Noname manuscript No.
(will be inserted by the editor)

Ideal words

A vector-based formalisation of semantic competence

Aurélie Herbelot · Ann Copestake

Received: date / Accepted: date

Abstract In this theoretical paper, we consider the notion of semantic competence and its relation to general language understanding – one of the most sough-after goals of Artificial Intelligence. We come back to three main accounts of competence involving a) lexical knowledge; b) truth-theoretic reference; and c) causal chains in language use. We argue that all three are needed to reach a notion of meaning in artificial agents and suggest that they can be combined in a single formalisation, where competence develops from exposure to observable performance data. We introduce a theoretical framework which translates set theory into vector-space semantics by applying distributional techniques to a corpus of utterances associated with truth values. The resulting meaning space naturally satisfies the requirements of a causal theory of competence, but it can also be regarded as some 'ideal' model of the world, allowing for extensions and standard lexical relations to be retrieved.

Keywords formal semantics · distributional semantics · competence

1 Introduction

From a high-level perspective, research in Natural Language Processing (NLP) can be said to be dedicated to the question 'Can we give machines the faculty of language?' Seen from a theoretical linguistics point of view, this question boils down to solving the problem of *competence* acquisition. However, the notion of competence itself has received relatively little attention in recent NLP and AI frameworks, where focus has been on acquiring specific linguistic skills from a linear signal consisting essentially of surface forms. As pointed out by various researchers, the practice of applying statistical techniques to enormous amounts of text is unlikely to yield human-like language, including its relation to the world around us, its pragmatic nuances, or the fact that it can be acquired from very limited data (Bender and Koller, 2020; Trott et al., 2020; Linzen, 2020). The present paper seeks to provide a more encompassing computational framework by coming back to the main theories of competence in the linguistic literature, focusing specifically on the acquisition of *meaning*.

The fundamental distinction between competence (knowing one's language) and performance (using one's language) is introduced in the opening of *Aspects of the theory of syntax* (henceforth *Aspects*, Chomsky, 1965). The distinction is meant to capture the fact that native speakers of a language seem to be able to reliably make grammaticality judgements, even though their observable utterances exhibit errors as well as various types of limitations on their form, length and complexity. In a word, people know the rules of their language but don't always apply them in practice. Performance is degraded competence.

Whilst the notion of competence is attractive for the study of syntax and grammaticality judgements, its semantic equivalent has proved extremely difficult to pinpoint in linguistic theory. At first glance, it would seem that semantic competence should be the ability to recognise utterances that are true of a given world (Cresswell, 1978; Partee, 1979; Asher, 1988; Soames, 1989). Or perhaps, it should simply be about satisfying some notion of lexical selectional restriction (Chomsky, 1965; Katz and Fodor, 1963). But it has been noted that the boundary between felicity and infelicity,

Aurélie Herbelot
Center for Mind/Brain Sciences,
Department of Information Engineering and Computer Science
University of Trento
E-mail: aurelie.herbelot@unitn.it

Ann Copestake
Department of Computer Science and Technology
University of Cambridge
E-mail: ann.copestake@cl.cam.ac.uk

particularly with regard to truth conditions, is very hard to elicit (Matthewson, 2004). This in itself might only be a matter of gradation (syntactic judgements are not always perfect either). But the more fundamental issue at hand is that the various semantic theories of competence, whether related to truth values, to the lexicon or to anything else, have different philosophical underpinnings. Reconciling them remains an extremely challenging task (Partee, 2014).

Beyond philosophical considerations, we must further take into account Chomsky's epistemological reflections on the study of linguistics. His argument in *Aspects* is that the status of linguistics as a science depends on having competence as its object of study, that is, on the investigation of the *mental* phenomenon that supports observable performance. In short, the job of linguistics is not only to describe the formal structure of competence, as theoreticians would have it, but also to explain the cognitive processes that might lead to its acquisition from performance data. Following this ideal, we focus in this paper on the goal of finding a formal representation which would be amenable to defining various types of semantic competence (thus accounting for theoretical matters), and which could be shown to be acquirable from performance data (thus accounting for cognitive reality and, of importance to us, allowing for the *computational* simulation of specific aspects of linguistic cognition).

Theoretically, we draw the consequence of performance being defined as an incomplete or degraded competence, namely that performance and competence are made 'of the same stuff'. If performance, the observable part of language, can be characterised in terms of utterances, so should competence. Formally, we define both competence and performance as generating a set of sentences uttered about some world(s) using some grammar. We further acknowledge the various incarnations of semantic competence and hypothesise that our representations should allow for at least three levels of meaning to be extracted: the truth-theoretic level, the lexical level, and the level of language use.

Cognitively, we posit that our representation of performance sentences should allow a learner to infer from it the building blocks of competence, at all relevant levels of meaning. To model learnability, we use *distributional semantics* (DS: Erk, 2012; Lenci, 2008; Boleda, 2020), a vector-based representation of sentence constituents. DS defines meaning through usage and generates representations through the computational analysis of large corpora. That is, it relies on observable data – the data of performance –, as recorded from the many individual speakers who produced the utterances included in a given corpus.

Combining the theoretical aspects of competence with DS presupposes a representation which accommodates model theory as well as distributional learning. The contribution of this work is therefore the formal re-definition of a truth-theoretic model in terms of a dynamic vector space, with dimensions consisting of the individuals (both singular and plural) in a given universe. Predicates are defined with respect to those dimensions, resulting in a framework where meanings are a function of the entities that instantiate them. A minimal example of such a model is shown in Fig. 1, showing two single instances of trees and their corresponding plurality as a 3D space, and some predicates living in that space as boolean vectors, within a cube. This space has a number of properties desirable in both formal and distributional semantics, which we will describe in the course of the paper: ability to compute pluralities and differentiate collectives from distributive predicates, compositionality, amenability to probabilistic approaches and word meaning contextualisation.

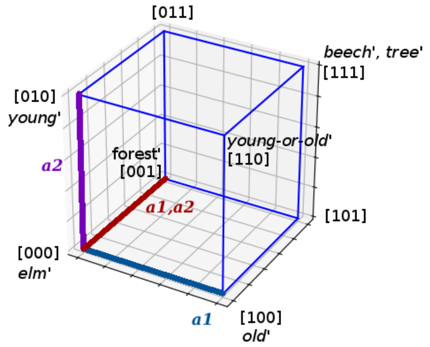
2 Competence and performance

We will first position our paper with respect to previous approaches to the competence / performance distinction. In what follows, we introduce various frameworks, starting with the canonical Chomskian definition of competence, and subsequently highlighting specific attempts to port the original notion to semantics. We discuss proposals with different foci and look at semantic competence from the point of view of a) lexical semantics (Katz and Fodor, 1963); b) 'ideal' truth theory (Partee, 1979); and c) a causal theory of reference (Kripke, 1972), which contends that people simply use words as others have used them before. Our aim is to position ourselves at the junction of those proposals, hoping that our formalisation provides a bridge across them.

2.1 Competence and performance in syntax

Chomsky (1957, 1965) claims that syntactic competence corresponds to some unconscious knowledge of a speaker-hearer, which reflects the grammar of his or her language. Competence is 'error-free' and not constrained by speaker limitations like working memory size or processing time. Performance, in contrast, refers to the observable side of language, including associated production errors, memory limitations, etc. Linguistics, under that view, is the study of competence, that is, of what it means to know one's language, and of the processes that leads to its acquisition. As such, linguistics can be regarded as a branch of psychology.

According to Chomsky, the acquisition of competence from performance data implies the existence of an underlying Universal Grammar (UG), i.e. an innate system shared by all human beings, which kick-starts the process of learning one's native language. The existence of UG is justified by several observations. First, all human languages seem to share some properties. Second, children learn their language extremely rapidly, despite being exposed to relatively sparse



	$\{a1\}$	$\{a2\}$	$\{a1, a2\}$
beech’	1	1	1
tree’	1	1	1
old’	1	0	0
young’	0	1	0
elm’	0	0	0
forest’	0	0	1

Fig. 1: A model with two entities and their plurality, in a space with basis $B_M = \{\{a1\}, \{a2\}, \{a1, a2\}\}$, corresponding to some universe U. Predicates $P_L = \{beech, tree, old, young, elm, forest\}$ are boolean vectors, thus defining the vertices of a cube: old$'$, the set of old things, is given by the vector $[100]$ (the bottom right vertex of the cube), corresponding to the set $\{a1\}$. We will show in the paper how to derive composed predicates such as young-or-old$'$, and how to relate the entities to their plurals.

data (‘poverty of the stimulus’), and within a language community, they seem to converge towards the same language even though they are exposed to different utterances. Furthermore, they acquire a notion of grammaticality even in the absence of explicit information about ungrammaticality. Finally, there seems to be some ‘ordering’ in the way that various constructions are acquired. The question of innateness is an interesting one for AI practitioners, as it encourages the field to question whether purely data-driven approaches can account for human-like acquisition, and what kind of inbuilt knowledge comes with a specific machine learning architecture. But the notion of Universal Grammar is not straightforwardly applicable to semantics, prompting the question of defining competence with regard to meaning.

Partee (2015) gives a thorough account of the relation between Chomskian theory and semantics, highlighting how the syntax-semantics interface figures prominently in all of Chomsky’s writing – and this, despite his reservations about the importance of semantics. *Aspects* (Chomsky, 1965) introduces the notion of *deep structure* as the input to semantics. The specific proposal in that book is that syntax is what generates such deep structure, and that deep structure forms the basis of semantic interpretation. The semantic component assumed by Chomsky was first developed in an account by Katz and colleagues (Katz and Fodor, 1963; Katz and Postal, 1964), which we introduce in the next section.

2.2 Competence and performance in semantics

Competence as lexical semantics Following the path of ‘psychological’ linguistics, Katz and Fodor (1963) pick up on the notion of generative grammar advocated by Chomsky, and argue that the ability to determine the meaning of a novel sentence cannot be given by syntax alone: two sentences with identical syntactic structures can mean different things, while two sentences with different syntactic structures can mean the same thing. They propose that the object of semantics should be what is left when “subtracting grammar from the goals of a description of a language” (Katz and Fodor, 1963: p172). In other words, semantics should model whatever in language is left unexplained by a theory of grammar. In that paper, the ‘leftovers’ of grammar can all be seen as elements of lexical semantics: e.g. the relations of hyponymy or antonymy, as well as word senses. Katz and Fodor argue that having knowledge of such relations lets the speaker detect non-syntactic ambiguities (e.g. the meaning of *bill* in *the bill is large*), resolve them (in *the bill is large but need not be paid*, only one sense of *bill* applies), and also identify semantic anomalies (**the paint is silent*). A competent speaker, thus, should be able to distinguish those meanings and relations between them. Following on this work, Katz and Postal (1964) propose a compositional account of such components, stating that transformations in a generative grammar will be meaning-preserving. Notably, Katz and Fodor do not make any assumption with regard to the innateness of semantics, although later work by Fodor will famously argue for the innateness of concepts (Fodor, 1975).

Competence as ‘ideal’ truth theory Moving to the relationship between cognitive approaches to linguistics and formal semantics, Putnam (1975) argues that it is possible to not know the intension of a term and still have some lexical knowledge about it: whilst not being able to tell a beech from an elm, he is aware that the two kinds are different from each other and that they are some types of trees. Further, he seems to be able to use the terms appropriately (‘competently’). So semantic competence, he claims, may be observable at the level of language use without the speaker mastering truth-theoretic values. Despite appearances, people don't seem to *know* their language, at least extensionally.

