Linguistics **I**ssues in **L**anguage **T**echnology – LiLT

Perspectives on Semantic Representations for Textual Inference

Cleo Condoravdi, Valeria de Paiva and Annie Zaenen (eds)

CENTER FOR THE STUDY OF LANGUAGE AND INFORMATION

LiLT volume 9, issue 6 February 2014

Frege in Space: A Program for Compositional Distributional Semantics

Marco Baroni[1], Raffaella Bernardi[2] and Roberto Zamparelli[3]

To Emanuele Pianta,
in memoriam

Abstract

The lexicon of any natural language encodes a huge number of distinct word meanings. Just to understand this article, you will need to know what thousands of words mean. The space of possible sentential meanings is infinite: In this article alone, you will encounter many sentences that express ideas you have never heard before, we hope. Statistical semantics has addressed the issue of the vastness of word meaning by proposing methods to harvest meaning automatically from large collections of text (corpora). Formal semantics in the Fregean tradition has developed methods to account for the infinity of sentential meaning based on the crucial insight of *compositionality*, the idea that meaning of sentences is built incrementally by combining the meanings of their

[1]Center for Mind/Brain Sciences and Department of Information Engineering and Computer Science, University of Trento (IT)
[2]Department of Psychology and Cognitive Science, University of Trento (IT)
[3]Department of Psychology and Cognitive Science, University of Trento(IT)

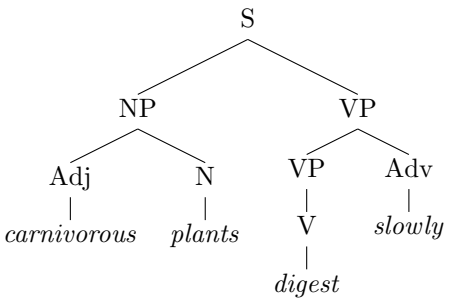
LiLT Volume 9, Issue 6, February 2014.
Perspectives on Semantic Representations for Textual Inference.

constituents. This article sketches a new approach to semantics that brings together ideas from statistical and formal semantics to account, in parallel, for the richness of lexical meaning and the combinatorial power of sentential semantics. We adopt, in particular, the idea that word meaning can be approximated by the patterns of co-occurrence of words in corpora from statistical semantics, and the idea that compositionality can be captured in terms of a syntax-driven calculus of function application from formal semantics.

1 Introduction

Semantic compositionality is the crucial property of natural language according to which the meaning of a complex expression is a function of the meaning of its constituent parts and of the mode of their combination. Compositionality makes the meaning of a sentence such as *"carnivorous plants digest slowly"*, in (1), a function of the meaning of the noun phrase *"carnivorous plants"* combined as a subject with the meaning of the verb phrase *"digest slowly"*, which is in turn derived by the combination of the meaning of the verb *digest* and a modifier, the adverb *slowly*.

(1)



Together with the property of syntactic recursivity, which grants humans the possibility of constructing indefinitely long grammatical sentences, semantic compositionality allows us to compose arbitrarily complex meanings into sentences, and extract meanings from arbitrarily long and complex sentences, starting from a finite lexicon.

While the compositional nature of human language has been in some sense acknowledged since Aristotle's subject-predicate distinction, it has been brought to the foreground only in modern times, mainly with the work of the German logician Gottlob Frege (hence the alternative name *Frege's Principle*, see especially Frege, 1892).[4] Compositionality

[4]As Emilano Guevara and Dominic Widdows (p.c.) point out, surprisingly, the principle was never explicitly stated by Frege (Pelletier, 2001), and it was arguably

was later operationalized by Richard Montague (see in particular Montague, 1970b, 1973). Together with the assumption that meaning can be accounted for *denotationally* (that is in terms of whether a linguistic utterance is truthful with respect to a *model* of the world) compositionality has since informed the whole area of theoretical linguistics known as *natural language semantics* or *formal semantics* (see Partee, 2004, and Werning et al., 2012 for a broad overview of compositionality covering the application of the principle also in general cognition and computer science).[5]

Compositionality is well-known not to apply in expressions such as *"red herring"* or *"kick the bucket"*, whose dominant, idiomatic meanings have nothing to do with fish or pail-hitting. However, even in these cases a 'literal', compositional route cannot be blocked (i.e., these expressions could be used, and understood, as descriptions of a real herring dyed red, the kicking of a real bucket, etc.). This suggests that the compositional mode of meaning construction is a primitive in human language, and a crucial ingredient in any theory that tries to model the way humans use language.

Consider again the schematic binary tree structure in (1). Most (though not all) theories of sentential semantics implement compositionality by building meanings bottom-up. Moving from the lexicon, (here *digest, carnivorous, plants* and *slowly*), which can be retrieved from memory, they follow the syntactic tree up to the main clause (S), combining pairs of sister nodes (both lexical, like [*carnivorous plants*] and non-lexical, like [NP VP]) by means of a small set of primitive operations, possibly just one, *function application*. In function application, one of two sister nodes is treated as a function and applied to the other to return the meaning of their mother node, which can in turn be an argument or a function in further combinations (see Heim and Kratzer, 1998, for a thorough introduction to the function application approach to composition).

This general line of research, split into many different strands and flavours which will not concern us here, has enjoyed a great success in terms of constructions covered and depth of the explanation, generating thorough descriptions of the behaviour of quantifiers and articles, long-distance dependencies, coordination and comparatives, and many other individual phenomena most often linked to the lexicon of *grammatical*

already assumed by Boole (1854) decades before Frege's work. We will stick to the traditional name despite its historical inaccuracy.

[5]Following standard practice, in this article we sometimes use the term *denotational semantics* synonymously with *formal semantics*, especially when we want to emphasize the denotational side of the latter.

elements, such as determiners and conjunctions. However, the success of a scientific enterprise can and ultimately must be measured outside of a lab. For a semantic theory, which has few products to send out to the market, this means the ability to give understandable descriptions of the whole semantic behavior of an arbitrary sentence, which can easily contain dozens of intertwined semantic phenomena. In contrast, most semantic studies have dealt with individual constructions, and have been carried out under highly simplifying assumptions, in true lab conditions. If these idealizations are removed it is not clear at all that modern semantics can give a full account of all but the simplest sentences.

Consider for instance the problem of lexical ambiguity: Nearly all papers in formal semantics start from the assumption that all the words in the examples under study have been neatly disambiguated before being combined. But this is not an innocent or easy assumption. To evaluate the truth or falsity of (2) in a model, one would have to assume that *paper* is not the material, but the text in a certain page format (and not in a purely abstract sense, as in *"his paper has been reprinted in ebook format"*); that *runs* means something like *extends*, and not the *running* of horses, the *running* of cars or what Obama did as he *"ran for presidency"* (despite the presence of *for*). In turn, *for* does not indicate a goal (*"I ran for the exit"*), nor a benefactive (one could say: *"in the Olympics, the British team ran for their Queen"*, but nobody would run for a simple *page*, not even for 110 of them).

(2) This paper runs for 110 pages.

Often, choosing to combine the 'wrong' meanings does not lead to falsity, but to non-sensicality. It is not easy, however, to figure out which possibilities are odd without trying the semantic combinations, and many will remain uncertain. With more complex sentences, such as (3), a wealth of additional complexities emerge: You have to understand that in this context *analysts* are technology analysts, that *every* actually ranges over a subpart of them, that *paid* has a special meaning due to the presence of *attention* before it, that *attention* here means 'amount of attention', that *unbelievable* actually means 'hard to believe', etc.

(3) Every analyst knows that the attention Jobs paid to details was simply unbelievable.

Even in the limited domain of modification, the variability of meaning that emerges has long been recognized, but its consequences not always

fully appreciated. As Lahav remarks:

> In order for a cow to be brown most of its body's surface should be brown, though not its udders, eyes, or internal organs. A brown crystal, on the other hand, needs to be brown both inside and outside. A book is brown if its cover, but not necessarily its inner pages, are mostly brown, while a newspaper is brown only if all its pages are brown. For a potato to be brown it needs to be brown only outside. . . Furthermore, in order for a cow or a bird to be brown the brown color should be the animal's natural color, since it is regarded as being 'really' brown even if it is painted white all over. A table, on the other hand, is brown even if it is only painted brown and its 'natural' color underneath the paint is, say, yellow. But while a table or a bird are not brown if covered with brown sugar, a cookie is. In short, what is to be brown is different for different types of objects. To be sure, brown objects do have something in common: a salient part that is wholly brownish. But this hardly suffices for an object to count as brown. A significant component of the applicability condition of the predicate 'brown' varies from one linguistic context to another.
>
> (Lahav 1993:76)

What happens with *brown* is replicated in the large majority of adjective-noun combinations. Treating them all like 'idioms' would turn the exception into the rule. After all, there must be many regularities in their combinatorial behaviour, or children would not be able to learn modification correctly, and indeed semanticists have long recognized that many cases of context-driven polysemy are systematic (on the notion of *regular polysemy* see, e.g., Apresjan, 1974, Pustejovsky, 1995).

Add to this that the meaning of abstract terms has only begun to be investigated by formal semantics (what is the denotational meaning of *numerosity* or *bravery*?), and the outlook for a semantic analysis that can span real-life sentences and not just sterilized constructions starts to look not particularly promising.

As is easy to see, many of the problems come from the lexicon of *content words*, such as nouns, verbs and adjectives, and not from grammatical terms. Content words constitute the area of the lexicon with the greatest amount of items (by the end of high-school, an average Western person might know the meaning of as many as 60,000 content words, see Aitchison, 1993), characterized by a lot of idiosyncratic meaning combinations. Of course, there have been important attempts to tackle the lexicon problem from the point of view of formal semantics, like Pustejovsky's (1995) theory of the generative lexicon. More recently, Asher (2011) has approached lexical semantics with a theory of predication that uses a sophisticated system of semantic types, plus a mechanism of

type coercion. This gives an interesting account of difficult phenomena that relate to the ambiguity and context-dependent nature of content words, such as *co-predication*, exemplified in (4), (where *lunch* is interpreted as referring to the food in one conjunct and to the event of eating in the other) or *predicate coercion*, where a predicate is provided to 'fix' a type mismatch, as in (5).

(4) Lunch was delicious but took forever.

(5) John enjoyed a beer. *he enjoyed DRINKING it*

However, the problem of lexical semantics is primarily a problem of *size*: even considering the many subregularities found in the content lexicon, a hand-by-hand analysis is simply not feasible for the thousands of elements that populate the content word lexicon. For many of these elements, we would have to manually specify how they are affected by the context-dependent process of polysemy resolution: from regular polysemy to non-systematic meaning shifts, down to the *co-composition* phenomena illustrated by the citation above (see Section 2.3).Without such an analysis, the goal of a semantic treatment for actual *sentences* rather than abstract *constructions* remains out of reach.

Similar problems are familiar elsewhere in linguistics. The problem of assigning reasonable (if not exhaustive) syntactic structures to arbitrary, real-life sentences is perhaps equally hard. Here, however, technology has provided an important part of the answer: Natural language parsers, which automatically assign a syntactic structure to sentences, have made great advances in recent years by exploiting probabilistic information about parts of speech (POS tags) and syntactic attachment preferences. This in turn has been made possible by the availability of medium-sized corpora annotated for POS and syntactic information, such as the Penn Treebank (Marcus et al., 1993), that serve as the basis for extracting probabilistic information from. Today's state-of-the-art parsers can process dozens of unannotated, possibly noisy real-life sentences per second (Clark and Curran, 2007, Nivre, 2003).[6]

Learning from pre-annotated data has been less directly applicable to the goal of providing a semantic representation for sentences because there are few learning samples marked for meaning (but see Basile et al., 2012). Moreover, the range, variety and often 'fuzzy' nature of

[6]The error margin remains high. Its bounds are given by the accuracy of the structures the parser has learned from. Better structures in the learning sample should lead to better parsing across the whole corpus to be parsed, and it is possible that in the future incorporating DS measures in parsing preferences might lead to better results, perhaps to the point of modeling human garden-path effects. See Manning (2011) for similar considerations with respect to part-of-speech tagging.

semantic phenomena makes the prospect of manual semantic markup of text data a lot less appealing than for syntax. As a consequence, data-driven semantics—which would in principle be a way to address the vastness of lexical meanings—has not advanced as rapidly as data-driven syntax.

What sort of data-driven methods could truly help semantics? If the main problem for the semantics of sentences is the content lexicon, we should try to find methods that use vast corpora to extract the meaning of content words and represent them in appropriate ways.[7] But these meaning representations should be objects that compose together to form more complex meanings, while accounting for how composition causes more or less sytematic shifts in word meaning, as in the co-composition, co-predication and coercion examples above. Moreover, the meaning of content words as we can extract them from a corpus should be able to combine with the meaning of *grammatical* words, formal semantics' special focus, in ways that account for the importance of structure in sentence meaning, and which can shed light also on the linguistic phenomena that interest the theoretical linguist. This is what this paper is about. We propose it as a research program for linguistics, both theoretical and computational, but also as a description of the thriving subfield of *compositional distributional semantics*.

As we shall see in the next sections, this field takes a stand which might be counterintuitive for the formal linguist—that the meaning of content words lies in their distribution over large spans of text. We believe that this aspect of meaning is very real and concrete, and that it complements a denotational approach in which words 'mean' the objects they stand for.

In the remainder of this article, we make our case as follows. Section 2 is a concise introduction to distributional semantics, focusing on its implications for linguistics and cognitive science. Section 3 presents *compositional* distributional semantics, and in particular the Fregean approach to the challenge of composing distributional representations we endorse. Section 4 contains a brief review of empirical evidence in favour of our specific proposal, while Section 5 offers more general moti-

[7] An alternative approach is to rely on *lexical resources* that contain rich semantic information about content words (e.g., Baker et al., 1998, Fellbaum, 1998, Kipper et al., 2008). We find the corpus route more appealing because it is not *a priori* limited by the amount of manually coded data entered in a resource. Many fuzzy aspects of word meaning are arguably better captured by the distributional representations we are about to introduce than by the hand-coded symbolic formalisms encoded in lexical resources. Moreover, we hope that the very process of *inducing* meaning from naturally occurring data will be very instructive about what meaning really is, and possibly about how we humans come to possess it.

vations for the need of a theory of compositional distributional semantics. The emphasis is not on applications, but on what this approach can bring to a theory of linguistic meaning. Section 6 briefly reviews some related work that has not been discussed in the previous sections, or to which we felt the need to return given how closely connected it is to our. The heroic reader who made it to the end of this unusually long paper will discover, in our valedictory in Section 7, that we left so many important issues unaddressed that we are already working on Part 2.

2 Distributional Semantics

The *distributional hypothesis* states that words occurring in similar (linguistic) contexts are semantically similar. This idea has its theoretical roots in various traditions, including American structuralist linguistics, British lexicology and certain schools of psychology and philosophy (Firth, 1957, Harris, 1954, Miller and Charles, 1991, Wittgenstein, 1953). It had a huge impact on computational linguistics mainly because it suggests a practical way to automatically harvest word "meanings" on a large scale: If we can equate meaning with context, we can simply record the contexts in which a word occurs in a collection of texts (a *corpus*) to create a summary of the distributional history of the word that can then be used as a surrogate of its semantic representation.

While nearly all corpus-based approaches to computational semantics exploit distributional information in one way or another, we focus here on *Distributional Semantic Models* (DSMs), that are the most direct realization of the distributional hypothesis in computational linguistics (Clark, 2013b, Erk, 2012, Landauer and Dumais, 1997, Lund and Burgess, 1996, Sahlgren, 2006, Schütze, 1997, Turney and Pantel, 2010). In a DSM, each word is represented by a mathematical *vector*, that is, an ordered list of numbers. The values in the vector *components* are a function of the number of times that the words occur in the proximity of various linguistic contexts in a corpus. As a toy example, suppose that our target vocabulary contains the nouns *dogs*, *hyena* and *cat* and our contexts are the words *barks* and *runs*. We traverse the corpus and find out that *dog* occurs in proximity of *runs* 1 time and near *barks* 5 times. We can thus represent *dog* with the *distributional vector* that constitutes the first column of Table 1. Similarly, *hyena* and *cat* are represented by the next two columns in the table, reflecting how many times they co-occur with *runs* and *barks* in the corpus.

Intuitively, based on this evidence we can deduce that *dog* is more

	dog	**hyena**	**cat**
runs	1	1	4
barks	5	2	0

TABLE 1 Distributional vectors representing the words *dog*, *hyena* and *cat*.

