Journal of experimental psychology: General*, 104(3):192. American Psychological Association, 1975. Cited on pages 29 and 86.

Eleanor Rosch. Principles of categorization. In Eleanor Rosch and Barbara Bloom Lloyd, editors, *Cognition and categorization*, chapter 2, pages 27–48. Lawrence Erlbaum Associates, 1978. Reprinted in: Eric Margolis and Stephen Laurence, editors (1999), *Concepts: Core Readings*, chapter 8, pages 189–206. Cited on pages 29 and 86.

Charles Ruhl. *On monosemy: A study in linguistic semantics*. State University of New York (SUNY) Press, 1989. Cited on pages 33, 66, and 73.

Mehrnoosh Sadrzadeh, Stephen Clark, and Bob Coecke. The Frobenius anatomy of word meanings I: subject and object relative pronouns. *Journal of Logic and Computation*, 23(6):1293–1317. Oxford University Press, 2013. Cited on page 45.

Mehrnoosh Sadrzadeh, Dimitri Kartsaklis, and Esma Balkır. Sentence entailment in compositional distributional semantics. *Annals of Mathematics and Artificial Intelligence*, 82(4): 189–218. Springer, 2018. Cited on page 36.

Ivan Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. Multiword expressions: A pain in the neck for NLP. In *Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*, pages 1–15. Springer, 2002. Cited on page 136.

Magnus Sahlgren. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. PhD thesis, Stockholm University, 2006. Cited on page 17.

Gerard Salton. Mathematics and information retrieval. *Journal of Documentation*, 35(1):1–29. 1979. Cited on page 18.

Barbara A. C. Saunders and Jaap Van Brakel. Are there nontrivial constraints on colour categorization? *Behavioral and Brain Sciences*, 20(2):167–179. Cambridge University Press, 1997. Cited on page 31.

David Schlangen, Sina Zarrieß, and Casey Kennington. Resolving references to objects in photographs using the words-as-classifiers model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, pages 1213–1223, 2016. Cited on page 30.

Hinrich Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1): 97–123. Massachusetts Institute of Technology (MIT) Press, 1998. Cited on page 34.

John R. Searle. The background of meaning. In John R. Searle, Ferenc Kiefer, and Manfred Bierwisch, editors, *Speech Act Theory and Pragmatics*, chapter 10, pages 221–232. D. Reidel Publishing Company, 1980. Cited on pages 42, 70, 71, and 72.

Carina Silberer and Mirella Lapata. Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, pages 721–732, 2014. Cited on page 27.

Edward E. Smith and Douglas L. Medin. *Categories and Concepts*. Harvard University Press, 1981. Cited on page 28.

Noah Smith. Squashing computational linguistics. Invited talk at the 55th Annual Meeting of the Association for Computational Linguistics (ACL), 2017. Cited on page 15.

Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In David E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume I: Foundations*, chapter 6, pages 194–281. Massachusetts Institute of Technology (MIT) Press, 1986. Cited on page 85.

Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the NIPS 2010 Deep Learning and Unsupervised Feature Learning Workshop*, 2010. Cited on page 38.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing (EMNLP) and Computational Natural Language Learning (CoNLL)*, pages 1201–1211. Association for Computational Linguistics, 2012. Cited on page 38.

Lars Jørgen Solberg. *A Corpus Builder for Wikipedia*. Master's thesis, University of Oslo, 2012. Cited on pages 21 and 103.

Sanghoun Song. *Modeling information structure in a cross-linguistic perspective*. Topics at the Grammar-Discourse Interface. Language Science Press, 2017. Cited on page 24.

Karen Spärck-Jones. *Synonymy and Semantic Classification*. PhD thesis, University of Cambridge, 1964. Reprinted in 1986 by Edinburgh University Press. Cited on pages 16 and 33.

Karen Spärck-Jones. Statistics and retrieval: past and future. In *Proceedings of the International Conference in Computing: Theory and Applications (Platinum Jubilee Conference of the Indian Statistical Institute)*, pages 396–405. Institute of Electrical and Electronics Engineers, 2007a. Cited on page 18.