Following such claims, Partee (1979) remarks that it is indeed difficult to find a notion of semantic competence which is compatible with both formal semantics and psychological, Chomskyan linguistics. She questions what it might mean to have full competence in a truth-theoretic, Montague semantics, and explores the notion of a perfect, 'godly' speaker, who would have perfect ability to match words to extensions (i.e. a perfect interpretation function), and would be logically omniscient. Such a speaker, she proposes, might embody (intensional) semantic competence. The model incompleteness and the erroneous beliefs observed in actual speakers could just be put down to performance factors. She however rejects this proposal in view of issues related to propositional attitudes: even if P and Q are logically equivalent, the godly speaker will not make the inference from *Irena believes that* P to *Irena believes that* Q. That is, the godly speaker cannot follow the premises of Montague semantics that a) logically equivalent constituents are substitutable; and b) intensions of sentences ('knowing what the sentence means') is a function from possible worlds to truth values. This kind of truth-theoretic super-competence only works if all other speakers are similarly godly.[1]

Competence as causal theory Another issue highlighted by Partee (1979) is that of rigid designators such as proper names. In formal semantics, proper names are taken to have the same extension in all possible worlds. But this view is not compatible with a psychological theory of meaning because across speakers, we will observe differences in representations of such terms: a speaker may not know who Frege is, or have misunderstood who Frege is, and still be able to use the word appropriately, for instance when they ask *Who is Frege?* Kripke (1972) argues for a 'causal theory of reference' to explain such effects: in a nutshell, people use the word *Frege* in a way that is consistent with what they have observed in other speakers' utterances. In this case, as in the beech / elm example, competent usage follows from simple exposure to performance data, without assuming fully competent extensional knowledge. Again, such a view does not seem to account for a view of semantic competence in the formal truth-theoretic tradition.

A compromise view Partee (2014) makes the interesting point that a formal view of competence as fully *knowing* one's language may have been mistaken. She draws on the following claim from Burge (2010): "[...] both perceptual reference and the specific ways individuals perceive the world (their perceptual groupings and categorizations) depend more on the ways individuals are physically and functionally related to specific types of entities in the environment than on individuals' ability to describe or know something about what they perceive" (Burge 2010:xvi-xvii). Mirroring this view of perception, Partee claims that semantic competence does not have to be godly *super*-competence. It is acceptable to assume that there is a relation between constituents of our language and external reality *and at the same time* that language users are sometimes mistaken or don't possess competence in all aspects of meaning. In other words, truth-theoretic semantics may be able to live with imperfect truth. This is the position we will adopt in this paper.

2.3 How to position this paper

The present paper makes an attempt at piecing together the various arguments and ideas about competence acquisition that we have related above. Our position is that linguistic competence *is* the result of cognitive processes but that it does not preclude the formal definition of an intensional semantics over incomplete models, dependent on a speaker's exposure to performance data. That is, following Partee (2014), *competence is not super-competence*. We will explore what this means in terms of the formalisation of a model.

Our hypothesis, as stated in §1, is that the acquisition of semantic (and syntactic) competence should be derivable from performance data. The formalisation of competence should have the same components as that of performance, so that performance can be seen as 'incomplete' or 'degraded' competence rather than a fully different type of linguistic object. We have seen that semantic competence can refer to various notions. One relates to the knowledge of core lexical relations (Katz and Fodor, 1963), another to the ability to retrieve the extension of a term (Partee, 1979), yet another to the 'acceptable' use of a term (Katz and Fodor, 1963; Putnam, 1975; Kripke, 1972). We endeavour in this paper to find a common formalisation underlying these three notions, whilst at the same time acknowledging that they may not emerge jointly (and consequently not fail jointly). The limitations of speakers' competence that we presented in §2.2 (e.g. not knowing the *extensional* difference between elms and beeches) should be explicable in terms of the very nature of the performance data they were exposed to. A consequence of our approach is that Katz and Fodor's lexical relations should be discoverable from performance data rather than assumed to be innate, and they should be tightly bound to the state of the syntax-semantics interface in the learner. We will cover this in §4.3.

The currently most popular approach to learning meaning from performance data is *distributional semantics* (henceforth DS – for introductions to the topic, see Erk, 2012; Lenci, 2008; Boleda, 2020). DS is a corpus-driven technique to acquire lexical meaning, in the tradition of distributionalists such as Harris (1954). By virtue of being corpus-driven,

[1] At which point, we note, there is not much point in speakers talking to each other, since they all have perfect ontological knowledge.

DS is usually considered a representation of performance, to be distinguished from the type of lexical relations that might be extractable from truth-theoretic approaches. We believe that vector-based semantics is the right tool to accommodate the requirements we set for a full account of competence. But in its current form, it is completely unsuitable for representing essential ingredients of a formal semantics – crucially, by failing to encode extensions. A large part of the present paper is thus dedicated to building a kind of vector semantics which will be amenable to both set-theoretical work and the type of lexical knowledge that DS excels at.

3 Preliminaries

In this section, we present the formalisms that we will be using throughout the paper. We include a short overview of Distributional Semantics (DS), a brief presentation of Minimal Recursion Semantics (our chosen sentence representation), and some pointers to Linkian semantics, which we use to represent plurality.

3.1 Distributional Semantics

Distributional semantics models build a representation of each term in a vocabulary by adding up the number of times a context occurs with it, thereby producing a co-occurrence frequency matrix which is usually re-weighted using information measures such as Pointwise Mutual Information. The resulting representations can be viewed as vectors in a multidimensional space.

DS has proved to be powerful in modelling psycholinguistic phenomena at the word level, including similarity and priming (Lund et al., 1995; Landauer and Dumais, 1997). Interestingly, Dumais & Landauer's *Solution to Plato's problem* (1997) proposes an answer to the 'poverty of the stimulus' problem, involving one of the first highly popularised distributional semantics model, further incarnated in Latent Semantic Analysis (LSA). There is thus a historical connection between DS techniques and some of the linguistic phenomena usually seen as part of competence acquisition.

Beyond its success at the single word level, DS has made small progress on the matter of compositionality. Clark (2012) and Erk (2012) give extensive introductions to composition in count-based models. More recent developments have focused on training neural systems to represent sentences directly (ELMo, Peters et al., 2018; BERT, Devlin et al., 2019), and as a by-product, contextualised word representations, following previous insights from count-based models (Erk and Padó, 2008; Thater et al., 2011). With respect to lexical relations, DS has had some success with e.g. hyponymy (Baroni and Lenci, 2010; Lenci and Benotto, 2012; Roller et al., 2014). However, it is fair to say that it still struggles when encoding relations that require both lexical and denotational knowledge, such as antonymy.

The problems experienced by DS models at the level of lexical relations are symptoms of a more fundamental issue, namely that such models are not designed to cater for referential information. For similar reasons, the framework has failed so far to account for logical phenomena that formal semantics naturally models, such as quantification. It has essentially focused on modelling generic, conceptual information. It is unclear how DS should be transformed to represent the specific attributes of individual entities and sets of entities. In response to such issues, a new sub-area has developed, referred to as 'Formal Distributional Semantics (FDS)' (for an introduction, see Boleda and Herbelot, 2017). Although still in its infancy, this area of work is promising, both at a theoretical and experimental level. We leave a brief review of relevant FDS proposals to the end of this paper (§7), where we show their relation to our framework.

In what follows, we will adopt the formal definition of a distributional model given by Erk (2016). A distributional model D is a structure of the form

$$< T_D, O_D, B_D, C_D, X_D, A_D, S_D >$$

T_D and O_D are respectively the set of target words and the set of context items under consideration. B_D are the dimensions (the basis) the vector space. C_D is the input corpus, which can be considered a collection of target and context items: $C_D \in (O_D \cup T_D)^*$ (any word not in the target or context set is then ignored). X_D is an extraction function which takes the corpus and produces a frequency space: $X_D : (O_D \cup T_D)^* \rightarrow ((T_D \times O_D) \rightarrow \mathbb{N}_0)$. Any post-processing such as weighting or dimensionality reduction is bundled into an aggregation function $A_D : ((T_D \times O_D) \rightarrow \mathbb{N}_0) \rightarrow ((T_D \times B_D) \rightarrow \mathbb{R})$. Finally, the similarity function over terms in the space is defined as $S_D : (T_D \times T_D) \rightarrow \mathbb{R}$.

3.2 Grammar and logic

We assume an underlying **grammar** G, which could in principle use any formalism. Whenever we talk of the compositional rules in G, we will use context-free notation such as $VP \rightarrow V\ NP$, but this is only for convenience. The terminals T_G in the grammar correspond to predicates P_L and logical operators L_L in a **logic** L, which has the structure $L =< P_L, L_L, V_L >$. V_L is a set of variables. In order to match the underspecified logical representation we are about to introduce, we assume a constant-free logic. But there is no principled reason why constants cannot be expressed in our overall framework.

As with the grammar, any type of logic could in principle be plugged into the framework we are to propose. Because of this, we will build our formalisation around Minimal Recursion Semantics (MRS: Copestake et al. 2005), a meta-language which lets us encode an underspecified representation of logical forms. MRS has been shown to be compatible with HPSG grammars such as the English Resource Grammar (ERG: Flickinger 2000) and context-free grammars (Copestake, 2007).

In more detail, MRS is based on the principle that the compositional semantic representation should capture the information available from syntax but it does not make distinctions that syntax cannot resolve. Thus MRS representations are underspecified for certain ambiguities which are not resolved by syntax, such as scope ambiguity. An MRS structure consists of **elementary predications** (EP) consisting of a predicate and its arguments, identified by variables. EPs are implicitly conjoined by a $\wedge$ connective: e.g., the representation for *young tree* is $young(x4), tree(x4)$ rather than $young(x4) \wedge tree(x4)$. There are no specific quantifier or disjunction operators. Those are handled by dedicated elementary predications, as is the rest of the lexicon. For instance, *an elm is not old* would make use of a negation operator neg, together with a scoping mechanism:

$$l1 : elm(x1), l1 : old(x1), h1 : neg(\_2, x1), h1\ qeq\ l1$$

where $l1$ and $h1$ label the predicates and negation operators respectively, and the qeq relation indicates the scope of neg. Scope can be left underspecified: an MRS structure with underspecified scope can be related to a set of scope-resolved MRSs, interpreted as a disjunction. The mechanism avoids the need for explicit nesting in the MRS structure (the syntax is 'flat').

Formally, a bare-bone MRS without scoping mechanism is a logic L where P_L is a set of predicates corresponding to the elements in T_G (the terminals in the grammar), and L_L is the single $\wedge$ connective represented by a comma. In this paper, we will consider a logic where predicates have one argument only, that is, the logical form LF of a sentence will be a string of EPs so that each $EP \in (P_L \times V_L)^*$. (We express n-place predicates as unary predicates with a single, potentially ordered tuple argument: see footnote 4 in §4.)

MRS representations can be obtained for sentences from an automatic parser, and in that form, are independent of a model of the world (as opposed to traditional representations in Montague semantics). We will however need to link them to extensional representations in the course of this work. We thus introduce our notion of model in the next section.

3.3 Model

We define a model M in the standard way, as a structure $< U, ||.|| >$. U is the universe containing a non-empty set of objects. $||.||$ is an interpretation function which maps an n-place predicate to a set of ordered n-tuples of objects in U, and a proposition to a truth value. For instance, assuming that elm is a predicate in the grammar with a one-place tuple argument, we might have $||elm|| = \{\{a_1\}, \{a_2\}\}$, meaning that the predicate elm maps onto the singletons $\{a_1\}$ and $\{a_2\}$ in the universe. $||elm||$ is the *extension* of elm. We will also use the prime notation whenever convenient, so $\text{elm}' = ||elm|| = \{\{a_1\}, \{a_2\}\}$. Note that we do not disambiguate extensions: if a can truthfully be called a *tree*, then $\text{tree}'(a)$ is true, whether the tree is a living being or a graph. We will discuss later how several conceptual categories can nevertheless emerge from such ambiguities (§5.3).