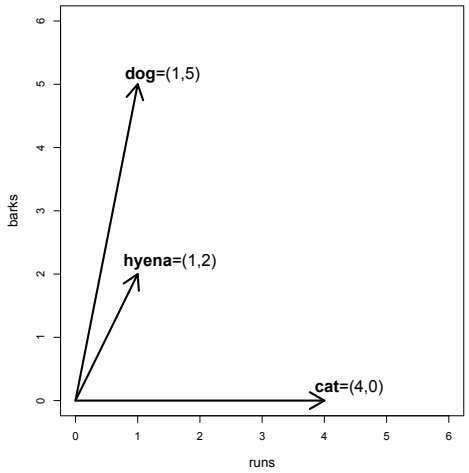


FIGURE 1 Geometric representation of the vectors in Table 1.

similar to *hyena* than to *cat* because they both occur one time with *runs* and multiple times with *barks*, whereas *cat* occurs more frequently with *runs* and never with *barks*. Distributional vectors allow a precise quantification of similarity deriving from their representation as geometric objects. In Figure 1, the vectors are represented as oriented segments ("arrows") running from the origin of a Cartesian plane to x and y coordinates corresponding to the values in their first and second components (e.g., the endpoint of the **dog** vector has coordinates $x = 1$ and $y = 5$).[8]

In this geometric view, the similarity of the contexts in which words occur (and thus, according to the distributional hypothesis, their se-

[8]Following common practice, we use boldface lowercase letters, e.g., $\mathbf{a}$, to represent vectors, boldface capital letters, e.g., $\mathbf{A}$, to represent matrices and Euler script letters, e.g., $\mathcal{X}$, to represent higher-order tensors (matrices and tensors are introduced in Section 3.3 below). However, when we want to emphasize the linguistic interpretation of a mathematical structure, we might denote it by its linguistic label in italics: Depending on the context we might refer to the vector representing the word *dog* either as **dog** or as *dog*.

mantic similarity) is measured by the distance of the corresponding vectors on the Cartesian plane. In particular, DSMs typically use the *cosine* of the angle formed by two vectors as a measure of semantic similarity. The cosine is a function of the width of the angle, and ranges from 1 for parallel vectors to 0 for perpendicular (or orthogonal) vectors.[9] In the running example, **dog** has a cosine of 0.96 with **hyena** (the angle between the corresponding vectors is very narrow), and a much lower cosine of 0.2 with **cat** (wider angle).

We refer to the number of components of a vector as the *size* of the vector. The vectors in our toy example have size 2. Of course, real DSMs encode many more contexts, and consequently work with vectors of much larger sizes (ranging from hundreds to millions of components). The same geometric properties we can easily visualize on the Cartesian plane for the size-2 case (such as angular width) generalize to vectors of arbitrary size.

Mathematicians refer to the set of all possible vectors of size N as the *N-dimensional vector space*. Hence, sets of distributional vectors are often said to inhabit a "semantic" or "distributional" space. However, when we interpret the values in the components of a distributional vector as (function of) co-occurrences with contexts, the same N-dimensional vector space will have different interpretations depending on the labels (linguistic contents) attached to the components. For example, in the running example we associated the first component to *runs* and the second to *barks*. If we associated the two components to *red* and *banana*, respectively, mathematically we would still be operating in the same mathematical vector space (the 2-dimensional vector space), but the vectors we would be obtaining would represent different linguistic meanings. Thus, in what follows, whenever we talk of a vector space, we implicitly refer to the set of all possible vectors of a fixed size *whose components are associated, in the same order, to the same set of linguistic contents*. Under this definition, there are many (possibly infinite) distinct vector spaces of dimensionality N.

In the next subsections, we briefly survey the main steps necessary to build a DSM (Section 2.1), we review how DSMs have been used in practice (Section 2.2), and then turn to some theoretical issues pertaining to them (Sections from 2.3 to 2.6). More thorough recent introductions to DSMs are provided by Clark (2013b), Erk (2012) and Turney and Pantel (2010).

[9] If the vectors contain components with negative values, the cosine can also be negative, with the minimum value of -1 for parallel vectors pointing in opposite directions. In distributional semantics, negative values may arise when counts are transformed into other kinds of scores; see Section 2.1 below.

2.1 Parameters of DSMs

Most research on DSMs focuses on the many parameters of the pipeline to extract distributional vectors from corpora.[10] Surprisingly, there is relatively little research on how the nature of the source corpus affects the quality of the resulting vectors, but, as in many other areas of computational linguistics, the general consensus is that "more data is better data" (Curran and Moens, 2002b). The most popular data source is the British National Corpus,[11] a 100 million word corpus attempting to provide a "balanced" sample of various registers and genres of both written and spoken English. More recently, larger corpora (in the order of a few billion words), often made up of Web documents (including Wikipedia pages), are also widely used.[12]

Probably the most important decision when developing a DSM pertains to defining what is a *context* for purposes of counting co-occurrences. Definitions of context range from simple ones (such as documents or the occurrence of another word inside a fixed window from the target word) to more linguistically sophisticated ones (such as the occurrence of words of certain syntactic categories connected to the target by special syntactic relations) (Curran and Moens, 2002a, Grefenstette, 1994, Padó and Lapata, 2007, Sahlgren, 2006, Turney and Pantel, 2010). Different contexts capture different kinds of semantic similarity or "relatedness" (Budanitsky and Hirst, 2006). At the two extremes, counting documents as contexts captures "topical" relations (the words *war* and *Afghanistan* will have a high cosine, because they often co-occur in documents), whereas DSMs based on word co-occurrence within narrow windows or constrained by syntactic relations tend to capture tighter "taxonomic" relations (such as the one between *dog* and *hyena*). Unsurprisingly, no single definition of context is appropriate for all tasks, and the jury on the "best" context model is still out (Sahlgren, 2008).

Next, raw target-context counts are typically transformed into *association scores* that discount the weights of components associated to contexts with high probability of chance occurrence (Evert, 2005). For example, co-occurring with a relatively rare word such as *barks* is enormously more informative about the meaning of *dog* than co-occurring

[10] Bullinaria and Levy (2007, 2012) provide a sysstematic evaluation of how some of the pipeline parameters affect DSM quality.

[11] `http://www.natcorp.ox.ac.uk/`

[12] See for example `http://wacky.sslmit.unibo.it/`. A potential problem with Web corpora is their systematic skewness, as in the probable overassociation of *page* with *home*. This can presumably be addressed with better sampling and filtering techniques (see Fletcher, 2004, 2012).

with *the*, despite the fact that, in any corpus, *dog* will occur many more times with the latter than with the former. An association measure such as Pointwise Mutual Information or Log-Likelihood Ratio will increase the value in the *barks* component, dampening the one in the *the* component.

Optionally, the collection of vectors of association scores produced in the previous step globally undergoes *dimensionality reduction*, after which the same target words are represented in a lower-dimensionality space whose components (deriving from the original ones via a statistical process that considers their correlation patterns) should capture more robust "latent" aspects of meaning (Blei et al., 2003, Dinu and Lapata, 2010, Griffiths et al., 2007, Landauer and Dumais, 1997, Sahlgren, 2005, Schütze, 1997).

Although it is not strictly a parameter in the construction of DSMs, researchers measure distributional (and thus semantic) similarity of pairs of target words with different *similarity functions*. The already introduced cosine of the angle formed by vectors is the most natural, geometrically justified and widely used of these functions.[13]

We conclude this short survey of DSM engineering by observing that, while in our discussion below we assume that the components of a semantic space can be interpreted as a distribution over contexts in which words tend to occur, in real DSMs, after the transformation into association scores and dimensionality reduction, the relationship between target words and contexts is often rather indirect.

[13] A natural alternative to cosines is the Euclidean distance between vectors, that is, the length of the segment that connects their endpoints. An important property of Euclidean distance is that it is sensitive to vector length. *Hyena* is a less frequent word than *dog*. Consequently, it occurs less often in the contexts of interest and its distributional vector is geometrically shorter. Thus, in Figure 1 the endpoints of the **dog** and **hyena** vectors are relatively distant, while the width of the angle between the vectors is not affected by length. If we kept collecting *dog* data without finding any further *hyena* occurrences in the corpus, as long as *dog* maintained the same rate of occurrence with *runs* and *barks*, the angle (and consequently the cosine) between the two vectors would not be affected, while Euclidean distance would keep growing. The cosine can thus be seen as a more robust similarity measure than Euclidean distance. This is not to say that vector *length* by itself is not an informative quantity. For example, since it is a function of how frequently the word represented by the vector was encountered in the corpus (modulo possible statistical transformations of the input values), it is a measure of the reliability of the distributional evidence encoded in the vector. Finally, note that Euclidean distance and cosine are in a bijective functional relation if Euclidean distance is computed on vectors that have been normalized to length 1.

2.2 Applications and cognitive simulations

The large-scale semantic representations provided by DSMs can profitably be embedded in applications that require a representation of word meaning, and in particular an objective measure of meaning similarity. Such applications range from document retrieval and classification to question answering, automated thesaurus construction and machine translation (Dumais, 2003, Turney and Pantel, 2010). DSMs are also very effective in simulating psychological and linguistic phenomena related to word meaning, such as predicting similarity judgments and semantic priming, categorizing nominal concepts into hypernyms, generating salient properties of concepts (and qualia of nouns), capturing intuitions about the thematic fit of verb arguments and even spotting the alternation classes of verbs (Baroni and Lenci, 2010, Baroni et al., 2010, Landauer and Dumais, 1997, Lenci, 2011, Lund and Burgess, 1996, McDonald and Brew, 2004, Padó and Lapata, 2007, Padó et al., 2007, and references there).

For example, starting with the classic work of Landauer and Dumais (1997), researchers have shown that cosines in distributional space predict which word, among a set of candidates, is the synonym of a target item (e.g., DSMs pick *pinnacle* as synonym of *zenith* over the foils *completion, outset* and *decline*). DSM performance on this task approximates that of native English speakers with a college education (Rapp, 2004). Padó and Lapata (2007) and others have shown how the cosines between vectors of word pairs can predict whether the corresponding words will "prime" each other or not (that is, whether a subject will recognize the second word faster when the first one has just been presented). Kotlerman et al. (2010) and others use DSMs to predict lexical entailment (discovering whether the concept denoted by one word implies the one denoted by another; for example, *dog* entails *animal*). Padó et al. (2007) show that (simplifying somewhat) the cosine between a vector representing the typical subject or object of a verb and a vector representing an arbitrary noun correlates with human intuitions about the plausibility of the noun as subject or object of the verb. For example, the *monster* vector is closer to the average "subject-of-*frighten*" vector than to the corresponding object vector, reflecting subject intuitions that monsters are more likely to be frighteners than frightees.

2.3 Polysemy and word meaning in context

We have suggested in the introduction that traditional formal semantics might not be the right approach to capture the rich polysemous patterns of (mainly) content words. On the face of it, standard DSMs do

not address the issue of polysemy, since they represent each word with a single distributional vector. In the common case in which a word has more than one facet of meaning (ranging from full-fledged instances of homonymy such as *river bank* vs. *central bank* to subtler alternations such as *chicken in the farm* vs. *chicken in the oven* or the cases of co-predication and coercion discussed in the introduction), the distributional vector will be a summary of these facets. There has however been a lot of work on handling polysemy in DSMs (e.g., Boleda et al., 2012a, Erk, 2010, Pantel and Lin, 2002, Schütze, 1998) showing that these models are actually very well-suited to capture various kinds of polysemy on a large scale. Note that polysemy is naturally modeled in terms of the contexts in which a word appears: In a sentence containing words such as *farm*, *free-range* and *outdoors*, the word *chicken* is more likely to mean the animal than its meat (albeit an animal with a clear culinary destiny). Consequently, a large subset of work on polysemy in DSMs (e.g., Dinu and Lapata, 2010, Erk and Padó, 2008, 2010, Kintsch, 2001, Mitchell and Lapata, 2008, Reisinger and Mooney, 2010, Thater et al., 2009) has focused on the goal of modeling word meaning in context.

There is a clear connection between distributional models of word meaning in context and distributional models of compositionality, which is the main topic of this article. For example, Mitchell and Lapata (2008) discriminate between the senses of *running* in *water runs* vs. *horse runs* by composing vectors representing the two phrases, whereas Erk and Padó (2008) and others approach the same task in terms of how the *runs* vector changes due to contextual effects triggered by the presence of *water* vs. *horse*. Mitchell and Lapata construct a vector for *water runs*, Erk and Padó for *runs-in-the-context-of-water*, so to speak (both the task and the latter approach are reminiscent of Pustejovsky's 1995 notion of co-composition, whereby the meaning of a verb is affected by the arguments it composes with). In our research, we follow Mitchell and Lapata and focus on more general compositional methods, hoping to develop composition operations that are flexible enough to also capture word-meaning-in-context effects, and consequently handle polysemy correctly. In Section 3.4, we will see why the specific framework for composition we are proposing might be well-suited to handle context-driven meaning changes, and in Section 4.2 we will present evidence that we can successfully capture the sort of co-compositional effects briefly sketched above.

There are cases where immediate (co-)composition does not suffice for disambiguation, and only the wider context makes the difference. However, disambiguation based on direct syntactic composition takes

precedence; in a sentence such as *"The fisherman saw from his boat that the bank near the river docks was being robbed"*, the verb *robbed* overrides all the contextual features pointing to the *river* sense of bank. But in cases like *"The heavy rains made the river grow wild. The bridge fell and the banks collapsed"*, no current model of syntax-driven composition could disambiguate *banks*, since financial institutions are prone to collapsing just as easily as river sides. In the long term, discourse composition models will probably have their say on these cases. Note, finally, that the problem of representing homonymy in terms of DSMs will ultimately require a way to handle *disjunctive* meanings. We will return to this issue in Section 7.

2.4 DSMs as linguistically and psychologically viable feature-based theories of word meaning

The process to build DSMs relies on the standard toolbox of computational linguists (corpus parsing, statistical measures of association, etc.), but the resulting set of distributional vectors is not a tool in and of itself. It is not immediately obvious, for instance, how to use it to improve the performance of syntactic parsers, semantic role labelers or sentiment polarity detectors. A DSM should rather be viewed as a portion of the lexicon containing semantic representations for (a considerable proportion of) the words in a language. Unlike definitions or subject-derived labels, these representations are intrinsecally graded, thus ameanable to be processed by a range of mathematical techniques, depending on the specific task we want to use them for. Baroni and Lenci (2010), in particular, have shown how the very same set of distributional vectors can be exploited for a large variety of different lexical semantic tasks. A DSM, in this sense of a set of distributional vectors for the words of a language, is rather more like a concrete implementation of a feature-based theory of semantic representation, akin to the Generative Lexicon (Pustejovsky, 1995) or Lexical Conceptual Structures (Jackendoff, 1990). Unlike these theories, however, DSMs can be induced on a large scale from corpus data, which makes them both attractive from the acquisitional point of view and amenable to systematic evaluation on realistically sized data sets. DSMs, moreover, use hundreds of features (that is, vector components), and assign them (automatically induced) real-valued scores.[14]

[14]This latter property makes them particularly well-suited to capture prototypicality effects (*penguins* are less "birdy" than *robins*) and more in general all the fuzzy, continuous aspects of lexical meaning that are intensively investigated by psychologists (Murphy, 2002) but problematic for formal semantics approaches to the lexicon (Kamp and Partee, 1995).

A cautious view of DSMs is that they are a handy engineering surrogate of a semantic lexicon. Various considerations support, however, the bolder stance that DSMs *are* models of a significant part of meaning as it is encoded in human linguistic competence (see Lenci, 2008, for related conjectures on the status of DSMs).[15]

First, these models are successful at simulating many aspects of human semantic performance, as briefly reviewed in Section 2.2 above.

Second, they achieve this performance using only large amounts of naturally occurring linguistic data as learning input, a kind of input that human learners also receive in generous doses (although, admittedly, text corpora are very different sorts of linguistic data from those that children are exposed to when acquiring their native tongue, in terms of amount of data, order and presence of an associated external context).

Third, even those language acquisition theorists who stress the role of extra-linguistic cues (e.g., Bloom, 2000) recognize that the vocabulary size that teenagers command by end of high-school (in the order of tens of thousands of words) can only be acquired by bootstrapping from linguistic data.[16] This bootstrapping is likely to take the form of distributional learning: We all have the experience of inferring the meaning of an unknown term encountered in a novel just from the context in which it occurs,[17] and there is psycholinguistic evidence that statistical patterns of co-occurrence influence subjects' intuitions about the meaning of nonce words just as they do in DSMs (McDonald and Ramscar, 2001).[18]

Fourth and last, in neuroscience there is strong support for the view that concepts are represented in the brain as patterns of neural activa-

[15]We do not claim that distributional vectors are the *only* kind of semantic representations that people store in their heads, just one of possibly many aspects of meaning that might be stored in the semantic lexicon.