Karen Spärck-Jones. Computational linguistics: what about the linguistics? *Computational Linguistics*, 33(3):437–441. Massachusetts Institute of Technology (MIT) Press, 2007b. Cited on page 15.

Mark Steedman and Jason Baldridge. Combinatory Categorial Grammar. In Robert D. Borsley and Kersti Börjars, editors, *Non-Transformational Syntax: Formal and Explicit Models of Grammar*, chapter 5, pages 181–224. Blackwell, 2011. Cited on page 47.

Luc Steels. The symbol grounding problem has been solved. So what's next? In Manuel de Vega, Arthur Glenberg, and Arthur Graesser, editors, *Symbols and embodiment: Debates on meaning and cognition*, chapter 12, pages 223–244. Oxford University Press, 2008. Cited on page 26.

Isidora Stojanovic. Situation semantics. In Albert Newen and Raphael van Riel, editors, *Identity, Language, and Mind: An Introduction to the Philosophy of John Perry*, chapter 5. Center for the Study of Language and Information (CSLI) Publications, 2012. Cited on page 22.

Peter R. Sutton. *Vagueness, Communication, and Semantic Information*. PhD thesis, King's College London, 2013. Cited on pages 31 and 59.

Peter R. Sutton. Towards a probabilistic semantics for vague adjectives. In Henk Zeevat and Hans-Christian Schmitz, editors, *Bayesian Natural Language Semantics and Pragmatics*, chapter 10, pages 221–246. Springer, 2015. Cited on page 32.

Peter R. Sutton. Probabilistic approaches to vagueness and semantic competency. *Erkenntnis*. Springer, 2017. Cited on pages 32 and 60.

Kevin Swersky, Ilya Sutskever, Daniel Tarlow, Richard S. Zemel, Ruslan R. Salakhutdinov, and Ryan P. Adams. Cardinality Restricted Boltzmann Machines. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 3293–3301, 2012. Cited on pages 85 and 93.

Mariarosaria Taddeo and Luciano Floridi. Solving the symbol grounding problem: a critical review of fifteen years of research. *Journal of Experimental & Theoretical Artificial Intelligence*, 17(4):419–445. Taylor & Francis, 2005. Cited on page 26.

Mariarosaria Taddeo and Luciano Floridi. A praxical solution of the symbol grounding problem. *Minds and Machines*, 17(4):369–389. Springer, 2007. Cited on page 26.

Alfred Tarski and Robert L. Vaught. Arithmetical extensions of relational systems. *Compositio Mathematica*, 13:81–102. 1956. Cited on page 19.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581. 2006. Cited on page 44.

Michael Henry Tessler and Noah D. Goodman. A pragmatic theory of generic language. Unpublished draft, 2016. Cited on page 126.

Stefan Thater, Hagen Fürstenau, and Manfred Pinkal. Word meaning in context: A simple and effective vector model. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1134–1143. Asian Federation of Natural Language Processing (AFNLP), 2011. Cited on page 43.

Tijmen Tieleman. Training Restricted Boltzmann Machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 1064–1071. Association for Computing Machinery, 2008. Cited on page 95.

Tijmen Tieleman. RMSProp. Unpublished work cited by Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky, in Lecture 6e of *Neural Networks for Machine Learning*, Coursera, 2012. Cited on page 107.

Kristina Toutanova, Christopher D. Manning, Dan Flickinger, and Stephan Oepen. Stochastic HPSG parse selection using the Redwoods corpus. *Research on Language and Computation*, 3(1):83–105. Springer, 2005. Cited on page 103.

Peter D. Turney. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning (ECML)*, pages 491–502, 2001. Cited on page 17.

Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188. 2010. Cited on page 17.

Johan Van Benthem. Questions about quantifiers. *The Journal of Symbolic Logic*, 49(2):443–466. Association for Symbolic Logic, 1984. Cited on page 119.