Set representation: For our set representation, we adopt a Linkian semantics (Link, 1983), where sets are described as join-semilattices. This allows us to talk about plurality and collectivity, two aspects of formal semantics that are missing in current machine learning approaches to the modelling of language but are nevertheless essential in making correct inferences from utterances (see e.g. the distinction between *The children ate cake* $\rightarrow$ *A child ate cake* vs. *The children built a raft* $\nrightarrow$ *A child built a raft*).

A lattice is a partially ordered set in which any two elements have a unique least upper bound (their *join*) and a unique greatest lower bound (their *meet*). The lattices described by Link are join-semilattices, i.e. only the join constraint is enforced. An example of a join-semilattice is shown in Fig.2, for some set of trees $\{a_1, a_2, a_3\}$ in a mini-world. Note, for future reference, that subsets of that join-semilattice correspond to sets of single and plural individuals which can form the basis of an entity space, of the type shown in Fig. 1. We reproduce the cube from Fig. 1, with its three individuals $\{\{a_1\}, \{a_2\}, \{a_1, a_2\}\}$, to make this clear.

In Linkian plural semantics, the * (star) sign generates all individuals sums of members of the extension of some predicate P. So with $P = tree$, the extension of *tree* is a join-semilattice $^*tree$ representing all possible sums of trees in our domain (as shown in the picture). The sign σ is the sum operator. $\sigma a P a$ represents the sum, or supremum, of all objects that are P (so the top of the semilattice). In the example above, $\sigma a\ tree\ a$ is the supremum of all trees: $\{a_1, a_2, a_3\}$. Any individual plural can be retrieved via the individual sum operator $\oplus$. So $\{a_1\} \oplus \{a_2\}$ is the plural object consisting of a_1 and a_2, that is, $\{a_1, a_2\}$. Similarly, $\{a_1\} \oplus \{a_2\} \oplus \{a_1, a_2\} = \{a_1, a_2\}$.

Logical operators: As suggested above, MRS per se does not have an extensional interpretation, so that the meaning of quantifiers, for instance, is not defined. This allows us to set a meaning for some operators and not others, as needed. This property is important as we do not want to assume that a speaker necessarily masters such operators. Quantifiers are a case in point, being acquired relatively late by children (Inhelder and Piaget, 1964; Hollander et al., 2002). For the sake

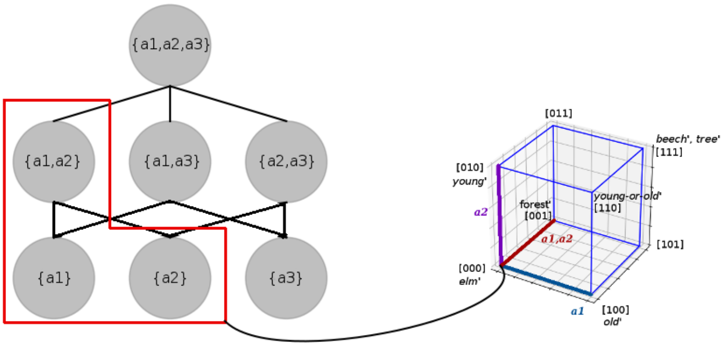


Fig. 2: An example join-semilattice with three atomic individuals and their pluralities. The cube from Fig. 1 corresponds to the subset of the semilattice with individuals $\{\{a_1\},\{a_2\},\{a_1,a_2\}\}$. The entire semilattice would fit into a space of dimensionality 7, to accommodate all its nodes.

of illustration, we will however treat $\exists$ and $\forall$ here as having their standard first-order logic formalisations, leaving other operators to be discussed later in this paper.

Assignment: Additionally, we will define an assignment function α which maps variables in the MRS representation to actual objects in the universe. For instance,

$$\alpha(x_{34}) = \{\{a_2\},\{a_3\}\}$$

Objects can be plural, so we might also have

$$\alpha(x_{34}) = \{\{a_2\},\{a_3\},\{a_1,a_3\}\}$$

Substitution: In combination with an assignment function, we will posit a substitution function Σ_U^α operating over MRS logical forms, which expands out quantifiers, mapping each variable bound by the quantifier(s) to the object in U given by the assignment function α. Given some assignment $\alpha(x) = \{a_1...a_n\}$ and a proposition Φ corresponding to $\forall x\phi(x)$, the substitution $\Sigma_U^\alpha(\Phi)$ returns the MRS $\{\phi(a_1),\phi(a_2),...,\phi(a_n)\}$ (i.e., a conjunction). Given some assignment $\alpha(x) = \{a_1...a_n\}$ and a proposition Φ corresponding to $\exists x\phi(x)$, the substitution $\Sigma_U^\alpha(\Phi)$ returns a set of MRSs $\{\phi(a_1)\},...,\{\phi(a_n)\}$ interpreted as a disjunction.

To take an example, if $\alpha(x_{34}) = \{\{a_2\},\{a_3\}\}$ and we have the logical form $\{all(x_{34}), elm(x_{34}), old(x_{34})\}$,[2] and *all* is defined in the logic as the standard $\forall$, then we obtain the following set of substitution instances with a single logical form:

$$\{\{\text{elm}'(a_2),\text{elm}'(a_3),\text{old}'(a_2),\text{old}'(a_3)\}\}$$

For the logical form $\{some(x_{34}), elm(x_{34}), old(x_{34})\}$, assuming that *some* corresponds to $\exists$, we would have a set of substitution instances containing two logical forms:

$$\{\{\text{elm}'(a_2),\text{old}'(a_2)\},\{\text{elm}'(a_3),\text{old}'(a_3)\}\}$$

The purpose of the substitution is to gain a representation of the properties / relations that apply to individuals in the universe, *according to the sentence* (which may or may not be true) and given a certain assignment. Truth values are then computed individually over the substitution instances, as we will show below. Note that after substitution, we use the prime notation over predicates to show that they now have an extensional interpretation (which, we recall, they did not have in the MRS). We will talk of 'substituted EPs' to refer to the translation of individual MRS elementary predications in the substituted instances.

Truth: Finally, we can compute the truth value of a MRS logical form Φ according to the obtained substitution instances. We will use the notation $\models_M^\alpha \Phi$ to say that Φ is true, and $\not\models_M^\alpha \Phi$ otherwise. Given a proposition Φ corresponding to $\forall x\phi(x)$ (universally quantified), we have $\models_M^\alpha \Phi$ iff every substitution instance in the set $\Sigma_U^\alpha(\Phi)$ is true. Given a proposition Φ corresponding to $\exists x\phi(x)$ (existentially quantified), we have $\models_M^\alpha \Phi$ iff some substitution instance in the set $\Sigma_U^\alpha(\Phi)$ is true.

4 A distributional account of semantic competence

In this section, we formally introduce our definition of a speaker's semantics. Our formalisation is to be given in a distributional framework and thus naturally fits in the Kripkean *causal theory* of competence (§2.2), which simply states that competent usage follows from exposure to performance data. We however also demonstrate in §4.1 that the account

[2] As pointed out in §3.2, MRS captures the scoping of quantifiers via additional notation on the elementary predications. In this case, the full scoped expression would be: $l0 : all(x,h1,h2), l1 : elm(x), l2 : old(x), h1 = l1, h2 = l2, top = l0$, where $h1$ shows the restriction of the generalized quantifier and $h2$ the scope. In the spirit of clarity, we do not show this in the examples.

can model the idea of *truth-theoretic super-competence* introduced by Partee (1979). Further, we also show in §5.3 that it allows us to retrieve the all-important *lexical relations* of Katz and Fodor (1963).

In a nutshell, our proposal is to redefine set-theoretic models as DSMs with the following shape:

$$M =< P_L, U, B_M, C_{G,L,\Sigma_U^\alpha}, X_M, A_M, S_M >$$

Note that we are now using the subscript M instead of D for the model's components, to clarify the difference between a standard distributional model, which computes statistics over a real corpus, and the approach proposed here, which computes truth values within an truth-theoretic language. The components of the model are as follows:

- $P_L = \{P_1 ... P_m\}$ the predicates of a logic;
- $U = \{\{a_1\}...\{a_n\}...\{a_1, a_2\}...\}$ a given universe with n atomic objects and the pluralities computable over those objects;
- B_M the vector basis of the model's space;
- C_{G,L,Σ_U^α} a corpus of substitution instances;
- $X_M : (P_L \cup U)^* \rightarrow ((P_L \times U) \rightarrow \{0, 1\})$, an extraction function attributing truth values to pairs of predicates / entities and returning a predicate by entity matrix (the *ideal entity matrix*);
- $A_M : ((P_L \times U) \rightarrow \{0, 1\}) \rightarrow ((P_L \times P_L) \rightarrow \mathbb{N}_0)$, an aggregation function returning a predicate by predicate matrix (the *ideal predicate matrix*);
- $S_M^U : (U \times U) \rightarrow \mathbb{R}$ and $S_M^P : (P_L \times P_L) \rightarrow \mathbb{R}$, two similarity functions acting over the entity or predicate matrices.

To illustrate the general idea, we can come back to Fig. 1, in which we see a cube that corresponds to a model M extracted from some corpus, with U the universe of 3 individuals expressed by the 3-dimensional basis B_M of that space, and P_L a set of predicates labelling the vertices of the cube. The right of the figure shows the corresponding matrix form of that space. The values in the cells of the matrix are the result of applying X_M to P_L and U: they tell us which properties attach to which individuals.

We will now explain how to derive the above definitions.

4.1 Formalisation of the super-competent speaker

Following Partee (1979), let's assume the existence of an ideal, truth-theoretic speaker – some godly being who knows what there is the world (i.e. has perfect ontological knowledge of the universe U) and knows how to name things (i.e has a perfect, deterministic interpretation function $||.||$). This speaker, according to Partee, might be said to have some semantic (truth-theoretic) super-competence. We will now show that such an ideal speaker can straightforwardly generate a truth-theoretic boolean vector space of the type shown in Fig. 1, that is, a model encapsulated by a high-dimensional hypercube.

The following contains a fair number of formal definitions, but the overall intuition of our method is extremely simple. Our godly being has a grammar, as defined in §3.2. He or she can generate all sentences allowed by that grammar, compute their substitution instances and the truth values of those substitutions, as shown in §3.3. The result of this procedure is the set of all sentences allowable by the godly being's language, marked as True or False. Our goal is to show that this information can be formalised as a vector space. We will first go through definitions and then provide a practical example of their applications in §4.2.

Let us define the language that can be produced by generating all valid sentences with our grammar G.[3] We will call this set of sentences C_G and simply refer to it as **language**. Let us also define the MRS representations of the sentences in C_G as a set of logical forms $C_{G,L}$. For each sentence in C_G, we have a unique underspecified MRS representation in $C_{G,L}$. We will call the set of logical forms in $C_{G,L}$ the **minimal logic language**. Using our notion of substitution Σ_U^α, each MRS in $C_{G,L}$ can be converted to its substitution instance, where objects replace variables (§3.3).