[16]Bloom and others emphasize the role of interaction and attention in language acquisition. Rather than providing extra cues to meaning, however, these cognitive functions help learners to focus on the most informative portions of the input stream. As such, they are compatible with distributional (as well as other forms of) learning.

[17]Humans often only require a single exposure to a word in context to learn its meaning, a phenomenon known as "fast mapping" (Carey and Bartlett, 1978). We are not aware of studies systematically evaluating the quality of distributional vectors extracted from single occurrences of words (although there is no doubt that the distributional representation of words generally improves with more contexts).

[18]Chung-chieh Shan (personal communication) remarks that demonstrating that humans rely on contexts to learn meaning is not the same as demonstrating that meaning *is* given by a summary of these contexts, as they are embedded in distributional vectors. However, as Shan suggested to us, Occam's razor favors a semantic representation that is close to the distributional cues it derives from, until the latter is proven wrong empirically.

tion over broad areas (Haxby et al., 2001), and vectors are a natural way to encode such patterns (Huth et al., 2012); this suggests intriguing similarities between neural and distributional representations of meaning. Indeed, recent work in brain-computer interaction (Mitchell et al., 2008, Murphy et al., 2012) has shown that corpus-based distributional vectors are good predictors of the patterns of brain activation recorded in subjects thinking of a concept. This suggests, albeit in a very speculative way, that there might be a direct link between the distributional information encoded in DSMs and the way in which concepts are evoked in the brain.

Of course, we do not want to suggest that DSMs are models of meaning *acquisition*. To name just one crucial difference, the sequence of texts used to form a context vector for a certain word is essentially random (corpora are not internally ordered), whereas the linguistic input that a child receives follows a fairly predictable progression both in form (cf. the "motherese" hypothesis; see for example Newport et al., 1977) and domains of conversation, which are also likely to affect the statistical properties of word associations (Hills, 2013). However, we do endorse the view that distributional semantics *is* a theory of semantics, and that DSMs are an important part of the semantic component of an adult speaker's mental lexicon. In short, the claim is that a core aspect of the meaning of a word is given by (a function of) its distribution over the linguistic contexts (and possibly the non-linguistic ones, see next subsection) in which it occurs, encoded in a vector of real values that constitutes a feature-based semantic representation of the word.

2.5 The symbol grounding problem

Since DSMs represent the meaning of a symbol (a word) in terms of a set of other symbols (the words or other linguistic contexts it co-occurs with), they are subject to the lack-of-grounding criticism traditionally vented against symbolic models (Harnad, 1990; from a philosophical perspective, the obvious reference is to the Chinese Room thought experiment of Searle, 1980). If symbols are not grounded in the sensory-motor system and thus connected to the external world, they cannot really have "meaning". A good DSM might know about the linguistic contexts in which the word *daisy* occurs so well that it can fake human-like intuitions about which other words are most similar, or accurately predict which sentences could contain the word *daisy*. Still, since the DSM has never seen a daisy and so it has never experienced its color, its shape, etc., we might be reluctant to admit that the model truly "knows" the meaning of the word *daisy* (references for the grounding debate in relation to DSMs include Andrews et al., 2009, Burgess, 2000,

Glenberg and Robertson, 2000, Louwerse, 2011, Riordan and Jones, 2011). It is indeed quite telling that DSMs have been applied with some success to the Voynich Manuscript, a 15^{th} century text written in an unreadable script (Reddy and Knight, 2011)—a case of 'semantic analysis' on a document of unknown content.

We believe that the current limitations of DSMs to *linguistic* contexts are more practical than theoretical. Indeed, by exploiting recent advances in image analysis, a new generation of DSMs integrates text data with visual features automatically extracted from pictures that co-occur with the target words, to attain a more perceptually grounded view of distributional word meaning (Bruni et al., 2011, 2012, Feng and Lapata, 2010, Leong and Mihalcea, 2011, Silberer and Lapata, 2012). With research continuing in this direction, DSMs might be the first symbolic semantic models (or even more generally the first fully implemented large-scale computational semantic models) to truly address the symbol grounding problem.

2.6 Meaning and reference

The symbol grounding challenge raised by philosophers and cognitive scientists pertains to the perceptual underpinnings of our generic knowledge of concepts (you need to have seen a dog to truly grasp the meaning of the word *dog*). The dominant tradition in formal semantics stresses instead another type of relation between linguistic signs and the external world, characterizing meaning in terms of reference to a specific state of the world, above and beyond our ability to perceive it. Knowing the meaning of the statement *Marco is a dog* is knowing under which outside-world conditions this statement would be true. To calculate this, standard *denotational semantics* takes as its primitives individuals (objects or events), truth values and propositions (possible worlds, states of affairs).

The focus of denotational semantics and DSMs is very different, and so are their strengths and weaknesses. In denotational semantics, proper names are the simple cases, those that directly point to individuals, unary predicates refer to sets of individuals having a certain property at a given time and world, binary ones refer to sets of ordered pairs of individuals, and so forth. In turn, quantifiers express relations between sets, modifiers typically reduce the size of sets, predicate conjunction intersects them, etc. (see, e.g., Heim and Kratzer, 1998). This model of meaning has been designed to express *episodic knowledge*—facts that are true of specific individuals at specific places and times. Capturing the meaning of *generic sentences*—statements about laws, regularities or tendencies of whole classes of objects—requires a com-

plex quantificational apparatus (Krifka et al., 1995, Cohen, 2004) and is a widely debated but still ill-understood topic.

DSMs, on the other hand, are extracted from large corpora where proper names, common nouns and other predicates refer to states of the world and events spanning a large chunk of time and space, reflecting different points of view, etc. So, if they are able to extract any factual information at all, this is very likely to take the form of generic knowledge. Indeed, a typical application of corpus-based semantics is the extraction of commonsense-knowledge "factoids" that are generally useful while not universally true: bananas are yellow, birds fly, etc. (e.g., Eslick, 2006, Schubert and Tong, 2003). Statistical notions are simply not native to the formal semantics setup, but they are at the heart of the DSM approach. We will return to the issue of generic knowledge in Section 5.2 below. Two things should be noted here.

First, it is perfectly possible to give a reference-based interpretation of DSMs, the question is what it tells us.[19] Suppose we characterize the meaning of a word w in the corpus C in terms of its sentential context, expressed via a vector of *binary* values: for every word g, $\text{vector}_w(\text{g}) = 1$ if g appears along with w in some sentence in C, 0 otherwise. Suppose we now call S the set of all words for which vector_w gives 1, i.e. the set of all words that cooccur with w some sentence, and treat the lexicon L of C as our domain of reference. Now S, built as an approximation of the 'meaning of w' in a DSM, is at the same time the denotation in L of the expression "*the lexical sentential context of word w*" in denotational semantics. With some added complexity, it is straightforward to give a referential translation of vectors which do not contain just binary numbers, but any integer value. What this example shows is that the divide between DSM and denotational semantics is not reference/lack-of-reference, but rather reference to *linguistic strings* (which we can easily record) or to *objects* (which we cannot). The question becomes what the referential meaning of the noun phrase *the linguistic context of "dog"* can tell us about the referential meaning of *dog*. As we hope to show, quite a lot.

But now what about *the linguistic context of "John"*? How can DSMs deal with objects that are often described as being "purely referential", empty of descriptive content? (proper names, demonstratives, personal pronouns, etc.; Kripke 1980). We believe that in DSMs there is no *principled* distinction between proper names and common nouns, but there are very far-reaching *practical* ones. If we consider names like *Barack*

[19]Indeed, we recently became aware of the attempt of Copestake and Herbelot (2012) to provide an extensional semantics for DSMs along lines that are very similar to the ones we sketch here. See Section 6 for a brief review of their approach.

Obama and bare nouns like *presidents*, the difference is small; both will appear in highly informative contexts; people will write contrasting things about *Barack Obama*, but so they will about *presidents* or just about any common noun. Jut like common nouns, proper names can be polysemous (*Italy lost to Spain*—the soccer team; *Italy is boot-shaped*—the land, etc.), and the same techniques mentioned above for common nouns can be used to make the right facets of meaning emerge in the proper combination. But moving on the scale of referential expressions from *Barack Obama* to *Obama*, then to *Barack*, to *that person*, to *him* (or *here, now*, any finite tense marker), the dimension of *homonymy* increases dramatically. Pure referential expressions are infinitely more ambiguous than descriptive ones, and this causes a proliferation of apparent inconsistencies.

There are different strategies to cope with this problem. One is to abandon the attempt to distinguish one *John* from another and focus on what descriptive content remains to names and deictics, for instance the fact that *John* will appear in contexts suitable for an anglophone male human being. At the other end of the spectrum, one could preprocess the input corpus with a (cross-document) *anaphora resolution* system (Ng, 2010, Poesio et al., 2010) to try to unify those names and deictics that are likely to be coreferential.[20]

At least for the time being, we will just treat denotational semantics and DSMs as covering complementary aspects of meaning. To exemplify, suppose we hear the sentence *A dog is barking*. Our distributional-feature-based representation of its constituents will provide us with a sketch of typical contexts in which it can be uttered truthfully, which can orient our perceptual system to pick up the relevant cues to determine if a dog is indeed barking right now, so that we can evaluate the referential meaning of the sentence. Indeed, to step into science fiction for a moment, given that state-of-the-art computational image analysis systems produce vectorial representations of objects (see, e.g., Grauman and Leibe, 2011), the process of verifying the state of affairs described by an utterance against perceptual input could take the form of operations on distributional and perceptual vectors, the former representing the (parts of the) utterance, the second representing objects and possibly events in the perceived world. This idea gains further plausibility if we adopt distributional vectors that record perceptual information coming from vision and other senses, as briefly discussed

[20] An exciting new development that might be useful for these purposes is that of distributional methods to *geo-locate* documents (Roller et al., 2012): The John Smith referred to in a document from Bakersfield is relatively unlikely to be the same John Smith mentioned in an article from Hyderabad.

at the end of the previous subsection.

Intriguingly, the view of the division of labour between DSMs and denotational semantics we just sketched is not too far from those interpretations of Frege's (1892) famous *sense* and *reference* distinction (e.g., Dummett, 1981) that see the sense of a linguistic expression as the *manner* in which we determine the referent (this would be the job of its distributional representation), whereas the denotational meaning is the referent itself.

Bridging denotational and distributional semantics to account for the semantic interpretation of episodic statements is an exciting research program, but it is probably too early to pursue it. However, there are many other aspects of semantics that can be captured independently of the ability to pick up reference from the current state of the world. We reviewed above the many lexical semantic tasks where DSMs representing single words have been very effective despite their lack of direct real-world referencing capabilities (spotting synonyms and other semantic relations, measuring verb-argument plausibility, etc.), and we will discuss in Section 5.2 below potential applications of non-real-world-referring DSMs to the semantics of phrases and sentences.

3 Composition by function application in distributional semantics

Given the success of distributional semantics in modeling the meaning of words (in isolation or in context), it is natural to ask whether this approach can be extended to account for the meaning of phrases and sentences as well. Some pursuers of distributional semantics think that the latter should be limited to modeling lexical meaning. We postpone to Section 5 below a discussion of our theoretical and practical motivations for constructing distributional representations of constituents above the word, since it will be easier to motivate phrasal/sentential distributional semantics after we have introduced (in this and the next section) how we intend to realize it and the current empirical support we have for our approach.

We suggested in the previous section that the (distributional) meaning of a word is a summary of the contexts in which the word can occur. We maintain a contextually-based meaning for phrases and sentences too. Since we typically use the other words in the same sentence as context for our lexical DSMs, many colleagues have asked us what we think the context for sentences should then be. There are many possibilities, and here are just a few: Since any sentence can be extended with adjuncts, coordinates, etc., the context of a sentence could be

given by words that would naturally occur in its extensions. For "*the boy kissed the girl*", context would include words occurring in "*the boy, being madly in love, passionately kissed the girl on her mouth in the park under the tree at midnight...*". In alternatively or in addition, the sentence context could include words or fragments of the previous and following sentences. Another possibility for sentence contexts (in line with DSMs such as LSA and Topic Models) is that they are distributions over the possible documents in which a sentence is more or less likely to occur. Importantly, the approach to composition we will develop in this section allows us to postulate different distributional spaces for different types of linguistic expressions, since we are not committed to the limiting view that word and sentence vectors must live in the same contextual space (we return to this point in Section 3.4 below).

For both words and larger expressions, distributional semantics must find ways to extract from finite evidence an estimate of how their distributional profile would look if we had an infinite corpus available. For words, a large but finite corpus provides a sample of possible contexts of sufficient size to constitute a decent surrogate of infinity; for most phrases and sentences, it does not (given that there is an infinite number of possible phrases and sentences), and we need a different, *compositional* strategy to come up with indirect estimates of their distributional profiles.

Building on what we originally proposed in Baroni and Zamparelli (2010), we present an approach to compositional distributional semantics that relies on Frege's (1892) distinction between "complete" and "incomplete" expressions. Specifically, we distinguish between words whose meaning is directly determined by their distributional behaviour, e.g. nouns, and words that act as functions transforming the distributional profile of other words (e.g., verbs). As discussed in Section 2, representations for the former can be directly induced from their patterns of co-occurrence in a corpus. We add to this standard practice a new view on the incomplete expressions and treat them as transformations, the simplest case of which is a mapping between the corpus-derived vector for a word to the corpus-derived vector for a larger constituent that contains that word. While distributional vectors are extracted from a corpus directly or are the result of a composition operation, distributional functions are induced from examples of their input and output representations, adopting regression techniques commonly used in machine learning. Finally, like in formal semantics, we take syntactic structure to constitute the backbone guiding the assembly of the semantic representations of phrases. In particular, following Montague

(e.g., Montague, 1970b,a), we assume a categorial grammar and define a correspondence between syntactic categories and *semantic types*. In our case, the latter are the types of the semantic spaces where words and other expressions live rather than their domain of denotation.

We first motivate the function-based approach comparing it to the current mainstream "component mixture" view of composition in distributional semantics (Section 3.1). The idea of distributional functions is presented in Section 3.2. Section 3.3 provides mathematical background for our concrete proposal concerning distributional functions, that is then introduced in Section 3.4. Section 3.5 describes how distributional functions can be induced from corpus data. Section 3.6, finally, shows how our approach can be used together with a Categorial-Grammar-based syntactic analysis of sentences to account for the syntax and (distributional) semantics of an interesting fragment of English in parallel.

3.1 Composition by vector mixtures

Mitchell and Lapata (2010), in what is probably the most influential paper on the topic (see also Mitchell and Lapata, 2008, 2009) have proposed two broad classes of composition models focusing on important special cases for each of the classes. These special cases are the additive and multiplicative models we discuss next.

If each word is represented by a vector, the most obvious way to "compose" two or more vectors is by summing them (that is, adding the values in each of their components), as illustrated by the center columns of Table 2. Indeed, this *additive* approach was also the most common one in the early literature on composition in distributional semantics (Foltz et al., 1998, Kintsch, 2001, Landauer and Dumais, 1997).

The other model studied in depth by Mitchell and Lapata adopts instead a *multiplicative* approach. The latter is exemplified by the rightmost columns of Table 2, in which the values in the components of the input vectors are multiplied to derive the composed representation.[21]

The components of additive vectors inherit the cumulative score mass from the corresponding input components, so if an input vector has a high value in a component, the same high value will appear in the composed vector, even if the same component was low or 0 in the other input vector(s): For example, **old+cat** inherits a relatively high

[21] Following Mitchell and Lapata, we use the $\odot$ symbol for component-wise multiplication, since the standard product symbol ($\times$) is used in linear algebra for matrix multiplication, an operation that is not even defined for the pairs of column vectors in Table 2.

				additive		multiplicative	
	dog	**cat**	**old**	**old + dog**	**old + cat**	**old ⊙ dog**	**old ⊙ cat**
runs	1	4	0	1	4	0	0
barks	5	0	7	12	7	35	0

TABLE 2 Left: distributional vectors representing the words *dog*, *cat* and *old*. Center: adding the *old* vector to *dog* and *cat*, respectively, to derive *old dog* and *old cat* vectors. Right: The same derivations using component-wise multiplication.

barks score from **old**. Multiplication, on the other hand, captures the interaction between the values in the input components. For example, since **cat** has a 0 *barks* value, **old** ⊙ **cat** has 0 for this component irrespective of **old**. When both input vectors have high values on a component, the composed vector will get a very high value out of their product, as illustrated for the second **old** ⊙ **dog** component in the table. Mitchell and Lapata characterize these interaction properties of the multiplicative model as a quantitative form of "feature intersection".