Kees Van Deemter. *Not exactly: In praise of vagueness*. Oxford University Press, 2010. Cited on page 31.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. Order-embeddings of images and language. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 2016. Cited on page 35.

Luke Vilnis and Andrew McCallum. Word representations via Gaussian embedding. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015. Cited on pages 32, 34, 36, and 45.

Veronika Vincze. *Semi-compositional noun + verb constructions: Theoretical questions and computational linguistic analyses*. PhD thesis, University of Szeged, 2012. Cited on page 136.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3156–3164. Institute of Electrical and Electronics Engineers, 2015. Cited on page 44.

Ivan Vulić, Daniela Gerz, Douwe Kiela, Felix Hill, and Anna Korhonen. HyperLex: A large-scale evaluation of graded lexical entailment. *Computational Linguistics*, 43(4):781–835. Massachusetts Institute of Technology (MIT) Press, 2017. Cited on page 36.

Shuohang Wang and Jing Jiang. Learning natural language inference with LSTM. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 1442–1451, 2016. Cited on page 40.

Su Wang, Stephen Roller, and Katrin Erk. Distributional modeling on a diet: One-shot word learning from text only. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP), Long Papers*, pages 204–213. Asian Federation of Natural Language Processing (AFNLP), 2017. Cited on page 45.

Julie Weeds, David Weir, and Diana McCarthy. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 1015–1021. International Committee on Computational Linguistics (ICCL), 2004. Cited on page 35.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pages 2249–2259. International Committee on Computational Linguistics (ICCL), 2014. Cited on page 36.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), Long Papers, pages 1112–1122, 2018. Cited on page 40.

Ludwig Wittgenstein. *Philosophical Investigations*. Blackwell, 1953. Translated by Gertrude Elizabeth Margaret Anscombe. The original German text was published in 1958 under the title *Philosophische Untersuchungen*. Cited on page 29.

Kimberly Wong, Frempongma Wadee, Gali Ellenblum, and Michael McCloskey. The devil's in the g-tails: Deficient letter-shape knowledge and awareness despite massive visual experience. *Journal of Experimental Psychology: Human Perception and Performance*. American Psychological Association, 2018. Cited on pages 55 and 56.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, 2015. Cited on page 44.

Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding Belief Propagation and its generalizations. In Gerhard Lakemeyer and Bernhard Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*, chapter 8, pages 239–269. Morgan Kaufmann Publishers, 2003. Cited on page 93.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. ABCNN: Attention-Based Convolutional Neural Network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics (TACL)*, 4:259–272. 2016. Cited on page 40.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics (TACL)*, 2:67–78. 2014. Cited on page 44.

Gisle Ytrestøl, Dan Flickinger, and Stephan Oepen. Extracting and annotating Wikipedia subdomains: Towards a new eScience community resource. In *Proceedings of the 7th International Workshop on Treebanks and Linguistic Theories*, pages 185–197, 2009. Cited on page 103.

Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353. Academic Press, 1965. Cited on pages 32 and 65.

Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning—I. *Information Sciences*, 8(3):199–249. Elsevier, 1975. Cited on page 32.

Sina Zarrieß and David Schlangen. Is this a child, a girl or a car? Exploring the contribution of distributional similarity to learning referential word meanings. In *Proceedings of the 15th Annual Conference of the European Chapter of the Association for Computational Linguistics (EACL), Short Papers*, pages 86–91, 2017a. Cited on page 30.

Sina Zarrieß and David Schlangen. Obtaining referential word meanings from visual and distributional information: Experiments on object naming. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), Long Papers*, pages 243–254, 2017b. Cited on page 30.

Matthew D. Zeiler. AdaDelta: An adaptive learning rate method. Unpublished draft, 2012. Cited on page 107.

Thomas R. Zentall, Mark Galizio, and Thomas S. Critchfield. Categorization, concept learning, and behavior analysis: An introduction. *Journal of the Experimental Analysis of Behavior*, 78(3):237–248. Wiley, 2002. Cited on page 30.