Let us define C_{G,L,Σ_U^α} as the set of substitution instances obtained by passing each logical form in $C_{G,L}$ through Σ_U^α. This set of substituted logical forms will be called the **substitution language**. The truth of each proposition in C_{G,L,Σ_U^α} can be computed given a particular assignment. We will call the combination of C_{G,L,Σ_U^α} and the corresponding truth value assignments a **truth-theoretic language**, denoted by $\mathcal{T}$. That is, $\mathcal{T} =< C_{G,L,\Sigma_U^\alpha}, ||.|| >$. As we see, the truth-theoretic language structure is very close to the definition of a model $M =< U, ||.|| >$. While the truth-theoretic language is a set of substitution instances over logical forms, together with an interpretation function, the model is the universe itself associated with the same interpretation function.

Now, let's note that C_G, $C_{G,L}$ and C_{G,L,Σ_U^α} are nothing other than 'corpora' of sentences, at different levels of representation. That is, we can define a distributional semantics model (DSM) over any of them. We will now produce a semantic space from $C_{G,L,}$, using our definition of a DSM as

$$D =< T_D, O_D, B_D, C_D, X_D, A_D, S_D >$$

Let T_D be the predicates of our logic, that is, $T_D = P_L$. Our DSM contexts will be the objects in our universe, so $O_D = U$. Our corpus C_D is C_{G,L,Σ_U^α}. We will define an extraction function X_D so that

$$X_D : (P_L \cup U)^* \rightarrow ((P_L \times U) \rightarrow \{0, 1\})$$

[3] Technically, these are sentences combined with their associated syntactic derivation.

X_D returns 0 whenever the truth value of a substituted EP (i.e. a proposition) is false, and 1 otherwise. As in standard distributional semantics, it results in a matrix of target-context pairs: a semantic space. For instance, the cell of the matrix at the intersection between row elm and column a_2, written as $elm \times a_2$, corresponds to the truth of the proposition $\text{elm}'(a_2)$ (e.g. 1 if it is true that $\text{elm}'(a_2)$). We will call the resulting matrix the **ideal entity matrix**, that is, the vectorial representation of the truth-theoretic language, expressed in terms of context entities.

Finally, we can define an aggregation function A_D which groups context elements by predicate (e.g. all objects that are elms are aggregated into a single elm' context):

$$A_D : ((P_L \times U) \rightarrow \{0,1\}) \rightarrow ((P_L \times P_L) \rightarrow \mathbb{N}_0)$$

This aggregation function returns a matrix of predicates by predicates, as standard distributional models do. We will call this aggregated matrix the **ideal predicate matrix**.

Two variants of the similarity function S_D can be straightforwardly defined over the space, before and after aggregation: one computing similarity over targets (predicates), another one over contexts (entities). That is,

$$S_D^P : (P_L \times P_L) \rightarrow \mathbb{R} \text{ and } S_D^U : (U \times U) \rightarrow \mathbb{R}$$

The semantic space obtained from passing C_{G,L,Σ_U^α} through X_D is nothing other than a model, expressed in vector form. But a range of distributional semantics techniques can now be applied to that model.

4.2 (Imperfect) illustration

We will now show the use of our formalisation on a simple example. By virtue of being 'simple', this example will fall short of producing an instance of ideal competence (we will discuss later in which ways exactly it is defective). But the exercise will nevertheless provide an illustration of the definitions we laid out in the previous section.

We will define a grammar and a logic as shown in Fig. 3 and 4. The predicates in P_L straightforwardly correspond to equivalent terminals in T_G. In L_L, $\exists$ corresponds to $a(n)$ and $\forall$ to all. We will also introduce a small model $M =< U, ||.|| >$ to match G, shown in Fig. 5. The universe in that model consists of six individual objects, all trees. Those objects can be old or young, and they are elms, beeches or oaks. The objects are labelled $a_1 ... a_6$ and since they are all trees, our universe U can be defined as the extension of $tree$ which, to include plurality, will be written as $^*tree$. That is, $U = {}^*tree$, which is

$$\{\{a_1\},\{a_2\}...\{a_3,a_4\}...\{a_1,a_2,a_3,a_4,a_5,a_6\}\}$$

Fig. 6 shows the interpretation of each predicate in P_L.[4]

Computing languages

C_G is the language that can be generated with G, that is, all the valid sentences obtainable from the grammar:

> C_G ={'an elm is old', 'an elm is young', 'a tree is old', 'a tree is young', 'an oak is old', 'an oak is young', 'a beech is old', 'a beech is young', 'all beeches are old', 'all beeches are young', 'all trees are old', 'all trees are young', 'all oaks are old', 'all oaks are young', 'all elms are old', 'all elms are young'}

(Note that our small grammar does not have a rule $VP \longrightarrow V \quad NP$, so sentences such as *An elm is a tree* are not generated. We will come back to this point later in the paper.)

The minimal logic language $C_{G,L}$ is the translation of C_G into MRS (variables are allocated as sentences are encountered, and we have $|C_G| = 16$):

> $C_{G,L} =$
> $\{a(x_1), elm(x_1), old(x_1);$
> $a(x_2), elm(x_2), young(x_2);$
> ...
> $all(x_{15}), elm(x_{15}), old(x_{15});$
> $all(x_{16}), elm(x_{16}), young(x_{16})\}$

The substitution language C_{G,L,Σ_U^α} is the set of substitution instances for the logical forms in $C_{G,L}$. It must be computed for each possible assignment α. Let's consider for instance the first MRS above, $a(x_1), elm(x_1), old(x_1)$. An assignment function α can associate six different entities with x_1: $x_1 \rightarrow a_1, x_1 \rightarrow a_2, x_1 \rightarrow a_3, x_1 \rightarrow a_4, x_1 \rightarrow a_5$, or $x_1 \rightarrow a_6$. This corresponds to six different substitution instances $\{\{\text{elm}'(a_1), \text{old}'(a_1)\}\}$ ... $\{\{\text{elm}'(a_6), \text{old}'(a_6)\}\}$. The assignment can be to sums of individuals: if $x_{16} \rightarrow \{\{a_2\},\{a_3,a_4\}\}$, then the substitution instance of

$$all(x_{16}), elm(x_{16}), young(x_{16})$$

is
$$\{\{\text{elm}'(a_2), \text{young}'(a_2), \text{elm}'(a_3,a_4), \text{young}'(a_3,a_4)\}\}$$

We note that if we had to write down a complete grammar G, we would have to deal with the fact that C_G may contain an infinite number of sentences. This is due to recursive grammar rules of the type $N \quad \longrightarrow \quad A \quad N$ which might return sentences such as *A young (young)* ... tree is young* (where the Kleene star indicates an indefinite number of repetitions of *young*). Thus the substitution language

[4] Note that for simplicity, we will not cover predicates with multiple arguments in this paper. We can generalise our framework to such predicates by assuming that our unary elementary predications can take ordered tuples as their (single) argument. For instance, chase' may be represented in a substituted EP as $\text{chase}'(A)$ where $A =< \{a_1\},\{a_2\} >$. This allows us to preserve the convenience of a two-dimensional matrix.

$G =< T_G, N_G, R_G, S_G >$:

$T_G = \{tree(s), beech(es), oak(s), elm(s), old, young, is, are, a(n), all\}$

$N_G = \{A, N, NS, V, VS, Det, NP, NPS, VP, VPS\}$

R_G is the following set of rules:

$$\begin{array}{lll|lll}
S_G & \longrightarrow & NP \quad VP & a(n), all & \longleftarrow & Det \\
S_G & \longrightarrow & NPS \quad VPS & tree, beech, oak, elm & \longleftarrow & N \\
NP & \longrightarrow & Det \quad N & trees, beeches, oaks, elms & \longleftarrow & NS \\
NPS & \longrightarrow & Det \quad NS & old, young & \longleftarrow & A \\
VP & \longrightarrow & V \quad A & is & \longleftarrow & V \\
VPS & \longrightarrow & VS \quad A & are & \longleftarrow & VS
\end{array}$$

Fig. 3: A grammar

$L =< P_L, L_L, V_L >$:

$P_L = \{tree, beech, oak, elm, old, young\}$

$L_L = \{\wedge, \exists, \forall\}$

$V_L = \{x_1, x_2, x_3, x_4, x_5, x_6, ...\}$

Fig. 4: A logic

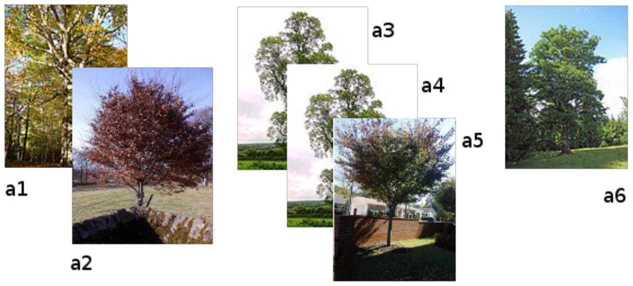


Fig. 5: A world of trees

$$\begin{array}{ll}
\text{beech}' & = \{a_1, a_2\} \\
\text{elm}' & = \{a_3, a_4, a_5\} \\
\text{oak}' & = \{a_6\} \\
\text{tree}' & = \{a_1, a_2, a_3, a_4, a_5, a_6\} \\
\text{old}' & = \{a_1, a_3, a_4, a_6\} \\
\text{young}' & = \{a_2, a_5\}
\end{array}$$

Fig. 6: Predicate extensions

C_{G,L,Σ_U^α}, even for a universe with a finite number of entities, may consists of an infinite number of propositions. This is in line with the idea of competence as the ideal system that allows a speaker to generate and interpret a potentially infinite number of sentences. The actual performance of a speaker, bounded in particular by memory limits and processing capacity, will only include a finite subset of those sentences.

The ideal entity matrix

Let's now create an entity matrix from C_{G,L,Σ_U^α}. Our space has dimensions U. It contains the following target vectors:

$$P_L = \{tree, beech, ..., young\}$$

We can use the extraction function X_M to compute the truth value of each EP in C_{G,L,Σ_U^α}. For instance, the proposition $\text{elm}'(a_1)$ in $\{\{\text{elm}'(a_1), \text{old}'(a_1)\}\}$ evaluates to True because a_1 is in the set of elms. The proposition $\text{elm}'(a_3, a_4)$ also evaluates to True because the set $\{a_3, a_4\}$ is in a subset of elms.

Because of space constraints, we cannot print the whole vector space here. We will first consider the subset U' of U containing singletons only. We will then show an example representation with a plurality, pointing out the relevance of the formalisation for dealing with collectivity and distributivity. Of course, in the spirit of modelling the truth-theoretic language, dimensions should actually be available for each possible plurality.

The semantic space for U', where

$$U' = \{\{a_1\}, \{a_2\}, \{a_3\}, \{a_4\}, \{a_5\}, \{a_6\}\}$$

is shown on the left of Table 1. The matrix can be read 'by row' as well as 'by column'. Row beech' returns all the objects which are beeches, that is, the extension of *beech*: $||beech|| = \{a : \text{beech}' \times a = 1\}$ (the individuals a so that the matrix cell $\text{beech}' \times a$ has a value of 1). Similarly, column a_3 returns all the predicates that are true of a_3. Whether the label of a particular column is in the set of things denoted by the label on a particular row is given by the value in the corresponding cell.