In these two models, the input vectors are perfectly symmetric: They contribute to the composed expression in the same way. However, linguistic intuition would suggest that the composition operation is asymmetric. For instance, in the composition of an adjective and a noun, the adjective modifies the noun that constitutes the head of the resulting phrase (an *old dog* is still a *dog*). The effect of syntactic constituency on composition is partially addressed by Mitchell and Lapata's *weighted additive model*, where the vectors are multiplied by different *scalar* values before summing.[22] For example, in an adjective-noun construction we might want the meaning of the noun head to have a stronger impact than that of the adjective modifier. We can then multiply the adjective vector by, say, 0.2 and the nominal one by 0.8 before summing them. In the example from Table 2, $(0.2 \times \mathbf{old}) + (0.8 \times \mathbf{dog}) = (0.8, 5.4)$, a vector that is considerably closer to **dog** than to **old**. Assigning different weights to vectors before summing can also address, to a certain extent, the problem that both addition and multiplication are commutative ($a + b = b + a$; $a \times b = b \times a$), and they thus produce the same vector for, say, *dog trainer* and *trainer dog*, or *dogs chase cats* and *cats chase dogs*.[23]

[22] In linear algebra, single numbers such as 31 or 0.212 are referred to as scalars, to keep them apart from vectors and other multiple-component numerical structures.

[23] It can be shown that, as long as we use the cosine as similarity measure (see Section 2 above), the multiplicative model will not be affected by scalar weights. The effect of multiplying one or both vectors by a scalar before applying the component-wise product is that the resulting composed vector will change its length while

Mitchell and Lapata show that the multiplicative and weighted additive models perform quite well in the task of predicting human similarity judgments about adjective-noun, noun-noun, verb-noun (Mitchell and Lapata, 2010) and noun-verb (Mitchell and Lapata, 2008) phrases.[24] They also show that these simpler models (that we will call, henceforth, the *ML models*) outperform approaches involving more sophisticated composition operations from the earlier literature, such as tensor products (Smolensky, 1990, Clark and Pulman, 2007).

Other studies have confirmed that the ML methods, in particular the multiplicative model, are very competitive in various composition tasks that involve simple phrases and do not test for word order or different syntactic structures (Erk and Padó, 2008, Grefenstette and Sadrzadeh, 2011a, Vecchi et al., 2011, Boleda et al., 2012b). Interestingly and surprisingly, Blacoe and Lapata (2012) recently found that the ML models reach performance close to the one of knowledge-intensive state-of-the-art systems on a full-sentence paraphrasing task. Given the weaknesses of the models we will present below, we can only conjecture that the sentences in this data set fail to test for some crucial syntactic aspects of language (a suspicion that is strengthened by the fact that Blacoe and Lapata obtain excellent results with versions of the additive and multiplicative models that ignore, if we understand correctly, all function words – determiners, negation, etc. – in the test sentences). The ML models are also very well-suited (and empirically effective) for tasks that we will not consider here under the rubric of compositionality but which do involve looking at sentences and larger passages, such as measuring textual coherence (Foltz et al., 1998) or predicting the next word that will be uttered (Mitchell and Lapata, 2009). Besides their good empirical performance, the ML models are extremely easy to implement, which makes them, undoubtedly, the best current choice for practical applications.

Criticism of vector-mixture models

There are principled reasons, however, to believe that the ML models can only account for the simple phrases made of *content* words (nouns, verbs, adjectives) that they have been generally tested on, and that they will not scale up to represent the meanings of sentences, or even sub-sentential constituents with more complex internal structure.

---

pointing in the same direction. Thus, the (cosine of the) angle of the composed vector with any other vector will stay the same.

[24]The other successful model of Mitchell and Lapata (2010), namely the *dilation* model, can be seen as a special way to estimate the weights of the weighted additive model, and we consider it as a special case of the latter here.

One important limitation stems from the fact that both additive and multiplicative models take as input corpus-harvested distributional vectors representing the individual words that form a larger constituent, and produce a *mixture* of these vectors to represent the constituent.[25] The meaning of the phrase *old cat* might indeed be seen as a mixture of the features of *old* things and *cats*. Consider however a determiner phrase (DP) such as *some cat*: The mixture view is suddenly a lot less appealing. First, we face the empirical problem of extracting a distributional vector for a *grammatical* word such as *some*.[26] Unlike the content word *cat*, *some* occurs all over the place in the corpus, without being associated to any specific domain or topic. It is unlikely that the corpus-based vector of *some* will provide meaningful distributional information about this word (and information that distinguishes it from other grammatical words such as *the*, *not* or *to*).[27] Even ignoring this issue, it is highly counter-intuitive to think of the features of *some cat* as a mixture of the features of "*some*" things and of the features of *cats*. Rather, as we will argue below, the role played by *some* and *cat* in composition is deeply asymmetric, with *some* acting like a function operating on the features of *cat*.[28]

The mixture models are moreover unable to capture the radical

[25]Instead of mixtures, we could also speak of averages, since the ML models represent phrases as (functions of) averages of the vectors that compose them. The result of (weighted) addition is a vector pointing in the same direction as the (weighted) arithmetic average of the input vectors. The components in the vector resulting from component-wise multiplication are squares of the geometric average of the values in the corresponding components of the input vectors.

[26]It might be tempting to get away from the thorny issue of representing grammatical words in distributional semantics by simply ignoring them. Indeed, the items in the Mitchell and Lapata (2008, 2010) and Grefenstette and Sadrzadeh (2011a) test sets we will partially introduce in Section 4 only contain content words: To tackle these tests, we are asked to model sentences and phrases such as *table shows results* or *lift hand*, rather than their more natural determiner-enhanced counterparts. This is convenient in the current early stages of compositional modeling, but eventually, even to capture similarity judgments about simple phrases, we will need to take grammatical words into account. For example, to model the intuition that *exterminating rats* is more similar to *killing many/all rats* than to *killing few rats*, you need to include the relevant quantifying determiners in the distributional representations you compare.

[27]If context is extracted from very narrow windows, the distributional vectors of function words might provide some useful information about their syntactic, rather than semantic, properties.

[28]The problem already arises with composition of certain content words, for example so-called "intensional" adjectives such as *former* (Kamp, 1975): A *former owner* is not somebody with a mixture of properties of *former* things and *owners*. See Boleda et al. (2012b), shortly reviewed in Section 4.3 below, for an account of intensional adjectives in compositional distributional semantics.

structural differences that depend on the choice of words in a phrase. Compare for example *lice on dogs* with *lice and dogs*. For the ML models, the two phrases only differ in that one contains *on*-vector-specific component values that are replaced by *and*-specific features in the other, in both cases mixed in the same way with the same *lice* and *dogs* vector components. This completely misses the fact that the two grammatical words reflect different semantic structures, with *on dogs* operating as a modifier of *lice* in the first, while *and* conjoins *lice* and *dogs* in the second. The ML models have no way to capture the different functional nature of words such as *on* (taking a DP to return a locative nominal modifier) or *and* (in this case, taking two DPs and returning a third one representing their conjunction).

Yet another reason to reject additive and especially multiplicative models of composition comes from *recursion*, a crucial property of natural language. One of its consequences is that there is no fixed limit to the number of modifiers. Consider the expressions *cat*, *Siamese*, *spayed*, *toilette-trained* and *short-haired*. In a DSM, they are all likely to have high values for the component corresponding to *pet*. But now, if we mixed their vectors to build a *nice toilette-trained spayed short-haired Siamese cat*, the resulting value for the *pet* component will be astounding, dwarfing any component whose value is high just because it is high in *one* of these expressions.[29]

The problems arising in phrases of just a few words will compound when representing sentences, that are formed by combining these phrases. We just cannot see how, by combining with addition or multiplication the vectors of the words in "*many dogs and some cats have big lice on the back*" we could come up with a meaningful representation for this sentence.

3.2 Composition with distributional functions

To overcome the *a priori* limitations of the additive and multiplicative models, we adopt the view from formal semantics that composition is largely a matter of function application. We thus propose that the distributional meaning of certain words (and certain larger expressions) is not encoded in vectors, but in *distributional functions* that take distributional vectors (or other linear algebraic objects, as we will see)

[29] On the other hand, this effect might be used to capture the Conjunction Fallacy (Tversky and Kahneman, 1983). People might find it more probable that the concept of *pet* is instantiated by a *nice toilette-trained spayed short-haired Siamese cat* than by just a *cat*, despite the fact that the former is a subset of the latter. If what we are after is modeling human judgment, we should strive to preserve a small amount of the effect given by multiplicative models also in our functional approach to distributional semantics.

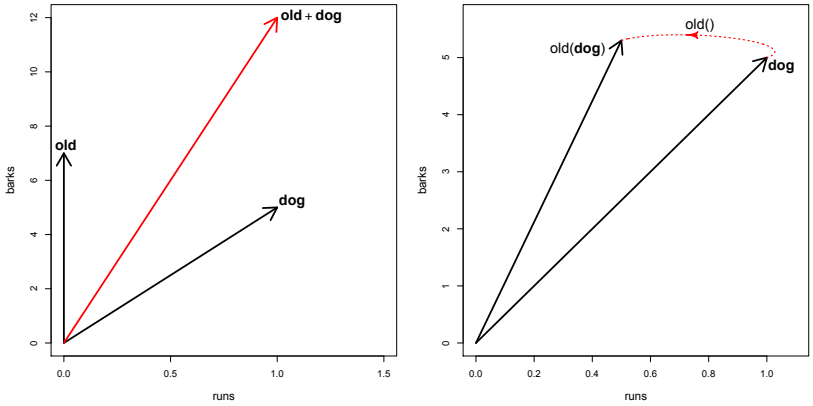


FIGURE 2 Mixture-based composition, such as the additive model illustrated in the left panel, takes distributional vectors representing two words and "mixes" them (e.g., adds their components) to obtain the representation of a phrase. In a function-based model, such as the one illustrated on the right, one of the two words is not a vector, but a function exerting an action on the argument vector to move it to a new position in semantic space.

as input and return other vectors (or other objects) as output by operating on the input components. Nouns, DPs and sentences are still represented as vectors, but adjectives, verbs, determiners, prepositions, conjunctions and so forth are all modeled by distributional functions. An approximate geometric intuition for the difference between a mixture and a functional approach is given in Figure 2.[30]

Under the functional approach, we no longer treat grammatical words such as *many* and *some* as highly problematic corpus-harvested distributional vectors to be mixed with content word vectors, but rather as functions from and onto representations of phrases that also include content elements (e.g., from *dogs* to *some dogs*).

Moreover, different grammatical words impose different compositional structures on phrases. Coming back to the *lice on dogs* vs. *lice and dogs* example, we would model *on* as a function from DPs onto distributional functions that, in turn, act as modifiers of nominal vectors: $(\mathcal{ON}(\mathbf{dogs}))(\mathbf{lice})$; whereas *and* could be an unary function from a pair of DP vectors onto one representing their conjunction: $\mathcal{AND}(\mathbf{lice}, \mathbf{dogs})$.[31]

[30] The figure only illustrates the simple case of functions operating on vectors and generating outputs that live in the same semantic space of the inputs.

[31] More precisely, in our approach we "curry" *and* into a unary function from DPs

The asymmetry between a noun modified by a prepositional phrase and a conjunction of DPs is seamlessly captured.

Anybody familiar with the classic treatment of compositionality in formal semantics will agree that the functional approach is more promising than vector-mixture methods. However, we must now spell out what it means, in concrete, to apply a function to a vector, and how we can come up with (i.e., in the jargon of machine learning, "learn") distributional functions that perform just the operations we want on their inputs. Before we discuss these issues, we introduce the relevant concepts from linear algebra.

3.3 Linear transformations, matrix-by-vector products and tensors

The class of functions on vectors known as *linear transformations* or *linear maps* plays a fundamental role in linear algebra, the branch of mathematics that studies the algebra of vectors (e.g., Axler, 1997, Meyer, 2000, Strang, 2003). When looking for an equivalent to compositional function application in the vector-based distributional framework, it is thus natural to start with the hypothesis that the relevant functions are linear transformations. A linear transformation takes a vector of size J and returns a vector of size I (where J might equal I), where each output component is a linear combination of all input components, that is, each output component is a weighted sum of the input components.[32]

There is an important correspondence between linear transformations and matrices (we introduce matrices in the next paragraph). Recall from Section 2 above that the N-dimensional vector space is the set of all possible vectors of size N.[33] Given the J- and I-dimensional vector spaces, any linear transformation from the first onto the second is entirely characterized by a matrix of shape $I \times J$. The application of the corresponding linear transformation is given by the product of the matrix by an input vector from the J-dimensional space.

A *matrix* of *shape* $I \times J$ is an array of numbers whose components (or

onto functions from other DPs onto the conjoined form.

[32]The defining characteristics of a linear transformation are that (i) the linear transformation of the sum of two vectors must equal the sum of the linear transformations of the two vectors ($f(\mathbf{a} + \mathbf{b}) = f(\mathbf{a}) + f(\mathbf{b})$), and that (ii) the linear transformation of a vector multiplied by a scalar equals the product of the scalar by the linear transformation of the vector ($f(k\mathbf{a}) = kf(\mathbf{a})$). As a consequence of this latter property, if vectors are parallel in the input space, their linear transformations will stay parallel in the output space, since multiplying a vector by a scalar changes its length but not its direction.

[33]In this section, given the emphasis on vector algebra, we ignore the association of components to linguistic labels in distributional semantic spaces.

cells) are indexed by two integers i, ranging from 1 to I, and j, ranging from 1 to J.[34] An $I \times J$ matrix is naturally thought of as a rectangular object with I rows and J columns. When we multiply an $I \times J$ matrix by a vector of size J, we obtain a vector of size I whose i-th component is a weighted sum of all J input components, each multiplied by the value in the ij-th cell of the matrix. In mathematical notation, given a matrix $\mathbf{M}$ of shape $I \times J$ and a vector $\mathbf{v}$ of size J, each component w_i of the I-sized vector $\mathbf{w}$ resulting from the product $\mathbf{w} = \mathbf{M} \times \mathbf{v}$ is given by:

$$w_i = \sum_{j=1}^{j=J} M_{ij} \times v_j$$

For example, the following 3×2 matrix $\mathbf{M}$ encodes a linear transformation from the 2- onto the 3-dimensional space.

$$\mathbf{M} = \begin{pmatrix} 1 & 5 \\ 1 & 2 \\ 4 & 0 \end{pmatrix}$$

Let us apply the linear transformation encoded in $\mathbf{M}$ to the 2 component vector $\mathbf{v}$:

$$\mathbf{v} = \begin{pmatrix} 3.1 \\ 1 \end{pmatrix}$$

We obtain the 3 component vector $\mathbf{w}$ as illustrated in the following equation:

[34] We prefer the non-standard term *shape* to the more commonly used *size* to stress the fact that matrices and the other multi-index objects we will introduce next are not only characterized by the number of components they have, but also by how the latter are arranged according to their indices. Consider for example a 2×3 matrix. By interpreting the product symbol in the usual way, we can correctly state that it is an object of size 2×3, in the sense that it has $2 \times 3 = 6$ components; but, using the term *shape*, we want to emphasize that the 6 components are arranged into a 2-by-3 array. When we say that two linear algebraic objects have the same shape, we mean that the objects have the same number of indices (below, we will call this quantity the *order* of the objects), *and* their indices have the same size (obviously, by "size of the index I" we mean I itself: the size of the first index of a 2×3 matrix is 2). A 2×3 matrix has a different shape from a 3×2 matrix, despite the fact that they have the same size (6 components) and number of indices (2 indices). Their indices, however, have different sizes (the first indices have sizes 2 and 3, the second indices sizes 3 and 2, respectively). Note that a vector with I components is, trivially, both an I-sized and an I-shaped object.

$$\mathbf{w} = \mathbf{M} \times \mathbf{v} = \begin{pmatrix} 1 & 5 \\ 1 & 2 \\ 4 & 0 \end{pmatrix} \times \begin{pmatrix} 3.1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \times 3.1 + 5 \times 1 \\ 1 \times 3.1 + 2 \times 1 \\ 4 \times 3.1 + 0 \times 1 \end{pmatrix} = \begin{pmatrix} 8.1 \\ 5.1 \\ 12.4 \end{pmatrix}$$

Note how, for example, $w_1 = 8.1$ is given by the sum of the $\mathbf{v}$ components weighted by the 1st row of $\mathbf{M}$ ($v_1 = 3.1$ is multiplied by the value in the $(1,1)$-th cell of $\mathbf{M}$, that is, $M_{1,1} = 1$; $v_2 = 1$ by that in the $(1,2)$-th cell, that is, $M_{1,2} = 5$).

The notion of linear transformations extends to arrays with more than two indices, that in linear algebra are called tensors. A *tensor*, more precisely, is any numerical array $\mathcal{T}$ whose values are indexed by n indices (i.e., any object of shape $I_1 \times \ldots \times I_n$). The number of indices of a tensor is called the *order* of the tensor. A vector is a first-order tensor (components are addressed via a single index), matrices are second-order tensors (one index for the rows, one for the columns), the $I \times J \times K$ object mapping vectors to matrices we are about to discuss is a third-order tensor, and so on. In this article, depending on the context, we might use the term tensor to refer to any indexed number array (including vectors and matrices), or, when contrasted with the vectors or matrices, to objects of order larger than second (the latter are also referred to as higher-order tensors).[35]

Coming back to linear transformations and how they extend to general tensors, suppose for example that we want to map from K-sized vectors onto $I \times J$-shaped matrices (since we just saw that matrices encode linear transformations, this is our equivalent of a function that returns a function). Again, each ij-indexed component of the output matrix will derive from a weighted sum of all K input vector components. A tensor $\mathcal{T}$ with shape $(I \times J) \times K$ stores the lists of K weights needed to derive each ij-th component, and the mapping is performed by the product of $\mathcal{T}$ with the input vector.[36] Similarly, if we want to map from $K \times L$ onto $I \times J$ matrices, the mapping tensor $\mathcal{T}$ will have shape $(I \times J) \times (K \times L)$, and the mapping will be performed by the product of $\mathcal{T}$ with the input matrix. In this case, the $ijkl$-th component

[35] Again, please pay attention to the terminology: The *size* of an index I is I; the *size* of a tensor is the total number of components it contains, that is, the product of the sizes of its indices; the *order* of a tensor is the number of indices used to arrange its cells; the *shape* of a tensor refers to both the number of indices and their respective sizes.

[36] Here and below, we add parentheses to the index structure of higher-order tensors, in order to emphasize their functional role. We might denote the shape of the same $I \times J \times K$ tensor with $(I \times J) \times K$ if we intend to use it to map from K-sized vectors onto $I \times J$ matrices (as in the main text), or with $I \times (J \times K)$ if we use it to map from $J \times K$ matrices onto I-sized vectors.

of $\mathcal{T}$ will contain the weight that the kl-th cell of the input matrix must be multiplied by when computing the weighted sum of all input cells that generates the ij-th output cell. We refer to the product operation needed to map from and onto objects other than vectors as *generalized matrix-by-vector product*[37] (vanilla matrix-by-vector multiplication is of course a special case of the generalized product).

The formula to compute each output component in generalized matrix-by-vector multiplication is as follows. Given input $\mathcal{V}$ with shape $J_1 \times \ldots \times J_n$ and components denoted by $V_{j_1 \ldots j_n}$, and a linear transformation encoded in a tensor $\mathcal{M}$ with shape $(I_1 \times \ldots \times I_m) \times (J_1 \times \ldots \times J_n)$ and components denoted by $M_{i_1 \ldots i_m j_1 \ldots j_n}$, each component $W_{i_1 \ldots i_m}$ of the output tensor $\mathcal{W}$ (of shape $I_1 \times \ldots \times I_m$) is given by a weighted sum of all input components as follows:

$$W_{i_1 \ldots i_m} = \sum_{j_1=1}^{j_1=J_1} \cdots \sum_{j_n=1}^{j_n=J_n} M_{i_1 \ldots i_m j_1 \ldots j_n} V_{j_1 \ldots j_n}$$

We conclude this brief survey of the linear algebra behind our approach by pointing out that there exists an operation of *tensor transposition* that, in the restricted version we need here,[38] swaps the last two indices of a tensor. Specifically, in the third-order case, each component of the transposed tensor $\mathcal{T}^T$ is given by: $T^T_{ijk} = T_{ikj}$. A useful property of transposition is that

$$(\mathcal{T} \times \mathbf{v}) \times \mathbf{w} = (\mathcal{T}^T \times \mathbf{w}) \times \mathbf{v}$$

That is, the result of multiplying a third-order tensor (in the generalized matrix-by-vector product sense) by one vector, and then the

[37] We use this term to underline the fact that the general product operation we assume here is equivalent to unfolding both the input and the output tensors into vectors, applying standard matrix-by-vector multiplication, and then re-indexing the components of the output to give it the appropriate shape. For example, to multiply a $(I \times J) \times (K \times L)$ fourth-order tensor by a $K \times L$ matrix, we treat the first as a matrix with $I \times J$ rows and $K \times L$ columns and the second as a vector with $K \times L$ components (e.g., a $(2 \times 3) \times (3 \times 3)$ tensor can be multiplied with a (3×3) matrix by treating the latter as a 9 component vector and the former as a 6×9 matrix). We perform matrix-by-vector multiplication and then rearrange the resulting $I \times J$-sized vector into a matrix of shape $I \times J$ (continuing the example, the values in the 6 component output vector are re-arranged into a 2×3 matrix). This is a straightforward way to apply linear transformations to tensors (indeed, there is a precise sense in which all tensors with the same shape constitute a "vector" space). There are alternative ways to multiply tensors of various orders (including standard matrix-by-matrix multiplication) that are not relevant for our current purposes (Bader and Kolda, 2006, Kolda and Bader, 2009).

[38] See Bader and Kolda (2006, Section 3.3) for a more general and detailed discussion of various properties of tensors including transpositions.

OLD	runs	barks
runs	0.5	0
barks	0.3	1

$\times$

	dog
runs	1
barks	5

$=$

	OLD(dog)
runs	$(0.5 \times 1) + (0 \times 5) = 0.5$
barks	$(0.3 \times 1) + (5 \times 1) = 5.3$

TABLE 3 The adjective *old* as the distributional function encoded in the matrix on the left. The function is applied to the noun *dog* via matrix-by-vector multiplication to obtain a compositional distributional representation of *old dog* (right).

resulting matrix by another vector is the same as that of multiplying the tensor transpose by the two vectors in the opposite order. We will see a linguistic application of this property in Section 3.6 below.

3.4 Distributional functions as linear transformations

We propose that distributional functions are linear transformations on semantic vector (or more generally tensor) spaces. First-order one-argument distributional functions (such as adjectives or intransitive verbs) are encoded in matrices. The application of a first-order function to an argument is carried out via matrix-by-vector multiplication as follows:

$$f(a) =_{def} \mathbf{F} \times \mathbf{a} = \mathbf{b}$$

where $\mathbf{F}$ is the matrix encoding function f as a linear transformation, $\mathbf{a}$ is the vector denoting the argument a and $\mathbf{b}$ is the vector output to the composition process.

Let us take as an example the composition of an adjective with a noun. Let us assume, as we have done above, that nouns live in a 2-dimensional space. Hence the adjective, as a function from nouns to nouns, is a 2×2 matrix (it multiplies with a 2 component vector to return another 2 component vector). Suppose that *old* is associated to the toy matrix on the left of Table 3. Then, applying it to the usual *dog* vector returns the vector for *old dog* shown on the right of the same table.

The matrix labels illustrate the role played by each cell of the distributional function matrix in mapping from the input to the output vector: Namely, the ij-th cell contains the quantity determining how much the component corresponding to the j-th input context element contributes to the value assigned to the i-th context element in the output vector. For example, the first cell of the second row in the toy **OLD** matrix indicates that the *runs*-labeled component of the input

noun will contribute 30% of its value to the *barks*-labeled component of the *old N* output.

In the case of *old*, we can imagine the adjective having a relatively small effect on the modified noun, not moving its vector too far from its original location (an *old dog* is still a *barking* creature). This will be reflected in a matrix that has values close to 1 on the diagonal cells (the ones whose weights govern the mapping between the same input and output components), and values close to 0 in the other cells (reflecting little "interference" from other features). On the other hand, an adjective such as *dead* that alters the nature of the noun it modifies more radically could have 0 or even negative values on the diagonal, and large negative or positive values in many non-diagonal cells, reflecting the stronger effect it has on the noun.

Table 3 also illustrates an important point about matrices and higher-order tensors when they are used as distributional functions. In Section 2, we remarked that a distributional semantic space is characterized not only by its mathematical properties. The linguistic tags associated to the components also matter. Similarly, when looking at matrices and tensors from the distributional point of view, the sets of labels associated to their cells also matter. Table 3 contains a 2×2 matrix with the same labels for rows and columns (this is not necessary: it happens here because adjectives, as we have already stated, map nouns onto the same nominal space), and where the first cell, for example, weights the mapping from and onto the *runs*-labeled components of the input and output vectors. A mathematically identical matrix where the first cell maps, say, from *banana* to *sympathy* components is a very different linguistic object. Just like for vectors, we can meaningfully speak of tensor spaces. Just like for vectors (that are of course a special case), a *distributional tensor space* is the space of all possible tensors with the same shape *and with the same associations of index elements to linguistic content labels* (that, ultimately, reflect information about possible context distributions).[39]

Different matrices can act differently on the same vectors, but also the same matrix can act differently on different vectors. Since a linear distributional function derives each component of the output vector by combining a set of input components with different weights, there is a lot of room for modeling varied semantic effects. Consider for example the case of *some* (that, in this paper for sake of simplicity, we assume to be a function taking a noun and returning a determiner phrase).

[39]When we say that a component (or cell) of a tensor is associated to a (set of) linguistic label(s), we mean, more precisely, that the index element or set of index elements needed to uniquely address that component are associated to the label(s).

Some should have a different effect depending on whether it is applied to a mass (*some coffee*) or count (*some cats*) noun. Consider another toy example where, for explanatory reasons, we promoted nouns from 2 to 3 components (with labels NMass, NCount1 and NCount2), and DPs are vectors with 4 components (DPOther, DPObj1, DPObj2, DPSubst). Suppose moreover that mass nouns tend to have high values in the first component (NMass) and count nouns in the second and third (NCount1, NCount2). Similarly, say that the quantifying characteristics of a DP that would be relevant to a substance are expressed in the fourth component (DPSubst), whereas those that are relevant to countable objects are captured by the second and third components, DPObj1 and DPObj2 (with the first component, DPOther, expressing other properties). Then, *some* could be a matrix with the following form:

SOME	NMass	NCount1	NCount2
DPOther	1	1	0
DPObj1	0	3	2
DPObj2	0	2	3
DPSubst	5	0	0

When the distributional *some* function is applied to a noun, that is, the **SOME** matrix is multiplied by the noun vector, the value in the first component of the resulting DP is determined, with equal weights, by both a mass and a count component of the input noun (since NMass and NCount1 have the same value and NCount2 is 0), reflecting the fact that this component expresses properties that are unrelated to the count/mass distinction. The second and third components of the DP (containing information about quantity characteristics of countable objects) will depend on the second and third components of the noun, that are high in count nouns only, with a strong positive dependence between the input and output components (the input values are multiplied by weights higher than 1 before summing them). The fourth component of the DP, encoding quantification information pertaining to substances, is just (positively) affected by the first component of the noun vector, the one that is high in mass nouns only.

This toy example should give an idea of the flexibility of the linear approach. With realistically sized vectors and matrices, it is possible to capture many more patterns, and in a more granular way (for example, including negative weights to capture inverse correlations between input and output components, and encoding semantic properties such as substance and objecthood as a distribution over a large set of input

and output components, rather than just one or two).[40] Indeed, it is the flexibility given by the wealth of information that can be encoded in large matrices that makes us hopeful that the linear approach will be able to handle the regular polysemy phenomena we discussed in the introduction and in Section 2.3. Probably not by chance, in Baroni and Zamparelli (2010), we found that some of the adjectives that are best modeled by the linear approach are very polysemous terms such as *new*, *great* or *large*.

Before we move on to discuss various aspects of the linear-transformation-based composition framework, we will highlight two properties of the operation employed to perform linear transformations, namely (generalized) matrix-by-vector products, that have important consequences for how we use them in the linguistic composition framework.

First, generalized matrix-by-vector multiplication is only defined when the last indices of the first term have the same shape as the second term. For example, you cannot multiply an $I \times I$ matrix by a J component vector, or by an $I \times J \times K$ tensor.[41] This resonates with types in formal semantics, where you cannot apply, say, an $e \rightarrow t$ function to any argument but those of type e, a point we will return to multiple times below.

Second, the generalized matrix-by-vector product, unlike the product of scalars, is not commutative. For example, $\mathbf{OLD} \times \mathbf{FAST} \times \mathbf{car} \neq \mathbf{FAST} \times \mathbf{OLD} \times \mathbf{car}$, and $\mathcal{CHASE} \times \mathbf{cats} \times \mathbf{dogs} \neq \mathcal{CHASE} \times \mathbf{dogs} \times \mathbf{cats}$, in accordance with our linguistic expectations about the corresponding constructions.[42]

Mapping between different distributional semantic spaces

The toy **SOME** matrix we just discussed highlights another important novelty with respect to the ML mixture models. Addition and multiplication will create composed vectors that must live in the same vector space as their inputs. On the other hand, linear transformations can map onto different spaces from those of their domains. For example,

[40] Speaking of flexibility, note that the multiplicative model is a special case of linear transformation where the matrix has as diagonal elements the components of a corpus-derived vector for one of the words in the phrase and 0s elsewhere. This matrix is then multiplied by the vector representing the other word. It is a bit more involved but also possible to encode the additive model as matrix-by-vector multiplication.

[41] At least, you cannot do it using the generalized matrix-by-vector product operation we defined in Section 3.3.

[42] Treating nominal conjunctions as third order tensors, we predict inequalities of the sort: $\mathcal{AND} \times \mathbf{cats} \times \mathbf{dogs} \neq \mathcal{AND} \times \mathbf{dogs} \times \mathbf{cats}$ that are not intuitive from a truth-theoretical point of view. It remains to be seen if they are justifiable in a distributional perspective.

SOME maps nominal vectors living in a 3-dimensional space to DP vectors that live in a 4-dimensional space, where the two spaces also differ in terms of the linguistic labels associated with the components. Linear transformations do not *need* to map vectors to an output space that differs from their domain (unlike earlier methods based on tensor operations where each composition step resulted in tensors of higher orders: see Mitchell and Lapata, 2010 for discussion). For example, we have already seen that the **OLD** matrix in Table 3 above maps nouns onto adjective-noun phrase vectors that live in the same space, in accordance with the standard analysis of (attributive) adjectives as modifiers that take nouns as input and return other nouns as output.[43] However, in other cases the possibility of defining different distributional semantic spaces for different constituents gives us further flexibility for encoding contextual information into features that are rich enough to capture different nuances of meaning. For example, nouns might live in a semantic space characterized by features (that is, linguistic labels associated to specific components) that are content words connected to the target nouns by interesting syntactic relations (along the lines of Curran and Moens, 2002a, and many others), whereas sentences might be vectors living in a space of topics of conversation, à la Griffiths et al. (2007), or even in a space characterized by more abstract contextual cues to discourse structure and such (see the beginning of Section 3 above for some conjectures about the nature of sentence space).

Of course, if different spaces have to be handcrafted to suit the needs of different categories we would run against the objection that any semantic effects uncovered using them might be due to our specific choice of vector dimensions. Even with the best intentions, the selection of the ideal features for each space might be problematic, especially if spaces begin to multiply.[44] To address this potential criticism, we ul-

[43]Indeed, this property of attributive adjectives make them an ideal illustration of an important property of language, recursivity. Since the application of an adjective to a noun gives a result in the same space as the original noun, the operation can be easily repeated an indefinite number of times. That is, if we have matrices representing *large*, *old* and *brown* separately, we can easily apply them in sequence to generate a meaning for *"large old brown dog"* (however, see footnote 44 for an alternative). Ongoing work aims to uncover the semantic effects of various deviant adjectival sequences (e.g., contradictions like *"old young dog"* and redundancies such as *"brown brown dog"*).

[44]A case in point is that of recursive structures, which we just discussed in the context of adjectival modification. According to some authors, complements are the only truly recursive cases in languages, modifiers are not. In this view (see Cinque, 2002, Cinque, 2010, Scott, 2002 and most representatives of the 'cartographic' approach to syntax), adjectives and adverbs follow a natural ordering, e.g., SIZE > COLOR > SHAPE, and there is a finite sequence of types of modifiers available

timately envision a fully automated labeled-component selection system. The general idea is that we should initially prepare a single huge semantic space whose components are associated with linguistic features at many different levels (lexical, categorial, syntactic-structural features, information-structural, topical, etc.), then extract the n features that are most informative in the representation of the various linguistic structures that we regard as non-functional (minimally Ns, DPs and S), e.g., features whose associated components exhibit the highest within-category variance.[45] These components and associated features will then characterize the space where the distributional meanings of the individual expressions in each non-functional category reside (the labels of components of higher-order tensors are, deterministically, those of their input and output categories).