The right of Table 1 shows us an example with three singletons and two plurals (we assume our grammar has been expanded to accommodate the relevant sentences in C_G). We have also added a new predicate forest' and for the sake of illustration, we will arbitrarily posit that three trees or more can be referred to as a (very small!) forest. This is retrievable from the representation: the set $\{a_1, a_2, a_3\}$ has a weight of 1 on the predicate forest', but the set $\{a_1, a_2\}$ hasn't. To get the set of beeches when considering plurals, we perform a Linkian sum operation on the objects which

	a_1	a_2	a_3	a_4	a_5	a_6
beech’	1	1	0	0	0	0
elm’	0	0	1	1	1	0
oak’	0	0	0	0	0	1
tree’	1	1	1	1	1	1
old’	1	0	1	1	0	1
young’	0	1	0	0	1	0

	a_1	a_2	a_3	a_1, a_2	a_1, a_2, a_3
beech’	1	1	0	1	0
elm’	0	0	1	0	0
tree’	1	1	1	1	1
old’	1	0	1	1	1
young’	0	1	0	0	0
forest’	0	0	0	0	1

Table 1: **Left:** Entity matrix, representation of model M, as extracted from C_{G,L,Σ_U^α}. **Right:** variation with plurals and a collective predicate.

	beech’	elm’	oak’	tree’	old’	young’
beech’	2	0	0	2	1	1
elm’	0	3	0	3	2	1
oak’	0	0	1	1	1	0
tree’	2	3	1	6	4	2
old’	1	2	1	4	4	0
young’	1	1	0	2	0	2

Table 2: Aggregated version of the distributional model in Table. 1.

have a weight of 1 in a particular row. So the extension of *beech* is $\sigma a\, beech\, a = \{a_1\} \oplus \{a_2\} \oplus \{a_1, a_2\} = \{a_1, a_2\}$.

Let’s now remark that the predications that are applied to plural individuals are underspecified: a weight of 1 on the dimension beech$'$ for $\{a_1, a_2\}$ does not explicitly tell us whether the predicate should operate distributively or collectively on the plural. Note that forest$'$ also has a weight of 1 on dimension $\{a_1, a_2, a_3\}$, but while a_1 and a_2 are *distributively* beeches, a_1, a_2 and a_3 are *collectively* a forest. Doing things this way allows us to have a more compact representation. However, we can easily *infer* the predicate status by unpacking the plural object and checking the weight of its component singletons on the relevant dimension. For example, the plural $\{a_1, a_2, a_3\}$ has a weight of 1 on both tree$'$ and forest$'$. We can however find out that tree$'$ acts distributively by noticing that $\{a_1, a_2, a_3\} = \{a_1\} \oplus \{a_2\} \oplus \{a_3\}$ and that tree$' \times a_1 = 1$, tree$' \times a_2 = 1$ and tree$' \times a_3 = 1$. Conversely, all of the entities a_1, a_2 and a_3 have a weight of 0 on the forest$'$ dimension, so forest$'$ acts collectively.

Aggregation function

Applying the aggregation function A_D to the space shown in Table 1 (left), we get the symmetric predicate matrix shown in Table 2. The cells in the diagonal of the matrix show the cardinality of the sets denoted by the predicate on the respective rows / columns. For example, the cell tree$' \times$ tree$'$ tells us that our universe contains six trees.

We can verify that the vector representation of beech$'$ ($[2, 0, 0, 2, 1, 1]$) is simply the pointwise addition of the columns for the beech objects in Table 1 (a_1 and a_2). Note that when performing this operation over plurals and collectives, we must perform a Linkian sum operation rather than simple addition. But the principle behind aggregation remains the same.

Similarity function

Vectors allow the use of standard distributional approaches to similarity. In the predicate matrix, similarity can be computed over target words as $S^P_M : (P_L \times P_L) \to \mathbb{R}$. The similarity between two lexemes corresponds to the degree to which their semantic properties are shared. For instance, oaks are more similar to elms than to beeches because they are more likely to be old, given our observations. It is also possible to compute similarity from the entity matrix. One particularly useful computation may be the similarity between objects S^U_M, allowing the model to compute spatial distance between any two individual or plural objects. As we will see later, this ability also relates to the formal definitions of antonymy and word senses (§5.3).

Compare this with DSMs based on word co-occurrences, where similarity essentially corresponds to the degree to which two lexemes share usage patterns. While S^P_M is derived from extensional information, it does also naturally capture lexical information: old$'$ and young$'$ are somewhat similar because they both apply to instances of beech$'$ and elm$'$, that is, they would both be found in sentences such as *an elm is old* or *an elm is young*. Thus, the similarity shown here does capture some ‘word co-occurrence’ information, as they would be observed in declarative sentences. This is an important point because it allows us to see our proposed truth-theoretic model as a special case of standard DSMs (as described in §3).

Fig 7 (left) shows a similarity heatmap for the entity matrix from Table 1. Each square of the heatmap shows how related the entities in the corresponding row and column are (as calculated using cosine similarity). We can see that a_3 and a_4, since they have identical vectors, display maximum similarity. Fig 7 (right) shows a similarity heatmap for the predicate matrix obtained in Table 2. We see from that heatmap that there is a weak similarity between oaks and young things, due to the fact that oaks are never young.

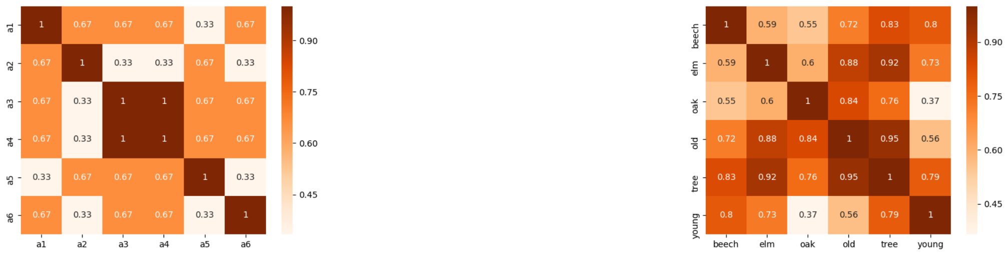


Fig. 7: **Left:** Similarity heatmap for ideal entity matrix in Table 1. **Right:** Similarity heatmap for ideal predicate matrix in Table 2.

4.3 Relation to performance

So far, we have presented our theoretical framework from the formal and ideal point of view of ‘super-competence’, that is, assuming a speaker with perfect ontological knowledge. In order to show that it is amenable to computational treatment, we now need to inspect its properties with respect to human, ‘non-godly’ competence. In particular, we must consider the fact that linguistic competence has to be *acquired* (by a human or a machine) and that our model must accommodate a speaker’s expanding information state and linguistic knowledge.

Let us come back to our definition of competence in terms of a set of utterances. A natural question that may be asked about our proposal is whether our object space could not be directly built from the model representation, in a grammar-free fashion: if the set of beeches is included in the set of trees *in the model*, we should be able to derive the equivalent vectors without going through the hassle of producing a corpus of sentences. In other words, if we have a model $M =< U, ||.|| >$, why do we need the truth-theoretic language $\mathcal{T} =< C_{G,L,\Sigma_U^\alpha}, ||.|| >$?

The simplest answer to this question may just be that in actual fact, non-godly beings *do not* have access to either U or $||.||$. In humans, U is *incomplete* because no one has complete ontological knowledge. U may also be biased in various ways because a lot of what we know about the world comes from ‘being told’ rather than having direct perceptual experience of the relevant situations – or alternatively because our perception and inferential abilities are themselves imperfect. $||.||$ is similarly deficient, partially for the same reasons, but also because some predicates are more difficult to model truth-theoretically than others. Abstract terms are probably the most obvious area of difficulty. But we also note classic disagreements across speakers, such as the notorious cup / mug example (what is a cup for me may be a mug for you: Labov, 1978).

Perhaps less obviously, the semantics that the speaker acquires should be the semantics *of their language*, that is, a particular rather than a universal semantics, which matches the speaker’s grammar at the syntax / semantics interface. Arguably, a semantics directly based on a true model of the world is too powerful and will not account for cross-linguistic variability.[5] This has an important consequence for the completeness property of the truth-theoretic language $\mathcal{T} =< C_{G,L,\Sigma_U^\alpha}, ||.|| >$. In order to be a complete description of the world, $\mathcal{T}$ would require some ‘ideal grammar’. Such a grammar may be *more* than the grammar of a competent speaker, in that it would presumably include an ideal lexicon and an ideal set of composition rules which would afford an ontologically perfect representation of *what there is*.

To make this point clearer, it suffices to inspect the completeness of C_{G,L,Σ_U^α} with respect to the corresponding entity matrix. Let’s recall that C_{G,L,Σ_U^α} are the substitution instances of logical forms that are obtained by parsing the sentences in C_G. The entity matrix is the result of putting C_{G,L,Σ_U^α} through a truth-theoretic extraction function X_M. If we look again at the entity matrix on the left of Table 1, we note that we can easily generate propositions from the matrix, with their associated truth values. Specifically, the set of true propositions given assignment α, written as $\{\phi :\models_M^\alpha \phi\}$, is given by all possible combinations of object / predicate pairs with a value of 1 in the object matrix: $\{\phi :\models_M^\alpha \phi\} = \{\{a, P\} : P \times a = 1\}$. More generally, the truth of a random proposition $\phi = \{\{P_1(a_1), P_2(a_2), ..., P_k(a_k)\}\}$ is the product of the truth values of its (substituted) EPs: $(P_1 \times a_1)(P_2 \times a_2)...(P_k \times a_k)$. As expected, this product will be 0 if one single EP is false.

Let’s now illustrate what this means in terms of our example object matrix in Table 1 by generating a few propositions from this matrix by simply picking up, for each proposition, a number of random cells:

$$\phi_1 = \{\{\text{beech}'(a1)\}\}$$
$$\phi_2 = \{\{\text{beech}'(a1), \text{tree}'(a1)\}\}$$
$$\phi_3 = \{\{\text{elm}'(a2), \text{tree}'(a2), \text{old}'(a4)\}\}$$

[5] We thank Nicholas Asher for this insight.

ϕ_1 is true because $beech \times a1 = 1$. ϕ_2 is also true because $beech \times a1 = 1$ *and* $tree \times a1 = 1$. ϕ_3 is false because $\text{elm}'(a2) = 0$.

One important observation about this exercise is that we are generating true propositions which are not derivable from our original corpus C_G. Note, for instance, that $\phi 2$, which might roughly be expressed as the sentence *There exists a beech which is a tree*, is *not* in C_G. This happened for the simple reason that our grammar G, as we have set it up, does not include a rule $VP \rightarrow V \quad NP$. This illustrates an important point: a true description of the world is not necessarily a comprehensive one. The set of true sentences that can be generated from a given grammar, as given by the truth-theoretic language $\mathcal{T}$, may not correspond to what the speaker *knows* about the world. In other words, syntactic competence and semantic competence may be out of sync.

Putting these considerations together, we see that we must downgrade our idealised notion of **super-competence** $M =< U, ||.|| >$ by acknowledging that, in a real speaker, mastery of U and $||.||$ is imperfect, resulting in a notion of **human competence** $M^H =< U^H, ||.||^H >$, where knowledge of the universe is limited to a certain information state. By extension, the **truth-theoretic language** $\mathcal{T} =< C_{G,L,\Sigma_U^\alpha}, ||.|| >$ can itself be considered bounded by the speaker's grammatical competence, with the interpretation function being perfect *for a given state of grammar* (and logic), as we've shown above. This results in **human language**,

$$\mathcal{T}^H =< C^H_{G,L,\Sigma_U^\alpha}, ||.||^H >$$

i.e. the sentences that a speaker is able to parse and/or generate given their grammatical competence, together with that speaker's belief about their truth values.

This, now, looks very much like *performance*: a corpus of grammatically imperfect utterances, mapped to an incomplete universe associated with an equally flawed interpretation function. The consequence of this is that the general structure of our formalisation can be retained when learning from standard corpora and/or grounded data. We will give a concrete example of this in §6.

5 Features of the semantics

We now consider features of our semantics, including its relation to compositionality, its amenability to probabilistic treatments, and the way it encodes lexical relations.