Modeling functions operating on functions

In Section 3.3 we showed how linear transformations by matrix-by-vector multiplication can be generalized to structures with an arbitrary number of indices. This is fundamental for their use in semantics, where functions must manipulate other functions both as input and as output. Let us start with an example of the second case. If nouns are I component vectors, DPs J component vectors and we adopt the standard analysis of a preposition such as *on* (in a noun-modifying context) as taking an input DP to return an adjective-like function that takes and returns a noun, in distributional terms we will treat *on* as an $(I \times I) \times J$ tensor. When this object is multiplied by a J-sized DP vector, it returns an $I \times I$ matrix (analogous to our representation of the adjective *old* above), that can then be multiplied by a noun, as in the following derivation of an I-sized vector for *louse on dogs*:

for each noun. This view could be easily (if laboriously) accommodated within the present framework by assuming that in, say, *"large blue car"* the COLOR adjective *blue* does not project *car* back into its original noun-space, but into a slightly different space which is that of nouns-plus-COLOR-information. In turn, *large* would map elements from the nouns-plus-COLOR-information to the nouns-plus-COLOR-and-SIZE-information space, and so forth. This entails that *large* could not be directly applied to *car*; rather, one would have to first apply an invisible mapping from *car* to *car-with-unspecified-color* (to move the input to the correct space), then apply *large* to it. The oddness of applying multiple adjectives in the wrong order ("*red large car*") would be the effect of applying a function to an input in the wrong space. It is an open question whether this additional complexity is justified. What matters here is that it would pose no special theoretical problems to our approach.

[45] Needless to say, the initial computation to calculate the most informative features for each primitive category will be humongous. But it needs to be carried out only once.

$$(\mathcal{ON}_{(I\times I)\times J} \times \mathbf{dogs}_J) \times \mathbf{louse}_I = [\mathcal{ON}(\mathbf{dogs})]_{I\times I} \times \mathbf{louse}_I$$
$$= \{[\mathcal{ON}(\mathbf{dogs})](\mathbf{louse})\}_I$$

In our running toy example, nouns are 3-component vectors and DPs have 4 components, so $\mathcal{ON}$ would be tensor of size $(3 \times 3) \times 4$ mapping from a 4-dimensional vector space onto a 3×3-dimensional tensor space.

Similarly, suppose that an intransitive verb is analyzed as a VP (verb phrase), that is, a function from DPs to sentences, and thus as a matrix of shape $K \times J$ mapping from the J-dimensional DP space (in which subject vectors live) onto the K-dimensional sentence space. A transitive verb will then be a third-order $(K \times J) \times J$ tensor mapping from the J-dimensional DP space (where of course also DPs that function as sentential objects live onto a VP, that is, an intransitive-verb-like $K \times J$ matrix.) If, to continue the toy example, DPs are vectors with 4 components and sentences vectors with, say, 2 components, then intransitive verbs would be 2×4 matrices and transitive verbs would be $(2 \times 4) \times 4$ tensors. Readers familiar with the standard theory of semantic types will recognize, again, the analogy between the role of denotation-based types in the standard theory and the shape of tensors (plus the associated index labels!) in the distributional approach, a point we return to more explicitly in Section 3.6.

Consider next the case of a higher-order function that takes other functions as arguments. As we know from formal semantics (see, e.g., Heim and Kratzer, 1998, Chapter 5), a relative pronoun acts as a bridge between a verb phrase and a noun: it modifies the noun with the verb phrase. In the denotational view, it is represented by the 2-argument function that takes a verb phrase (viz., a property) and a noun (viz., a property) to return a noun denoting the intersection of the two properties, for instance ⟦dogs⟧ ∩ ⟦eat meat⟧. Note that this is the semantics of intersective conjunction, as we see it in the examples in (6), applied to verbs, nouns, DPs and adjectives.

(6) a. Bill [walked] and [talked].
b. My [friend] and [colleague] gave me a long hug.
c. As [a mother] and [a well-respect researcher], Sue has much to share with us.
d. A [tall] and [handsome] gentleman

How can we capture the same intuition in distributional semantics? We do not have an overt *and* as in the examples above, so in first approximation we must capitalize on the presence of the relative pronoun,

treated as a function that takes a VP such as *"eat meat"* as input and returns the noun modifier *"which eat meat"*.[46] What kind of linear algebraic object should *which* be to serve this role? Suppose, as before, that nouns live in 3-dimensional space, DPs in 4-dimensional space and sentences in 2-dimensional space. Consequently, VPs are 2×4 matrices and a noun modifier (such as an adjective) is a 3×3 matrix. It follows that *which* is a fourth-order tensor with shape $(3 \times 3) \times (2 \times 4)$ mapping from 2×4 input VP matrices to 3×3 output noun-modifier matrices. More generally, if the noun space is I-dimensional, the DP space is J-dimensional and the sentence space is K-dimensional, then a relative pronoun such as *which* is an $(I \times I) \times (K \times J)$ tensor.

Measuring similarity of tensors

In the same way that we can measure degrees of similarity (and other properties) of two or more vectors living in the same vector space, we can measure the similarity (and other properties) of matrices and higher-order tensors, as long as they have the same shape (as we remarked in footnote 37, the set of all same-shape tensors constitutes, indeed, a vector space).[47] In particular, we can always represent an n-th order tensor of shape $I_1 \times \ldots \times I_n$ as a vector with a number of components that equals the product $I_1 \times \ldots \times I_n$, and thus we can use exactly the same methods we adopt to measure the similarity of vectors (such as the cosine comparison introduced in Section 2) to measure the similarity of tensors of any order but with the same shape. Intuitively, the cells of a matrix (or higher-order tensor) contain weights specifying the impact that each component of the input has on each component of the output. Two matrices (or tensors) are similar when they have a similar weight distribution, i.e., they perform similar input-to-output component mappings (we might expect the **DECREPIT** matrix to dampen the *runs* component of an input noun just like the **OLD** matrix in Table 3 does).

On the other hand, there is no straightforward way to compare tensors of different orders or shapes. This entails that it is possible to compare all and only the linguistic structures that live in the same distributional semantic space, a limit which we regard as a positive feature, if the goal is a more constrained theory of language: In the ML models, all words and larger constituents live in the same space,

[46] In Section 3.6, we will handle the more difficult case in which the pronoun acts as object of the relative: *"meat which animals eat"*.

[47] In line with our idea of distributional space as a linear algebraic space enriched with linguistic index labels, a meaningful comparison will only be possible between identically shaped tensors that also share the same associations of index elements to labels.

so everything is directly comparable with everything else. This is too lax: asking for the degree of similarity between, say, *the* and *eating carrots* is asking an ill-conceived question. At the same time, we are aware that the ban imposed by our method can sometimes be too strong. In its pure form, it allows nouns to be compared to nouns, since they are represented by vectors with the same number of components, but not to adjectives, which are matrices. As a result, *Rome* and *Roman*, *Italy* and *Italian* cannot be declared similar, which is counter-intuitive. Even more counter-intuitively, *Roman* used as an adjective would not be comparable to *Roman* used as a noun.

We think that the best way to solve such apparent paradoxes is to look, on a case-by-case basis, at the linguistic structures involved, and to exploit them to develop specific solutions.[48] For example, a way to measure similarity between an adjective and a noun would be to apply the adjective matrix to a number of vectors representing nouns that are frequently modified by the adjective, average these adjective-noun vectors, and compare the resulting averaged vector (that, as a sum of adjective-noun vectors, is still a vector living in nominal space) to the noun of interest. For example, *Italian* could be represented for these purposes by an average of the vectors of *Italian citizen*, *Italian flag*, *Italian government*, *Italian food*, etc. (Baroni and Zamparelli, 2010, show that a similar method gives good results in an adjective clustering task).

In the case of pairs such as *Italy* and *Italian*, perhaps the right linguistic intuition to capture is not about similarity, but about the fact that these two forms are related by a morphological process of derivation, whereby the lexical function *-(i)an* is applied to nominal roots to obtain the corresponding denominal adjectives. As we discuss in Section 3.5 below, a nice feature of our approach to composition learning is that it naturally extends to lexical functions of this sort (in the case at hand, *-(i)an* would be a tensor mapping noun vectors to adjective matrices). Then, "similarity" of *Italy* and *Italian* could simply be modeled by observing that in our system the latter (at least when used with a transparent denominal meaning) is derived from the former.[49]

[48] One could also adopt purely mathematical methods to project tensors of different orders and sizes onto the same space. We doubt that such general methods would be very effective empirically (the naturalness of the task is cued by the fact that one of the methods to pursue it is called "Procrustes Analysis"; Wang and Mahadevan, 2008), they are so general that they would then also allow the unconstrained similarity comparisons we want to avoid (e.g., the same method used to compare *Italy* and *Italian* could also be used to compare *the* to *eating carrots*).

[49] As a reviewer observed, as a model of word formation this would lead to over-

As intuitively clear, such special methods to capture similarity cannot be applied anywhere and for any category. Some cases are and will remain incomparable. It is an empirical issue whether this restriction is too severe, or if, on the contrary, our assumptions impose just the right constraints on the scope of similarity and related properties.

3.5 Inducing distributional functions from corpus data

We have argued that distributional functions might be a more appropriate representation to capture composition than vector mixtures. However, we have not yet addressed the fundamental issue of how the operations performed by the distributional composition functions corresponding to individual words and constructions are specified. Assuming that distributional functions are linear transformations, the question can be framed more precisely as: How do we determine the values to fill the cells of the tensor representing a distributional composition function?

Obviously, we do not want to fill them by hand. It would be highly impractical, since realistically-sized tensors will contain at least a few thousand cells, and a useful lexicon should contain thousands of such objects: one per adjective, one per verb, etc. The manual approach would also be theoretically undesirable, since we are pursuing systems that, like humans, acquire semantic knowledge from naturally occurring data.

We propose instead to learn a distributional function by extracting examples of how its input and output tensors should look like from the corpus, and using standard machine learning methods to find the set of weights in the matrix that produce the best approximations to the corpus-extracted example output vectors when multiplied by the corresponding input vectors (the input and output vectors used to estimate the matrix weights are called *training examples* in the machine learning literature, and the estimation process *training*; we use the terms *training*, *learning* and *inducing* more or less as synonymous). Consider the case of the determiner *some*. The idea is to collect directly from the corpus pairs of distributional vectors matching the templates <*N*, *some N*> (<*dog*, *some dog*>; <*cats*, *some cats*>; <*coffee*, *some coffee*>; etc.). We then use a statistical algorithm (regression) to find the sets of weights that, on average, provide the best approximation to each output component as a weighted sum of the corresponding input components across the training set. These weights will fill the **SOME**

generation. However, the techniques described in Vecchi et al. (2011) for detecting semantically anomalous AN combinations might be applied to exclude, at least, semantically deviant root-affix combinations such as **first-er*.

matrix.[50]

Clark (2013b) wonders if extracting composed vectors directly from the corpus is in the true spirit of compositional semantics. We think it is, since we only use these vectors in the learning phase as examples of how the output of compositional processes should look like: It is acceptable even to a Fregean compositionalist to try to find out what a compositional function does by comparing examples of its input and output. Less compositionally inclined researchers of language development, such as Tomasello, 2003, actually view the acquisition process more as one of *decomposing* larger chunks by discovering their internal structure, than one of putting pieces together to build those chunks. Note that we can throw away the corpus-extracted examples of phrase vectors after learning, and use our fully compositional system to (re-)generate all phrases and sentences. But this might not always be a good move: As we briefly discuss at the end of Section 3.5 below, in some cases there might be good reasons to prefer a "dual-route" view where both compositionally-derived and directly corpus-induced phrase representations are available.

Learning by regression

Algorithms to predict a continuous numerical value (such as a the value in a component of the output vector) from a set of features (such as the input components) are called *regression* methods, and they are widely studied in statistics and machine learning (Hastie et al., 2009). We do not need to delve here into the complexities of regression algorithms. As linguists, we limit ourselves to borrow state-of-the-art methods from the relevant literature. Suffice to say that alternative regression algorithms mostly differ in how they find a trade-off between fitting the training data as best as possible (i.e., finding sets of weights that produce output values that are very similar to those in the example output vectors) and avoiding "overfitting", that is, avoiding very *ad-hoc* weight settings that might produce an excellent approximation of the training set, but won't generalize to new data, since they over-adapted to the random noise present in any set of examples, including the training set.

From a linguistic perspective, it is more interesting to ask whether distributional vectors, directly harvested from the corpus for the composed expressions we want to model, are a good target for function learning. Theoretically, since distributional vectors are summaries of

[50]Incidentally, the idea of using corpus-harvested phrase vectors as targets of learning is not restricted to our functional approach. We could for example use minimum distance between composed and corpus-derived vectors from a training set as the criterion to choose the best settings for the weighted additive model.

the contexts in which a linguistic expression occurs, it is reasonable to expect that a vector directly constructed from corpus contexts is a good model of what we would like to learn by composition. If we want to define a composition function generating the distributional vector of *some coffee* from that of *coffee*, it stands to reason that we define a function that approximates the actual distributional vector of *some coffee*.

Of course, not many corpus-extracted phrases (and very few sentences) are common enough to find enough occurrences of them in a corpus to extract meaningful distributional vectors (that's why we want composition in the first place). However, we only need a few, reasonably frequent examples for each composition function to be learned by regression. In the transitive verb experiments of Grefenstette et al. (2013), good results were obtained with as little as 10 training examples per verb.

Corpus-extracted phrase vectors as targets of learning

Given the centrality of learning from phrase examples for our approach, we have collected various forms of empirical evidence that, at least for adjective-noun constructions (ANs) and DPs, phrase vectors directly extracted from the corpus make good semantic sense. It is thus reasonable to use them as our target of learning.

In Baroni and Zamparelli (2010), we have presented qualitative evidence that the nearest neighbours (the nearest vectors in semantic space) of the corpus-derived AN vectors are reasonable. See Table 4 (taken from Baroni and Zamparelli, 2010) for examples of nearest neighbours of nine randomly selected ANs.

A series of recent unpublished experiments provided quantitative support for the intuition about the good quality of corpus-derived ANs suggested by the data in Table 4. The experiments showed that the nearest neighbour in distributional semantic space of a corpus-derived AN vector is systematically picked by subjects as its most closely semantically related term over other plausible alternatives.[51] Subjects were presented with an AN (e.g., *serious decision*), the nearest neighbour of the corresponding distributional vector in semantic space (*crucial decision*), and another relevant term, for example the nearest neighbour of another AN sharing the same head noun (e.g., *wrong decision*, which is the nearest neighbour of *correct decision*). Subjects were asked which of the two terms they found most closely related in meaning to the target AN (without, of course, being aware of how the two terms

[51] These experiments and the *some N* nearest neighbour examples in Table 5 are based on DSMs similar to those described in Section 4.1 below.

bad luck	*electronic communities*	*historical map*
bad	electronic storage	topographical
bad weekend	electronic transmission	atlas
good spirit	purpose	historical material
important route	*nice girl*	*little war*
important transport	good girl	great war
important road	big girl	major war
major road	guy	small war
red cover	*special collection*	*young husband*
black cover	general collection	small son
hardback	small collection	small daughter
red label	archives	mistress

TABLE 4 The 3 nearest neighbours of the corpus-derived distributional vectors of 9 randomly selected ANs (from Baroni and Zamparelli, 2010).

were selected). Overall, 5,000 distinct triples were evaluated, with the alternative foils including, besides random terms, nearest neighbours of the adjective, of the noun, of ANs sharing the same noun and of ANs sharing the same adjective. In all settings, subjects showed a strong, statistically significant preference for the true nearest neighbour (in the running example, *crucial decision* was picked over *wrong decision* as the term most related to *serious decision*).

Note that, differently from the ML models, our approach to distributional function induction does not require harvesting vectors for grammatical words such as prepositions or determiners. Instead, we collect vectors for *phrases* that contain such words combined with content words. We do not extract a (presumably uninformative) vector from all contexts in which *some* occurs, but pairs of example vectors such as <*cats*, *some cats*> and <*coffee*, *some coffee*>. A choice of nearest neighbours of the corpus-harvested vectors for *some cats* and *some coffee* is presented in Table 5.

Note first in Table 5 how all the neighbours are intuitively semantically close to the target DPs, involving nouns from the same domain and mostly the same or a related quantifying determiner. Note moreover how the neighbours of the count usage of *some* in *some cats* are, consistently, other expressions involving counting of distinct individuals. The mass usage with *coffee*, on the other hand, tends to attract other constructions involving quantifying amounts of substances. It should be possible to learn, by regressing on training examples of this sort, that *some* has a different meaning when modifying a count or a mass noun, as illustrated in the toy **SOME** matrix in the Section 3.4 above.

some cats	*some coffee*
some dogs	some tea
most cats	some breakfast
most dogs	some dinner
many cats	some chocolate
many dogs	another bottle
most rabbits	some beer
some breeds	another drink
most animals	some cake
some horses	some toast
some babies	more beer

TABLE 5 A choice of nearest neighbours (among top 20) of the corpus-derived vectors for *some cats* and *some coffee*.