5.1 Composition

A fully compositional account of our framework is beyond the scope of this paper, but we will sketch how some of the relations typically considered in distributional semantics can be modelled using our approach. In particular, we will exemplify how composition returns both a set-theoretic representation of the composed constituents and still preserves our expectations of distances in the vector space.

We will use two operators when performing composition, which act over elements of the basis B_M. The sum operator $+$ performs disjunction: for instance, when we pick out the denotation of *young or old*, we select all dimensions activated by the predicate young' and add all dimensions activated by the predicate old', obtaining a subspace of dimensionality equal to the number of individuals in the *young* semi-lattice *plus* the number of individuals in the *old* semi-lattice. In contrast, the pointwise multiplication operator $\odot$ performs conjunction: in our boolean vector space, whenever we multiply two vectors, any dimension where one of the vectors has weight 0 will be set to 0, in effect making that dimension redundant to the interpretation of the predicate. For instance, in our cube in Fig 1, multiplying tree' with young' results in the vector $[111] \odot [010] = [010]$, effectively 'cancelling out' the first and third dimensions from the interpretation. The resulting universe of utterance consists of a unidimensional subspace corresponding to individual a_2 (the young tree).

What follows is a translation of a standard (simple) formal semantics account of composition into a vector account. We will assume an account of the syntax-semantics interface where each category in the grammar has a corresponding type $T \in G_R$ in the semantics. Semantic types have two main features. First, they have *argument slots* that can be filled by constituent vectors. Those slots are initially filled by a vector of 1 values (written as $\vec{1}$) and are related in the type by either the $+$ or $\odot$ operator, as explained above. For instance, the conjunctive *and* involves two arguments and the $\odot$ operator: $\vec{1} \odot \vec{1}$. Filling an argument slot with a predicate involves pointwise multiplication of the predicate vector with $\vec{1}$, resulting in the predicate itself. For instance, $[0, 1, 0] \times \vec{1} = [0, 1, 0]$. Arguments slots that remain unfilled are thus $\vec{1}$. Second, operations have to be wrapped in some function $b(\vec{v})$, the role of which is simply to return values above 1 to 1 (this is necessary because addition of predicates may result in non-boolean vectors):

$$b(\vec{v}_i) = \begin{cases} 0, & \text{if } \vec{v}_i = 0 \\ 1, & \text{otherwise} \end{cases}$$

The definitions below are given with respect to the ideal *entity* matrix, unless stated otherwise.

Intersective composition in phrases: Intersective composition has type $b(\vec{1} \odot \vec{1})$: the extension of young elms, for instance, is simply given by the pointwise multiplication of the vectors for elm' and young'. We can verify this in the entity

	beech'	elm'	oak'	tree'	old'	young'	old-elm'	old-oak'	young-beech'
beech'	2	0	0	2	1	1	0	0	1
elm'	0	3	0	3	2	1	2	0	0
oak'	0	0	1	1	1	0	0	1	0
tree'	2	3	1	6	4	2	2	1	1
old'	1	2	1	4	4	0	2	1	0
young'	1	1	0	2	0	2	0	0	1
old-elm'	0	2	0	2	2	0	2	0	0
old-oak'	0	0	1	1	1	0	0	1	0
young-beech'	1	0	0	1	0	1	0	0	1

Table 3: Aggregated version of the distributional model, with example of intersection.

matrix shown on the left of Table 1:

$$\begin{aligned}\text{young-elm}' &= b([0,1,0,0,1,0] \odot [0,0,1,1,1,0]) \\ &= [0,0,0,0,1,0]\end{aligned}$$

There is a single 1 in the resulting vector, corresponding to a subspace with a unique dimension a_5: that is, the set of young elms is the singleton $\{a_5\}$.

Conjunction: Conjunction is also of type $b(\vec{1} \odot \vec{1})$ (the conjoined predicates p_1 and p_2 both apply to the same entity). It operates essentially as intersective composition, corresponding to the pointwise multiplication of the coordinated vectors. For instance:

$$\begin{aligned}\text{elm-and-beech}' &= b([0,0,1,1,1,0] \odot [1,1,0,0,0,0]) \\ &= [0,0,0,0,0,0] = \vec{0}\end{aligned}$$

That is, nothing is both an elm and a beech.

Disjunction: Disjunction is of type $b(\vec{1} + \vec{1})$ (either p_1 or p_2 applies to the entity). Extensionally, the set of things that are elms or beeches is $\{a : \text{elm}' \times a = 1 \wedge \text{beech}' \times a = 1\}$. This extension can be computed by simple vector addition, passed through $b(\vec{v})$ (so that the resulting vector remains boolean even when values are greater than 1). For example, we can get the representation for *elms or beeches* by summing the relevant vectors:

$$\begin{aligned}\text{elm-or-beech}' &= b([0,0,1,1,1,0] + [1,1,0,0,0,0]) \\ &= [1,1,1,1,1,0]\end{aligned}$$

The resulting vector tells us that the entities that are elms or beeches are $\vec{1}$ in basis $\{a_1, a_2, a_3, a_4, a_5\}$.

Negation: Negation of a predicate corresponds to type $b(\vec{1})^{-1}$, where the exponent indicates that the selected basis is the complement of the negated constituent's basis. For instance:

$$\neg\text{young}' = b([0,1,0,0,1,0])^{-1} = [1,0,1,1,0,1]$$

The resulting vector selects $\vec{1}$ in basis $\{a_1, a_3, a_4, a_6\}$ which correspond to the old trees in our model.

Quantification: Quantifiers are a binary structure with two arguments, a restrictor and a scope. All quantifiers have the same type $b(Q(\vec{1} \odot \vec{1}))$. We note that the truth value of a quantified statement can be obtained via pointwise multiplication of the restrictor and scope. For instance, we may have:

$$\begin{aligned}&\models^{\alpha}_{M} \textit{all}(\text{N}' \odot \text{VP}') && \text{iff } \text{N}' \odot \text{VP}' = \text{N}' \\ &\models^{\alpha}_{M} \textit{some}(\text{N}' \odot \text{VP}') && \text{iff } \text{N}' \odot \text{VP}' \neq \vec{0} \\ &\models^{\alpha}_{M} \textit{no}(\text{N}' \odot \text{VP}') && \text{iff } \text{N}' \odot \text{VP}' = \vec{0}\end{aligned}$$

Note that the denotation of the NP (e.g., *all trees*) corresponds to the situation where the second slot of the quantifier is unfilled.

We can also regard quantification as depending on a ratio between set cardinalities, it is possible to use the information from a probabilistic version of the predicate matrix to assess truth values. Such a matrix will be introduced in §5.2 (see Table 5 for an example). Assuming we simply set the meaning of *most* to be 'more than half', then we can read off the matrix that *most trees are old* by noting that $\textit{tree} \times \textit{old} = 0.67 > 0.5$ (see Emerson, 2018 for a probabilistic account of quantifiers similarly compatible with a distributional model).

Similarity: We note that those composition operations return vectors which behave as expected with respect to similarity. For example, using cosine as our similarity measure, and reading from the predicate matrix after aggregation (Table 3), we can derive that old elms are more similar to old oaks than to young beeches:

$$\begin{aligned}&S^P_M(\text{old elm}', \text{old oak}') = \\ &\textit{cos}([0,2,0,2,2,0,2,0,0],[0,0,1,1,1,0,0,1,0]) = 0.5 \\ &S^P_M(\text{old elm}', \text{young beech}') = \\ &\textit{cos}([0,2,0,2,2,0,2,0,0],[1,0,0,1,0,1,0,0,1]) = 0.25\end{aligned}$$

Contextualisation: Finally, we note that by considering the subspace of utterance for particular constituents, we can model contextualisation effects on the lexical meaning of the predicates, in the spirit of other DS approaches (Erk and Padó, 2008; Devlin et al., 2019).

Let us first consider what composition is supposed to achieve, set-theoretically. Given a complex constituent, e.g. *'young or old'*, we want to return the extension of that constituent (or its truth value, at the sentence level). As we have seen before, the denotation of a predicate is the set of dimensions in the entity matrix where the predicate has value 1: the extension of *beech* is given by the dimensions that are beeches. So a denotation is a subset of the entire universe $U = B_M$, and getting the meaning of a constituent involves carving a set of individuals out of the original model hypercube, resulting in a new hypercube corresponding to the *universe of utterance* U_U, that is the set of entities that are actually referred to. (To visualise this: the cube in Fig. 1, reproduced in Fig. 2, is a subset of the 7-dimensional hypercube that expresses the entire semilattice in Fig. 2.)

Formally, we can say that the denotation of a (potentially complex) predicate P′ lives in the basis of a subspace of B_M where P′ $= \vec{1}$. For example, in Fig. 1, the basis formed by $\{a_1\}$ and $\{a_2\}$ contains the denotation of young-or-old′: it defines all individuals that are either young or old and in that 2-dimensional space, young-or-old′ $= [11] = \vec{1}$. Whenever the denotation of P′ is empty, we have a zero-dimensional subspace with basis $\{\vec{0}\}$.

The interesting aspect of the universe of utterance U_U is that it itself forms an entity matrix which describes a closed subset of the entire universe. Applying the aggregation function A_M to that new entity matrix gives us vectors contextualised with respect to U_U. This effect is exemplified in Table 4. We observe in particular that the similarity of elms to beeches after the speaker has heard the utterance *young tree* is now 0.67, compared to 0.59 in U (see heatmap for U in Fig 7). This is to be expected, since in U_U all elms and all beeches are young – in contrast with U where half of beeches and a third of elms are young.

5.2 Probabilistic interpretation and possible worlds

A predicate matrix of the type shown in Table 2 can be easily manipulated to give e.g. a probabilistic notion of set membership. Given enough data, it would for instance be valid to normalise each vector by the cardinality of the target set, giving a representation telling us the probability that a given instance of a set might have such or such property. So as an example, we can take the vector for tree′: $[2, 3, 1, 6, 4, 2]$ and normalise it by $|\text{tree}'| = \text{tree}' \times \text{tree}' = 6$ and obtain vector $[0.33, 0.5, 0.17, 1, 0.67, 0.33]$, telling us that a random tree has a probability of 0.33 to be young. Such a probabilistic predicate matrix is shown in Table 5. Each cell in this matrix is a simple conditional probability of the type $Prob(\text{p1}'(x)|\text{p2}'(x))$: for instance, cell tree′ × young′ corresponds to $Prob(\text{young}'(x)|\text{tree}'(x))$.

Using a probabilistic matrix, we can derive a traditional notion of possible worlds, following e.g. Goodman and Lassiter (2015), who show that possible worlds can be generated by sampling entities which have a certain probability of displaying a certain property. By randomly generating a large number of entity matrices (worlds) which are basically variations on our original universe U, we can define notions of possibility and necessity in the standard formal fashion.

5.3 Formalisation of lexical relations

To finish the exposition of our formalism, we will show that a number of lexical relations can be retrieved from both entity and predicate matrices, satisfying the requirement that a semantically competent speaker should master such relations.

Synonymy: Synonymy relations can be captured from the predicate matrix. Two words with high similarity value in S_M^P can be considered near-synonyms. We would also expect that for a given model, the utterances about two true synonyms such as *aubergine* and *eggplant*, together with their truth values, would form two identical subsets of C_{G,L,Σ_U^α} (and thus two identical vectors with similarity 1). We might also talk of two 'synonymous' entities if they share exactly the same properties (see e.g. a_3 and a_4 in Table 1).

Hyponymy: If A is a hyponym of B, then A′ ⊆ B′. This can be straightforwardly retrieved from the entity matrix, reading the rows and checking for inclusion relations. The inclusion of A′ in B′ can be expressed as a vector relation where A′ ⊙ B′ = A′. For instance, in Table 1 (left), we have elm′ ⊙ tree′ = elm′ so elms are trees. This relation is even easier to retrieve from the probabilistic predicate matrix: if A′ × B′ = 1 then A is a hyponym of B (all the instances of A *have to* be instances of B).

Note that when considering a matrix with plurals and collectives, the inclusion relation above should only be computed over predicates of the same type (either distributive or collective). We refer back to §4.2 for more detail on distinguishing distributives from collectives.

Antonymy: Geeraerts (2010) distinguishes between three basic types of antonymy: gradable, non-gradable and multiple antonyms. The gradable type refers to pairs of terms that describe opposite ends of a scale, for instance *cold* and *hot*. Non-gradable antonyms are those that express a discrete, binary opposition like *dead* and *alive*. The last class, multiple antonyms, refers to terms that denote several discrete points on a non-gradable, discontinuous scale: academic positions (*postdoc*, *lecturer*, *professor*, etc) are an example of such a scale. Binary gradable / non-gradable antonyms usually refer to adjectives, while multiple antonyms can take a variety of forms, including nouns (see above), adjectives (e.g.

$U \rightarrow$

	a_1	a_2	a_3	a_4	a_5	a_6
beech'	1	1	0	0	0	0
elm'	0	0	1	1	1	0
oak'	0	0	0	0	0	1
tree'	1	1	1	1	1	1
old'	1	0	1	1	0	1
young'	0	1	0	0	1	0

$\rightarrow U_u$

a_2	a_5
1	0
0	1
0	0
1	1
0	0
1	1

In U_U,
$S_M(\text{beech}', \text{elm}')$
0.67

$\leftarrow$

	beech'	elm'	oak'	tree'	old'	young'
beech'	1	0	0	1	0	1
elm'	0	1	0	1	0	1
oak'	0	0	0	0	0	0
tree'	1	1	0	2	0	2
old'	0	0	0	0	0	0
young'	1	1	0	2	0	2

$\hookleftarrow$
A_M

Table 4: Composition example. We start from the entity matrix for the entire universe U. The speaker hears *young tree*, which results in a universe retraction operation, ending up in a universe of utterance U_U in two dimensions (a_2, a_5). A new aggregation matrix can be computed by applying A_M to U_U's entity matrix. We note that in that matrix, similarities are different from the values for the entire universe U (0.67 vs 0.59 for the similarity of beech$'$ and elm$'$). The lexical meaning of the predicates has been contextualised to the universe of utterance.

	beech'	elm'	oak'	tree'	old'	young'
beech'	1	0	0	1	0.5	0.5
elm'	0	1	0	1	0.67	0.33
oak'	0	0	1	1	1	0
tree'	0.33	0.5	0.17	1	0.67	0.33
old'	0.25	0.5	0.25	1	1	0
young'	0.5	0.5	0	1	0	1

Table 5: Probabilistic interpretation of the aggregated space in Table 2.

colours) or even verbs (e.g. *walk, jog, run...*). The terms 'taxonomical siblings' and 'co-hyponyms' are sometimes used to refer to multiple antonyms, as they normally are classes of objects that have a common hypernym.

To give a general definition, we can say that antonyms refer to alternative and *incompatible* properties with respect to a particular class of objects, or with respect to a necessary property of that class. For instance, an instance of a living thing cannot be young and old at the same time but it must be one or the other (because having an age is a necessary property of a living thing). The antonymy relation can be found in the probabilistic predicate matrix by identifying groups of mutually exclusive predicates which are included in a common set of objects.

We can see an example of a set of taxonomical siblings in Table 5. The predicates elm$'$, beech$'$ and oak$'$ all have a weight of 1 at their intersection with tree$'$ but a weight of 0 at their mutual intersection (elm$'$ × beech$'$, elm$'$ × oak$'$, beech$'$ × oak$'$). Similarly, young$'$ and old$'$ are mutually exclusive properties of trees.

Formally, let $N = \{P_1...P_k\}$ be a set of predicates and P_C another predicate so that $P_C \not\subseteq N$. N is a set of antonyms if in the probabilistic predicate matrix, for each $p \in N$, $p \times P_C = 1$ and for each $q \in N - p$, $p \times q = 0$. I.e., it is necessary that the predicates in N be instantiations of P_C and it must be impossible that their denotations intersect.

We note that by virtue of relating to a common scale, antonyms are usually lexically related, and their similarity will be somewhat substantial. Note in our toy example that oak$'$ and young$'$ are also mutually exclusive sets and could in principle be considered antonyms (if we disregard the fact that it is unusual to consider antonymy across parts-of-speech). This effect is of course partly due to the size of our sample (we would expect some oaks to be young in a larger model). But perhaps more importantly, we can retrieve from the similarity heatmap in Fig 7 (right) that the similarity between oaks and young things is very low, making them unlikely candidates for antonyms. We will pursue this point further looking at word senses.

Word senses: The notion of word sense is complementary to the general antonymy relation. The biological sense of *tree* should be distinct from its representational sense, for instance. Extensionally, it means that the individuals that are biological trees will not intersect with the individuals that are, say, syntactic trees, that is, as in the antonymy case, we have to find mutually exclusive subsets of a general predicate. Unlike multiple antonyms, however, the discovered clusters may be lexically relatively dissimilar – or even fully dissimilar in the case of homonyms.

Let's give an example. Fig 8 shows the same semantic space as before, but expanded to include two new instances corresponding to syntactic trees. The similarity map for this matrix is shown on the right of the table. We clearly see senses emerging from that map. All vectors in the space are

	beech’	elm’	oak’	tree’	old’	young’	syntactic’
beech’	2	0	0	2	2	0	0
elm’	0	3	0	3	2	1	0
oak’	0	0	1	1	1	0	0
tree’	2	3	1	6	5	1	2
old’	1	2	1	4	4	0	0
young’	1	1	0	2	0	2	0
syntactic’	0	0	0	2	0	0	2

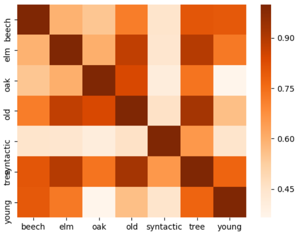


Fig. 8: Predicate matrix for a model including two syntactic trees, and corresponding similarity heatmap.

similar to *tree* (indeed, all *are* trees): this is visible when looking at the row / column for predicate tree′, which contains relatively dark cells. However, we also note a clear dissimilarity between things that are syntactic and other things that are trees: there is a clear ‘light’ line on the row / column for syntactic′, indicating that things that are syntactic are dissimilar to other things that are trees. We may conclude that we are observing two very different types of things which are nevertheless both referred to as ‘trees’, i.e. two sense clusters of the lexical item *tree*.

It is worth noting that this notion of sense is not a lexicographical one. It in fact aligns better with Kilgarriff’s *rejection* of word senses as fixed objects which would have some semantic integrity (Kilgarriff, 1997). Instead, it goes with a notion of sense as ‘sets of usages’, that is, a fuzzy notion of distributional similarity amongst utterances, which can dynamically change over time – an approach usually referred to as ‘meaning in context’ in the computational literature (Erk and Padó, 2008; Dinu and Lapata, 2010).

6 Implementation

We now briefly come back to our original discussion of semantic competence (§2.3), emphasising how the acquisition process should derive from real performance data, and eventually lead to three cornerstones of competence: the ability to refer, the mastery of lexical relations, and a shared intuition for acceptability judgements. This section makes use of results previously published in Herbelot (2020), and relates them to the formalisation presented in this paper.

Herbelot (2020) presents a system nicknamed EVA (Entity Vector Aggregator), which builds an entity matrix and associated predicate matrix from the Visual Genome dataset (VG: Krishna et al., 2017). The idea is that the bounding boxes in the dataset provide access to individual entities and their properties. Each image is taken to represent a ‘situation’. For instance, the first situation in the VG contains a tall brick building, identified by variable 1058508, as well as a black sign situated on that building, identified by variable 1058507. Converting the VG format to MRS, it is possible to obtain logical forms associated with each situation, e.g.:

$\text{building.n}'(1058508), \text{tall}'(1058508), \text{brick}'(1058508),$
$\text{sign.n}'(1058507), \text{black}'(1058507),$
$\text{on}(1058507, 1058508)$

Two-place predicates can be curried into two one-place predicates: the on predicate above becomes $on(1058507, \text{building.n}')$, $on(\text{sign.n}', 1058508)$.

Whilst being somewhat artificial, this type of annotation can be taken as an approximative description of some subset of the real world (that is, the subset encapsulated by the entire image corpus). In other words, it corresponds to some incomplete human language $\mathcal{T}^H =< C^H_{G,L,\Sigma_U}, ||.||^H >$, bounded by a speaker’s knowledge and the type of relations expressible in their grammar. An example of the VG’s incompleteness can be seen in the following instances of bear (objects referents 158539 and 1617277), annotated with various degrees of precision:

158539	bear.n has(-,eye.n) has(-,claw.n) has(-,paw.n) has(-,mark.n) beside(grass.n,-) has(-,ear.n) on(-,land.n) has(-,leg.n) has(-,nose.n)
1617277	bear.n has(-,fur.n) has(-,nose.n)

We see from this example that a learner might not get consistent information about the type of properties that necessarily apply to bears: entity 1617277 is not said to have paws or ears. Similarly, the ‘grammar’ of the VG is restricted to only two ‘rules’: *attributes* (mostly adjectives) and *relationships* (mostly two-place verb and prepositional predicates), taking *objects* as arguments.

Formally, the VG can be represented as a model $M =< P_L, U, B_M, C_U, X_M, A_M, S_M >$ as described in §4.1. We can then write a basic feature structure grammar associating syntactic rules with semantic constructions and their corresponding distributional compositional type, as explained in §5.1. For instance, adjective-noun phrases map onto type $\vec{1} \odot \vec{1}$ (we assume here for simplicity that all adjectives are intersective). Querying the system with e.g. the phrase *brown bear* in this way will return all entities that can be truthfully referred to as brown bears in the Visual Genome. That is, the model naturally encodes resolution of referring expressions (see paper for examples).

The EVA system tests the word vectors from the VG predicate matrix on various tasks, including the identification of lexical relations and the simulation of human acceptability judgements. The system performs in a manner comparable to a large pretrained embedding model, whilst having being exposed to a factor of 10^3 less data (2.8M words in total). This result is interesting because it indicates that the *type* of data a system is trained on can drastically accelerate the learning process. In the scope of the present paper, it may mean that sentences akin to the truth-theoretic language C_{G,L,Σ_U} are 'better' (or at least more efficient) data than large corpora without extensional information.

While the above results only test part of the formalisation presented here, they indicate that the basic features of our entity and predicate matrices are beneficial to acquisition from small grounded data.

7 Conclusion

To conclude this paper, we give a brief account of the specific ways in which our framework relates to other FDS proposals. We particularly emphasise the acquisition of semantic competence as the phenomenon of interest and highlight how this choice makes specific requirements on the formalisation. In doing so, we also highlight the aspects of the framework that require further work.

Meaning as truth-theoretic vectors: our account is close in spirit to Venhuizen et al. (2019), who propose a 'Distributional Formal Semantics' based on truth-theoretic vector representations of propositional meaning. Their meaning space contains propositional vectors defined in terms of a set of models (or possible worlds), and each vector records in which models the proposition is true. One main difference between the two accounts is the choice of a predicate- vs proposition-based semantics. Our reason for prioritising predicate-level co-occurrences in our framework is that we pursue the specific goal of competence acquisition. We ideally want to be able to learn from sentence fragments, for which no truth value is a priori available. In the long term, we want to be able to experiment with different theories of grammar, in particular how generative vs constructionist approaches might play out in the framework. It is therefore advantageous to us to be able to directly represent subpropositional expressions rather than *derive* them from a propositional semantics.