In Baroni et al. (2012), we have shown that corpus-harvested distributional vectors for DPs with a quantifying determiner contain enough information for a statistical algorithm to correctly learn and generalize the entailment status of pairs of DPs represented distributionally. For example, if we extract from the corpus distributional vectors for a few thousand entailing (*each dog*$\models$*some dog*) and non-entailing (*many cats*$\not\models$*all cats*) pairs, and we feed them as labeled training data to a machine learning program, the program is then able, given an arbitrary pair of DP vectors, to tell whether the pair is entailing or not, with accuracy significantly above chance. Generalization works even in the case in which the test pairs contain determiners that were not in the training data. That is, the program correctly predicts that, say, *several snakes*$\not\models$*every snake* even if it did not see any phrase containing *several* in the training data.

Further support for the hypothesis that corpus-harvested distributional vectors for phrases are high-quality examples of the composite meaning they represent come from Boleda et al. (2012b) and Turney (2012). Boleda and colleagues show that corpus-harvested vectors representing AN constructions instantiating different kinds of modification (intersective, subsective, intensional) display global patterns of similarity that reflect linguistic intuitions about adjectival modification (see also Section 4.3 below). Turney reports that corpus-harvested phrase vectors (which he calls "holistic" vectors) reach excellent performance when used in the task of finding the best single-word paraphrase for a noun phrase.

Together, these results suggest that, at least for simple phrases, we can indeed harvest meaningful examples of how we would want the out-

put of composition to look like directly from the corpus. The relative success of our method in predicting human intuitions about full sentences (see Section 4.2 below) suggests that meaningful training vectors can also be harvested for simple sentential constructions, since inducing representations for verbs (necessary to handle sentences) involves extracting example subject-verb and subject-verb-object vectors. Although the relevant techniques are introduced below, we discuss some issues raised by these "bare-bone" sentence vectors here, since they pertain to the general topic of the current section, namely the role played by corpus-extracted examples in our approach.

First, some have objected that our method might work for *simple* instantiations of a target construction, but how about complex ones? We might be right, the objection goes, that large corpora contain enough informative examples of *"spiders chase ladybugs"* to build a meaningful example vector for this bare-bone subject-verb-object construction. However, how would you ever expect to extract a meaningful corpus-based vector for "*sneaky black spiders quietly chase cute little ladybugs in the midnight garden*"? This objection forgets the very mechanisms of compositionality our entire framework rests upon, and confuses the corpus-extracted phrase vectors needed for learning (that only requires a small set of bare-bone instantiations of the target construction) with the vectors representing arbitrarily complex structures we can derive once our compositional system has been trained. After the system has been trained, a complex sentence like the one above can be constructed in steps by applying the relevant composition rules: recursive adjective modification to build *"sneaky black spiders"* and *"cute little ladybugs"* vectors, adverbial modification to derive a *quietly chase* tensor from *chase*, multiplication of the resulting transitive verb tensor by the object and subject vectors to derive the basic transitive sentence (see below), etc. Each of these rules can be trained from the simplest instantiations of the corresponding constructions, for which we should be able to find a sufficient number of training examples in the corpus: For example, the *chase* tensor will be learned from simple example subject-*chase*-object vectors such as the one for *"spiders chase ladybugs"*. There will never be the need to extract vectors directly from the corpus for complicated but composite structures such as the larger *"sneaky black spiders"* sentence above. It only makes sense to derive the latter compositionally.[52]

A related objection is that the corpus will contain few bare-bone

[52]Still, we do not want to deny that even for skeletal sentences with a subject-verb-object structure, we might incur into data sparseness problems. We briefly address the issue towards the end of this section.

sentences of the *"dogs bark"* or *"spiders chase ladybugs"* kind, that we need to learn verbs by regression, since real-life sentences are typically more complex than this (again, see below for the actual learning procedure). This objection overlooks the fact that a *"spiders chase ladybugs"* example vector can be extracted from sentences of *any* complexity, as long as they contain *spiders* as subject, *chase* as (main) verb and *ladybugs* as object, with all other lexical material in the sentence potentially treated as context for the target phrase. For example, if *"sneaky black spiders quietly chase cute little ladybugs in the midnight garden"* does occur in our training corpus, then during training we will treat it as a context in which *"spiders chase ladybugs"* occurs, and as evidence that it co-occurs with *sneaky, black, quietly, cute*, etc., which is precious information for constructing the corpus-based *"spiders chase ladybugs"* vector (see also the analogous *"boy kissed girl"* example we discussed at the beginning of Section 3).

We conclude our discussion of the corpus-extracted phrases we use in learning with some conjectures about the role of such example phrases once the compositional system has been trained. After learning, should we throw the example phrase vectors away, and prefer the compositional route in any case? To take a simple case, suppose we use the *red car* vector as training example to learn the *red* function. Given that we have extracted the vector during training, if we later need to use a distributional representation for *red car*, should we use the training vector directly extracted from the corpus or generate it anew by multiplying the estimated **RED** matrix by **car**? This is an open question.

Note that, although we use corpus-extracted phrase and sentence vectors to train distributional functions (and we just argued that they are of sufficient good quality to motivate this choice), it is not the case that corpus-extracted vectors (when available at all) are necessarily of a better quality than their composed counterparts. In Baroni and Zamparelli (2010), we have shown examples where the nearest neighbour of a composed AN vector is more reasonable than that of the corresponding corpus-derived vector. For example, the nearest neighbour of composed *special something* is *special thing*, that of corpus-derived *special something* is *little animal*; the nearest neighbour of composed *historical thing* is *historical reality*, the one of its corpus-derived counterpart is *different today*. We hypothesize that, in such cases, the corpus did not contain sufficient information to create a good representation of the phrase (e.g., because the phrase is too rare). Thus, applying the distributional adjective function, that has been trained on many more examples, to the noun vector produces a better approximation to the meaning of the phrase than the one we get out of direct evidence (in the

limit, the claim becomes trivial; a corpus-extracted vector representing a phrase that never occurs in the corpus, that is, a vector of 0s, will certainly be worse than its compositionally derived counterpart).

On the other hand, at least in certain cases both corpus-derived and composed vectors have a role to play. An obvious case is that of idioms.[53] A corpus-derived vector for *red herring* will probably have neighbours related to its "misleading cue" sense. On the other hand, the output of **RED** $\times$ **herring** will probably be a vector for the literal colored-fish meaning. An English speaker will be aware of the idiom, but she can also compositionally understand *red herring* as referring to the colored fish (indeed, the general consensus in psycholinguistics is that whenever an idiom is encountered, it is also automatically processed via a compositional route; see Cacciari, 2012). By providing (where possible) both directly corpus-derived and composed representations of the same phrases, our approach can capture the same dichotomy.[54]

Learning higher-order tensors

Having shown how regression can be used to estimate the weights of matrices (second-order tensors) and argued that corpus-extracted examples of the relevant output constructions are a suitable target for regression learning, we turn now to how this approach can be extended to functions of more than one argument. Recall that such functions are encoded in higher-order tensors (an n-argument function is encoded in an $n+1$-th order tensor), and thus the goal of regression is to estimate the values to be stored in the cells of such tensors.

In particular, when a function returns another function as output, e.g. when it acts on a vector and generates a matrix, we need to apply a two-step regression learning method, inducing representations of example matrices in a first round of regressions, and then using regression again to learn the higher-order function.[55]

Grefenstette et al. (2013) illustrated this for transitive verbs. An intransitive verb is naturally modeled as a VP, that is, a function from

[53]Note that we are speaking here of completely opaque idioms of the *red herring* and *kick the bucket* sort. We expect corpus-derived examples to provide enough evidence for our approach to pick up any systematic semi-lexicalized or metaphorical pattern, such as the political usage of the adjective *red* to refer to socialism. Indeed, as mentioned above, the compositional system for AN meanings of Baroni and Zamparelli (2010) performed particularly well with highly polysemous adjectives.

[54]Relatedly, automatically scoring the degree of semantic opaqueness of a phrase has recently been proposed as a benchmarking task for distributional semantic models (Biemann and Giesbrecht, 2011).

[55]Georgiana Dinu (p.c.) has developed a method to estimate higher-order tensors in just one step: However, the method requires the same training data as the multi-step method, that is conceptually simpler.

a DP (the subject) to a sentence. A transitive verb, then, is a function from a DP (the object) to a VP, i.e., to a function from DPs to sentences.[56] Let's go back to our toy DP semantic space of 4 dimensions (as in the "some" example of the previous subsection) and let's take sentences to live in 2-dimensional space. Hence, a VP is a 2×4 matrix. For example, both *jump* and *"eat cake"* are matrices of this shape.[57] A transitive verb such as *eat* is then a third-order $(2 \times 4) \times 4$ tensor, that takes an object DP vector (e.g., *cake*) to return the corresponding 2×4 VP matrix (*"eat cake"*).

To learn the weights in such tensor, we first use regression to obtain examples of matrices representing verb-object constructions with a specific verb. These matrices are estimated from corpus-extracted examples of <*subject*, *subject verb object*> vector pairs (picking, of course, subject-verb-object structures that occur with a certain frequency in the corpus, in order to be able to extract meaningful distributional vectors for them). After estimating a suitable number of such matrices for a variety of objects of the same verb (e.g., *"eat cake"*, *"eat meat"*, *"eat snacks"*), we use pairs of corpus-derived object vectors and the corresponding verb-object matrices estimated in the first step as input-output examples in a second regression step, where we determine the verb tensor components. The two-step estimation procedure is schematically illustrated for *eat* in Figure 3 (from Grefenstette et al., 2013). Of course, after the *eat* tensor has been estimated, it can be used to generate transitive sentences with subjects and objects that were not used in the training phase.

Next, let us consider the most complex case, that is, that of a higher-order function that takes other functions both as input and as output. In this case, we will first use regression to construct examples of both the input and output functions (e.g., matrices), and then use these examples to train the higher-order tensor we are interested in. Let's go back to the example of the relative pronoun *which* that we discussed in Section 3.4 above. We concluded there that *which*, as a function from VPs onto noun modifiers, is a fourth-order tensor mapping input VP matrices onto output noun-modifier matrices. In particular, in our toy lexicon we took *which* to be a third-order tensor with shape $(3 \times 3) \times (2 \times 4)$ mapping from 2×4 input VP matrices to 3×3 output noun-modifier matrices.

In this case, the first training step will generate examples of both the

[56] In the conclusion, we will come back to some important issues pertaining to this treatment of verbs, such as how to handle changes in argument structure.

[57] Like Grefenstette et al. (2013), we ignore for purposes of all examples discussed in this subsection the inflection of the verb and number of nouns and DPs.

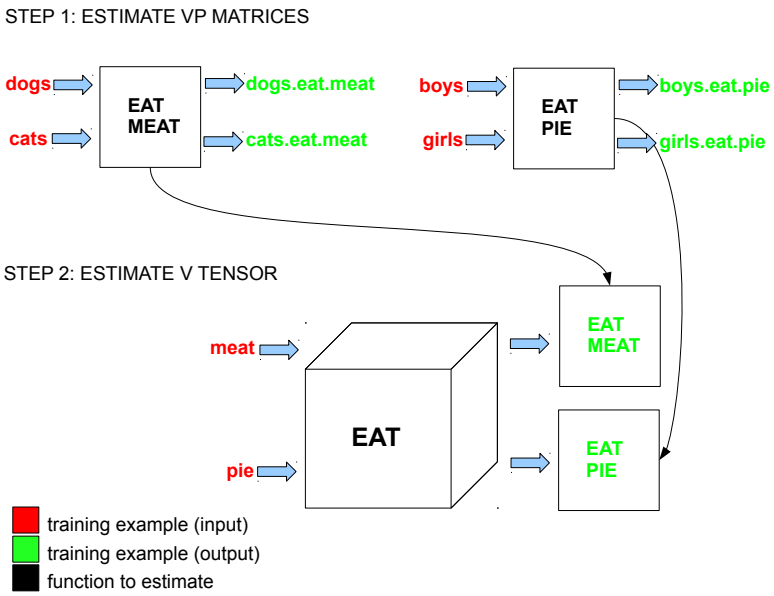


FIGURE 3 Estimating a tensor for *eat* in two steps. We first estimate matrices for *eat meat*, *eat pie*, etc., by regression on input subject and output subject-*eat*-object vector pairs (directly extracted from the corpus). We then estimate the tensor for *eat* by regression with the matrices estimated in the first step as output examples, and the vectors for the corresponding objects as input examples (from Grefenstette et al., 2013).

input and the output matrices. For the input VPs, training will proceed exactly like with transitive verbs (see Figure 3) to derive a set of VP matrices (2×4), that in this case can contain different verbs and be both transitive and intransitive (*"eat meat"*, *"chase cats"*, *sing*, *jump*, ...). The training of output noun modifiers (3×3) is similar, but here the corpus-extracted example vectors will be pairs of <*noun*, *noun which VP*>. For example, the *"which eat meat"* matrix will be trained from corpus-extracted vectors of pairs such as <*dog*, *dog which eats meat*>, <*cat*, *cat which eats meat*>, etc. In the second phase, we estimate the *which* tensor by optimizing, via regression, the mappings between VP-matched input-output matrix pairs trained in the first phase, e.g., <*eat meat*, *which eat meat*>, <*sing*, *which sing*>, and so on.

Two problems loom ahead: data scarcity and computational load. Consider the first. As the complexity of the structures to be learned

grows, it becomes increasingly difficult to find a sufficient number of frequent examples of their inputs and outputs in order to obtain meaningful training vectors from the corpus. In pursuing our regression-based program for learning compositional semantics, it will thus be crucial to devise ways to harvest and optimally exploit high-quality training examples for all structures of interest. This might involve, on the one hand, using regression methods that can learn successfully from very few examples, and on the other, coming up with ways to extend the training sets exploiting similarities between linguistic expressions to 'share' training examples across distributional functions. Intuitively, good training examples for the *"which eat pie"* matrix could also be recycled as training examples when learning *"which eat cake"*. We envision the use of clustering methods to discover when two forms are sufficiently close to be pooled together in the training phase.

A second problem is computing power and storage needs. Given realistically-sized input vectors, the number of components to estimate in the corresponding higher-order tensors is humongous. If we assume (like Grefenstette et al., 2013 did, see Section 4.1 below) that nouns, DPs and sentences live in 300-dimensional spaces, a transitive verb is a $(300 \times 300) \times 300$ tensor, that is, it contains 27 million components. A relative pronoun, being a $(300 \times 300) \times (300 \times 300)$ tensor, contains 8.1 billion components. Luckily, there aren't as many relative pronouns as there are transitive verbs, since structures of this size are pushing the boundary of what can be stored and manipulated with reasonable efficiency given our (and we suppose most linguists') computational power. However, a somewhat technical but important consideration must be made here. The giant tensor is derived from training examples of much smaller sizes and often correlated with each other. Hence, by using algorithms that might exploit similarities in input and output components to effectively reduce the dimensionality of the problem, there will be a lot of "redundancy" in its cells. Furthermore, it will be probably possible to express the values they contain as weighted combinations of a much smaller set of vectors. Thus, a careful implementation of both learning and generalized matrix-by-vector product application might be able to sidestep some of the worse computational issues.

Learning functions that are not triggered by words

Since our learning procedure only requires examples of the input and output to a composition function, our system can also be extended to learning composition processes where the functional element is not an autonomous word. For example, the semantics of an affix such as *-ment* can be learned from training pairs such as <*contain*, *contain-*

ment>, <*endorse*, *endorsement*>, etc. (the derivation of *Italian* from *Italy* discussed in Section 3.4 above works analogously).[58] In the ML models, on the other hand, one would need to extract a vector for *-ment* or *-(i)an*, which is very problematic.

Actually, composition is not even constrained to be associated to any phonological material at all. A simple case is that of "null determiners", needed to account for the fact that *dogs* and *meat* are nouns, but in *"dogs eat meat"* they are used as full noun phrases (DPs, in our notation). Again, this is not a problem for our approach, where we can train a null determiner function that takes as input noun vectors (extracted from all contexts in which the nouns occur) and, as output, vectors for the same nouns constructed from those contexts only where they are used as full DPs.[59] A similar strategy can be followed for the "invisible" relative pronoun in constructions such as *"the meat dogs eat (is very fat)*". We do not see how composition triggered by phonologically empty elements (equivalently: purely structural configurations) could be handled by the ML approaches.

Of course, for such cases to work properly within our framework, we will need to make sure that the relevant rules are automatically triggered in the appropriate contexts by specific structures produced by the syntactic parser.