Entities as semantic primitives: entities are core to our proposal – so much so that they form the basis of our vector space model. This design choice is unusual in distributional semantics, where both vectors and dimensions of the semantics space are usually regarded as lexical or 'kind' representations. Entities themselves do not usually belong to the standard DS apparatus, although there are (partial) exceptions (Gupta et al., 2015; Herbelot, 2015). Notably, Kruszewski and Baroni (2015) find a function to map distributional vectors of kinds to 'boolean' vectors in which each dimension roughly corresponds to the notion of an individual. From a representational point of view, this proposal is close to our framework, as the basis of the vector space consists of entity-like objects (although without plurality), and the property vectors are boolean. Emerson (2018) proposes a probabilistic semantics with a space of 'pixies' corresponding to a set of properties and denoting the set of individuals regarded by the speaker as having those properties. The main difference between our work and previous accounts is the way we choose actual individuals in a given universe to be the semantic primitives of our approach. We take the stance that experiences (semantic space dimensions) are primary, and that concepts (vectors) can emerge from them.

Gradation and probabilistic interpretation: a limitation of the present account is the underlying assumption that we are able to tell whether an attribute applies or not to an individual: our extraction function X_M returns boolean values and 'knows' whether e.g. a particular object can be called *red* or anything else. This follows from our simplistic view that all lexical items can be expressed as sets, including gradable predicates, and from the assumption that our interpretation function is perfect and deterministic with respect to the speaker's model of the world. We will relax those assumptions in future work. We note in particular that compatible probabilistic approaches provide useful accounts of a person's information state and beliefs (Erk, 2016; Emerson, 2018; Venhuizen et al., 2019).

Incrementality: an account of competence acquisition should be by nature incremental, and various DS proposals have kept this in mind (Baroni et al., 2007; Kabbach et al., 2019). One aspect of our framework that may be worrying is the exploding number of dimensions in the entity matrix. In principle, a full model would include one dimension per individual, making the model of a speaker at time t as large as the sum of their experiences. A truly incremental version of our framework would thus have to integrate plausible mechanisms of attention and forgetting. We think that the aggregation function A_M could be refined to provide such a service. In particular, we assume that after time, and unless there are pragmatic reasons for them to remain salient to the speaker, individuals would decay into their respective kinds (see Baddeley, 1997 on the role of forgetting for consolidation in long-term memory: we can imagine this process as the 'bottom' of the Linkian semi-lattice fading away). So with respect to world knowledge, a space would always be as large as the long-term memory of the speaker allows.

Ultimately, we hope to expand the existing implementation of our framework to test its features in a realistic simulation of language acquisition. We are particularly interested in the way that child-directed corpora and behavioural datasets can let us investigate the relationship between the 'flawed' performance data a speaker is exposed to and their competence level. We also want to integrate our formalisation into learning algorithms that let us evaluate *which* additional assumptions are necessary to explain the success or failure of the acquisition process under different conditions (what must be innate? where is explicit supervision or correction required?) But for now, we hope to have provided the theoretical frame which will guide hypotheses at the experimental stage.

Acknowledgements This paper is known to some of our friends and colleagues as *the Scottish paper*, in reference to Shakespeare's 'Scottish Play'. Those who understand this reference have our profound gratitude, for all the help and support they have provided over time. We want to specifically thank Katrin Erk for her advice and the many conversations that influenced this work, and for the invaluable comments she provided to version 784 and 787 of our manuscript. We are also indebted to Raffaella Bernardi, who advised us on a point of formalisation at a crucial time. Finally, we apologise to Eva for all the suffering this paper has caused.

Needless to say, we gratefully acknowledge the reviewers who provided us with precious feedback to improve our submission.

Conflict of interest

The authors declare that they have no conflict of interest.

References

Asher N (1988) Semantic competence, linguistic understanding, and a theory of concepts. Philosophical studies 53(1):1–36

Baddeley AD (1997) Human memory: Theory and practice. Psychology Press

Baroni M, Lenci A (2010) Distributional memory: A general framework for corpus-based semantics. Computational Linguistics 36(4):673–721

Baroni M, Lenci A, Onnis L (2007) ISA meets Lara: An incremental word space model for cognitively plausible simulations of semantic learning. In: Proceedings of the Workshop on cognitive aspects of computational language acquisition, Prague, Czech Republic, pp 49–56

Bender EM, Koller A (2020) Climbing towards NLU: On meaning, form, and understanding in the age of data. In: Proceedings of the 58th annual meeting of the Association for Computational Linguistics (ACL 2020), Seattle, United States

Boleda G (2020) Distributional semantics and linguistic theory. Annual Review of Linguistics

Boleda G, Herbelot A (2017) Formal distributional semantics: Introduction to the special issue. Computational Linguistics 42(4):619–635

Burge T (2010) Origins of objectivity. Oxford University Press

Chomsky N (1957) Syntactic structures. The Hague/Paris: Mouton

Chomsky N (1965) Aspects of the theory of syntax. MIT Press

Clark S (2012) Vector space models of lexical meaning. In: Lappin S, Fox C (eds) Handbook of Contemporary Semantics – second edition, Wiley-Blackwell

Copestake A (2007) Semantic composition with (robust) minimal recursion semantics. In: Proceedings of the Workshop on Deep Linguistic Processing, Prague, Czech Republic, pp 73–80

Copestake A, Flickinger D, Sag IA, Pollard C (2005) Minimal Recursion Semantics: an Introduction. Journal of Research on Language and Computation 3(2-3):281–332

Cresswell MJ (1978) Semantic competence. In: Guenthner F, Guenthner-Reutter M (eds) Meaning and Translation, Duckworth, London, pp 9–27

Devlin J, Chang MW, Lee K, Toutanova K (2019) Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, Minneapolis, MN, pp 4171–4186

Dinu G, Lapata M (2010) Measuring distributional similarity in context. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, MA, pp 1162–1172

Emerson G (2018) Functional distributional semantics: Learning linguistically informed representations from a precisely annotated corpus. PhD thesis, University of Cambridge

Erk K (2012) Vector space models of word meaning and phrase meaning: a survey. Language and Linguistics Compass 6:635–653

Erk K (2016) What do you know about an alligator when you know the company it keeps? Semantics and Pragmatics 9:17–1

Erk K, Padó S (2008) A structured vector space model for word meaning in context. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, Honolulu, HI

Flickinger D (2000) On building a more efficient grammar by exploiting types. Natural Language Engineering 6(1):15–28

Fodor JA (1975) The language of thought. Harvard University Press

Geeraerts D (2010) Theories of Lexical Semantics. Oxford University Press, Oxford, England, UK

Goodman ND, Lassiter D (2015) Probabilistic semantics and pragmatics: Uncertainty in language and thought. The handbook of contemporary semantic theory, 2nd edition Wiley-Blackwell

Gupta A, Boleda G, Baroni M, Padó S (2015) Distributional vectors encode referential attributes. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon,Portugal, pp 12–21

Harris Z (1954) Distributional Structure. Word 10(2-3):146–162

Herbelot A (2015) Mr Darcy and Mr Toad, gentlemen: distributional names and their kinds. In: Proceedings of the 11th International Conference on Computational Semantics (IWCS), London, UK, pp 151–161

Herbelot A (2020) Simulating the acquisition of core semantic competences from small data. In: Proceedings of the 24th Conference on Computational Natural Language Learning, online, pp 344–354

Hollander MA, Gelman SA, Star J (2002) Children's interpretation of generic noun phrases. Developmental Psychology 38(6):883

Inhelder B, Piaget J (1964) The early growth of logic in the child: Classification and seriation. Norton, New York

Kabbach A, Gulordava K, Herbelot A (2019) Towards incremental learning of word embeddings using context informativeness. In: Proceedings of the Student Research Workshop at the the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy

Katz JJ, Fodor JA (1963) The structure of a semantic theory. Language 39(2):170–210

Katz JJ, Postal PM (1964) An integrated theory of linguistic descriptions. MIT Press

Kilgarriff A (1997) I don't believe in word senses. Computers and the Humanities 31(2):91–113

Kripke SA (1972) Naming and necessity. In: Semantics of natural language, Springer, pp 253–355

Krishna R, Zhu Y, Groth O, Johnson J, Hata K, Kravitz J, Chen S, Kalantidis Y, Li LJ, Shamma DA, et al. (2017) Visual genome: Connecting language and vision using crowdsourced dense image annotations. International Journal of Computer Vision 123(1):32–73

Kruszewski G, Baroni M (2015) So similar and yet incompatible: Toward the automated identification of semantically compatible words. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics, Denver,CO, pp 964–969

Labov W (1978) Denotational structure. In: Farkas D, Jacobsen W, Todrys K (eds) Parasession on the Lexicon, Chicago Linguistics Society, pp 220–260

Landauer TK, Dumais ST (1997) A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological review pp 211–240

Lenci A (2008) Distributional semantics in linguistic and cognitive research. Italian Journal of Linguistics 20(1):1–31

Lenci A, Benotto G (2012) Identifying hypernyms in distributional semantic spaces. In: Proceedings of the First Joint Conference on Lexical and Computational Semantics, Montreal, Canada, pp 75–79

Link G (1983) The logical analysis of plurals and mass terms: A lattice-theoretical approach. In: Bäuerle R, Schwarze C, Von Stechow A (eds) Meaning Use and Interpretation of Language, Walter de Gruyter, pp 302–323

Linzen T (2020) How can we accelerate progress towards human-like linguistic generalization? In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, online, pp 5210–5217

Lund K, Burgess C, Atchley RA (1995) Semantic and associative priming in high-dimensional semantic space. In: Proceedings of the 17th annual conference of the Cognitive Science Society, Pittsburgh, PA, vol 17, pp 660–665

Matthewson L (2004) On the methodology of semantic fieldwork. International journal of American linguistics 70(4):369–415

Partee BH (1979) Semantics – mathematics or psychology? In: Semantics from different points of view, Springer, pp 1–14

Partee BH (2014) The history of formal semantics: Changing notions of linguistic competence. https://udrive.oit.umass.edu/partee/Partee2014Harvard.pdf, 9th Annual Joshua and Verona Whatmough Lecture, Harvard

Partee BH (2015) The garden of eden period for deep structure and semantics. In: Gallego AJ, Ott D (eds) 50 Years Later: Reflections on Chomsky's Aspects, MIT Press, pp 187–198

Peters M, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, New Orleans, LA, pp 2227–2237

Putnam H (1975) Philosophical Papers: Mind, Language, and Reality, vol 2. Cambridge University Press

Roller S, Erk K, Boleda G (2014) Inclusive yet selective: Supervised distributional hypernymy detection. In: Proceedings of the 25th International Conference on Computational Linguistics, Dublin, Ireland, pp 1025–1036

Soames S (1989) Semantics and semantic competence. Philosophical perspectives 3:575–596

Thater S, Fürstenau H, Pinkal M (2011) Word meaning in context: A simple and effective vector model. In: Proceedings of the 5th International Joint Conference on Natural Language Processing, Chiang Mai, Thailand

Trott S, Torrent T, Chang N, Schneider N (2020) (Re)construing Meaning in NLP. In: Proceedings of the 58th annual meeting of the Association for Computational Linguistics (ACL 2020), online

Venhuizen NJ, Hendriks P, Crocker MW, Brouwer H (2019) A framework for distributional formal semantics. In: International Workshop on Logic, Language, Information, and Computation, Springer, pp 633–646