3.6 Syntax-semantics interface

So far we have presented the general intuitions and the technical aspects behind our proposal on the semantic modeling of natural language expressions. Now, we will look at how our semantic model interacts with syntactic analysis to scale up to account for sentence structure. We base our proposal on Montague's lessons and on the type-logical view of the syntax-semantics interface that has been developed starting from his Universal Grammar (Montague 1970b). Following this tradition, we will adopt Categorial Grammar (CG) to account for syntactic constructions and employ the formal techniques of the type-logical view to define a tight connection between syntax and semantics. Note that the same syntactic structures can also be used as the basis to construct standard referential representation of meaning, in a parallel distributional/referential approach to semantics.

To capture the relation between the syntax and the semantic level,

[58] This approach to the semantics of affixes was in fact already proposed in Guevara (2009).

[59] In the fragment of grammar to be presented next, we instead adopt the simpler and less linguistically informed strategy to treat bare plurals, such as *dogs* in *"dogs bark"*, as primitive elements —corpus-extracted vectors— of category DP.

the type-logical view uses the following steps: define an atomic set of syntactic categories and of semantic types based on which functional syntactic categories and complex semantic types are built; define a recursive mapping between the syntactic categories and the semantic types; assign typed meaning representations to each lexical entry, where the types are those of the corresponding domain of interpretation; assign to the lexical entries syntactic categories that correspond to their semantic types. This procedure allows one to proceed in parallel in the composition of the syntactic and semantic constructions.

Besides this theoretical advantage, employing a CG framework has practical benefits, because of the existence of a fast and wide-coverage syntactic parser, namely C&C parser(Clark and Curran, 2007), based on (Combinatory) Categorial Grammar(Steedman, 2000). This parser is also integrated with Boxer, a system that builds a referential semantic tier using Discourse Representation Structures (Curran et al., 2007), thus allowing us to maintain the same large-scale approach that characterizes lexical DSMs in our compositional component, and providing a concrete infrastructure for the possibility of parallel distributional/referential representations built from the same semantic structures. Of course, other lexicalized formal grammars could also be considered; CG is just the one that might allow the integration in the most straightforward way.[60]

In denotational semantics the semantic types give the type of the domain of denotation (e.g., the domain of entities D_e, containing the denotation of proper names, or the domain of functions from entities to truth values, $D_{(e\to t)}$, used for intransitive verbs, verb phrases, and nouns). In a DSM, we take domains to stand for the distributional semantic spaces in which the expressions live, and, as in denotational semantics, we take these semantic spaces to be typed. The type records the shape of the tensors in the space (plus the associated index labels) as discussed in Sections 3.3 and 3.4. We mark atomic types with subscript indices standing for the shape of the items in the corresponding semantic space. Suppose, for example, that the noun *dog* lives in a 10,000-dimensional nominal space; then, its distributional representation will be a vector in this space and will have type $C_{n_{10000}}$, where

[60] Our choice of CG (and in particular Combinatory CG) over Dependency Grammar, another widely-used parsing framework in computational linguistics (Kübler et al., 2009, Mel'chuk, 1987), was also motivated by the fact that dependency-parsed structures are not binary, and do not make explicit the mutual scope of modifiers (in a parse of *"the hypothetical high percentage of voters"*, *the*, *hypothetical* and *high* would all be dependent on *percentage*, without any indication but word order that *hypothetical* scopes over *high*, but not vice-versa.)

we use C to remind ourselves that this is a type based on Contextual information, the subscript n points to the $n$ominal space (characterized by a specific index-to-labels mapping) and the subsubscript to the dimensionality of the space.

Similarly, functional types will correspond to space mappings (linear transformations) and will be represented by different tensors: matrices (that is, second order tensors) for first order 1-argument functions, third order tensors for first order 2-argument functions, etc. In general, we assume that words of different syntactic categories live in different semantic spaces. Note that we are *not* assuming that the complexity of the formal semantic type must correspond to a corresponding complexity in the shape and order of the corresponding DSM structure. A salient case is that of nominals. In Montague Grammar proper names are of type e (entities), but quantified DPs are of type $(e \rightarrow t) \rightarrow t$ (sets of properties). In our current experiments we do not cover proper names, due to the ambiguity issues pointed out in Section 2.6, but we treat quantificational DPs as first order tensors, i.e., vectors, albeit potentially living in a different space from that of nouns.[61] Note, moreover, that in denotational semantics the domain of interpretation is partially ordered by the inclusion relation ($\subseteq$) holding within the denotational sets. It is on the basis of this order that the logical entailment of phrases and sentences is computed; in our case semantic similarities are computed based on similarities of tensors living in the same space, as discussed in Section 3.4. Thanks to this notion of "typed similarity", once the whole framework is implemented, we believe that we can arrive to draw richer (or more "natural") kinds of reasoning based on distributional representations plus logical entailment, rather than on logical entailment alone.

Given the programmatic nature of this paper, we have touched upon many constructions for which a full computationally viable analysis is still underspecified. However, following the Montagovian tradition, we also want to give the reader a precise idea of how our system could handle a fragment of English. In the lexicalized view of CG, this means

[61]This might seem untenable in a view of semantics in which determiners are diadic functions over a restrictor and a predicate. As we have seen in Section 3.4, our determiners are unary functions over Ns, while VPs are unary functions over DPs. This approach is not feasible in Montague Grammar because the operation VPs perform on their subject is extremely simple: set membership. Thus, *John runs* will be true in a model M iff it is true that j' $\in [\![\text{run}]\!]^M$. In our approach, a VP function can be far more sophisticated (like the determiner, it takes an input vector, but of course it does not return just a binary value, "true" or "false"). So, while it is an empirical question whether our current approach is tenable in the long run, we do not see strong theoretical motivations against it.

defining the lexicon out of which sentences of the fragment are built.

The lexicon we chose is a representative sample, in that it includes expressions with primitive types, functions over primitive types and functions over functions.

Most compositions in natural language are 'local', in that they take place between adjacent expressions, and involve first order n-functional categories. However, all human languages have instances of non-local dependencies. In some of these, the elements that should combine are not adjacent, but still within the same tensed sentence ("clause-bound dependencies"); in others, the dependency is between elements that can be separated by arbitrary amount of material ("long-distance dependencies"). In our fragment we will consider a single, syntactically simple but semantically challenging case of clause-bound non-local dependency, that of relative clauses with object gap and an overt relative pronoun.

In an object relative, like *"a cat which dogs chase runs away"*, the noun *cat* plays the double role of being the subject of the main clause and the object of the relative clause. As an object, it depends on the verb *chase* to which it is not juxtaposed. From a formal semantic viewpoint, the relative pronoun is represented by the lambda expression in (7) which intersects two properties, e.g., $[\![\text{cat}]\!] \cap [\![\text{dogs chase}]\!]$ (as we have seen in Section 3.4), (8-a) gives the type for *which* and (8-b) spells out how *which* combines with the property denoted by the gapped clause it C-commands[62] (*"dogs chase"*), then with the property denoted by *cats* in a sister node, to yield the property denoted by *"cat which dogs chase"*.

(7) $\lambda X_{(e\to t)}.\lambda Y_{(e\to t)}.\lambda x_e[X(x) \wedge Y(x)]$

(8) a. $\text{which} \in (e \to t) \to ((e \to t) \to (e \to t))$
b. $(\textbf{which}([\![\text{dogs chase}]\!])([\![\text{cats}]\!])$

Now that we have laid the ground, we can start introducing the syntactic categories, the semantic types, and the fragment we will be dealing with.

We use small letters (e.g., a) for atomic syntactic categories, capital letters (e.g., A and B) for complex syntactic categories, and `Type` for the function mapping syntactic categories to semantic types:[63]

[62] A node N *C-commands* another node M if it does not dominate it nor it is dominated by, and the first branching node that dominates N also dominates M (Reinhart, 1976).

[63] We are adopting Steedman's (2000) CG notation.

$$\begin{array}{lcl} \texttt{Type}(a) & = & C_a \text{ (for } a \text{ atomic)} \\ \texttt{Type}(B\backslash A) = \texttt{Type}(B/A) & = & C_A \rightarrow C_B \end{array}$$

N (noun), DP (determiner phrase) and S (sentence) are our atomic syntactic categories, mapped to indexed types as follows: $\texttt{Type}(N) = C_{n_i}, \texttt{Type}(DP) = C_{dp_j}, \texttt{Type}(S) = C_{s_k}$. The types of the complex categories are obtained by the definition above. We use a fragment of English whose vocabulary consists only of words in the syntactic categories listed in Table 6, which are representative of the varieties of functional categories we have discussed above. For sake of clarity, in the table, next to the syntactic category, we indicate both the corresponding distributional semantic type, as well as the shape of the corresponding distributional representation.

Relative pronouns (RelPr) in subject or object relatives should ideally receive the same syntactic category in CG. This can be done using other connectives besides the traditional functional ones (\ and /), but since our interest is in the syntax-semantics interface rather than in syntactic issues per se, we adopt the easiest CG solution and consider two syntactic categories: $(N\backslash N)/(S\backslash DP)$ for subject gap and $(N\backslash N)/(S/DP)$ for object gap, both mapping to the same semantic type.

While many constructions are not captured in the fragment, given that we can harvest thousands of distributional representations for lexical items from corpora, the fragment actually covers a huge amount of sentences, including the following:

(9) The sneaky black spiders with long hairy legs that the boys love ate the cute little guinea-pig that the girls bought.

Expressions are composed by using the CG syntactic tree (or derivation) as the backbone and by defining a correspondence between syntactic and semantic rules. In Montague Grammar, two main types of semantic rules are used: function application and abstraction. The syntactic correspondence of abstraction is also used in the logic version of CG, namely in the Lambek calculi (Lambek, 1961, Moortgat, 1997), and a restricted instance of it (type raising) is also present in CCG, the combinatory version of CG (Steedman, 2000). In short, abstraction is used mostly for two cases: non-local dependency and inverse scope. In the current study, we are not interested in scope ambiguities since we believe that they are a challenge for syntacticians rather than semanticists; once the grammar provides the right representation for an ambiguous sentence (i.e., a disambiguated 'logical form') the semantic operations should be able to compute the proper meaning straightfor-

Lexicon			
Syn Cat	CG Cat	Semantic Type	Tensor shape
N	N	C_{n_i}	I vector (1st ord.)
NNS[a]	DP	C_{dp_j}	J vector (1st ord.)
ADJ	N/N	$C_{n_i} \to C_{n_i}$	$I \times I$ matrix (2nd ord.)
DET	DP/N	$C_{n_i} \to C_{dp_j}$	$J \times I$ matrix (2nd ord.)
IV	$S\backslash DP$	$C_{dp_j} \to C_{s_k}$	$K \times J$ matrix (2nd ord.)
TV	$(S\backslash DP)/DP$	$C_{dp_j} \to (C_{dp_j} \to C_{s_k})$	$(K \times J) \times J$ (3rd ord.)
Pre	$(N\backslash N)/DP$	$C_{dp_j} \to (C_{n_i} \to C_{n_i})$	$(I \times I) \times J$ (3rd ord.)
CONJ	$(N\backslash N)/N$	$C_{n_i} \to (C_{n_i} \to C_{n_i})$	$(I \times I) \times I$ (3rd ord.)
CONJ	$(DP\backslash DP)/DP$	$C_{dp_j} \to (C_{dp_j} \to C_{dp_j})$	$(J \times J) \times J$ (3rd ord.)
RelPr	$(N\backslash N)/(S\backslash DP)$ $(N\backslash N)/(S/DP)$	$(C_{dp_j} \to C_{s_k}) \to (C_{n_i} \to C_{n_i})$	$(I \times I) \times (K \times J)$ (higher ord.)

[a]Plural nouns phrases. In the fragment, we will be assuming that they can be directly mapped onto full DPs ("Bare plurals").

TABLE 6 Syntax-Semantics interface for a fragment of English

wardly. Hence, in what follows we will only look at abstraction cases motivated by non-local dependencies, and in particular at the case of relative pronouns that extract an object from relative clause.

Local dependencies

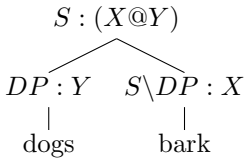
Since in natural language function-argument order matters, CG has two function application rules: backward (when the argument is on the left of its function) and forward (when the argument is on the right of its function.)

On the DSM side, the sentences in the fragment we are considering require the following function application cases.

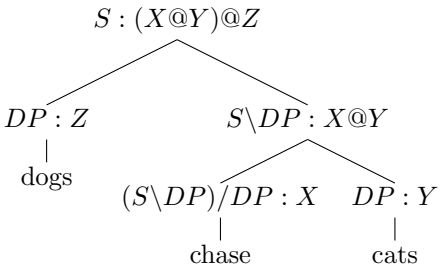
(10)
(a) A matrix (2nd order tensor) composes with a vector (ADJ N e.g., *"red dog"*; DET N e.g., *"the dog"*; DP IV e.g., *"the dog barks"*, *"dogs bark"*);
(b) A 3rd order tensor composes with two vectors (DP TV DP, *"dogs chase cats"*; N Pre DP, *"dog with tails"*; DP CONJ DP, *"dogs and cats"*);
(c) A higher-order tensor composes with a matrix ((c1) Rel IV, e.g., *"which barks"*; Rel TV DP, *"which chases cats"*; and (c2) Rel DP TV, *"which dogs chase"*)

To emphasize the relation with the Montagovian framework, we begin by building a labeled syntactic tree. The labels record the operational steps and are therefore called *proof terms* or "derivation terms". A proof term can then be replaced with the appropriate corresponding semantic representation. Under the classic Montagovian view, it will be replaced by λ-terms standing for the meaning of the words (like the lambda terms discussed before for the relative pronoun). In a Continuation Semantics, it would be replaced by λ-terms that take the context into account (see Bernardi and Moortgat, 2010, Barker and Shan, 2006). In our setting, we replace proof terms with the corresponding tensors. To help reading the proof terms, we use the @ symbol to indicate the application of a function to an argument ($f@a$). For instance, when parsing the expressions *"dogs bark"*, *"dogs chase cats"* and *"which chase cats"*, CG produces the structures and terms in the trees in (11) and (12).

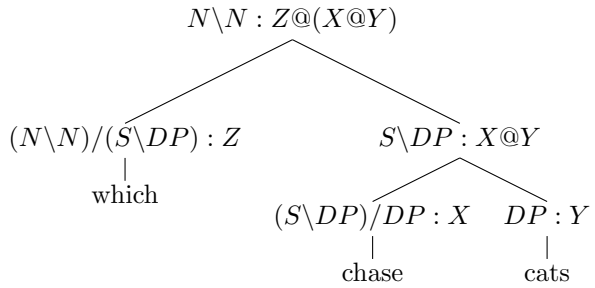
(11) a.



b.



(12)



We replace the variables with the corresponding DSM representations obtained from corpora and compute the vectors representing the sentences. In particular, in (11-a) the labels X and Y are replaced with the matrix **BARK** and the vector **dogs**, respectively, giving $\mathbf{BARK} \times \mathbf{dogs}$; whereas in (11-b) X is replaced by the 3rd order tensor representing the meaning of *chase*, and Y and Z are replaced with the vectors representing the meaning of *dogs* and *cats*, respectively. Hence, we obtain $(\mathcal{CHASE} \times \mathbf{cats}) \times \mathbf{dogs}$. Similarly for (12), where we obtain $\mathcal{WHICH} \times (\mathcal{CHASE} \times \mathbf{cats})$. Once we have built this sort of representation of the sentence, we can compute its meaning using the generalized matrix-by-vector product introduced in Section 3.3.

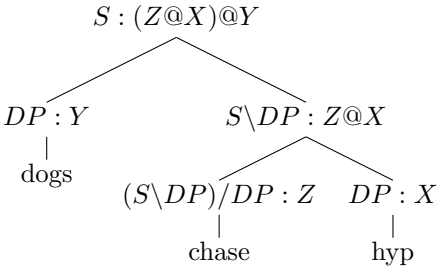
Crucially, our vectorial representations have been built on the output of a CG parse of the sentence, a representation commonly used in formal semantics as input to build the logical form of sentences compositionally (van Benthem, 1986, Moortgat, 1997, Steedman, 2000). Indeed, as we have mentioned above, the same CCG parser that produces the trees we use for our compositional operations is integrated with the logic-based Boxer system (Curran et al., 2007). Thus, we offer a clean and practical implementation of the parallel construction of logical and distributional semantic representations of sentences. A representation that, as we are about to see, also extends to the non-local dependency case we handle in our fragment.

Non-local dependencies

So far we have been dealing with local dependencies: the dependent element was always juxtaposed to the head. In CG terms, we can say that the function always found its syntactic argument next to it. We will consider the case of non-local dependency that is part of our fragment of English, namely the one of relative clauses with object gap and an overt relative pronoun, introduced above. As an example of this clause-bound dependency, we consider the sentence *"a cat which dogs chase runs away"* in which the object position of *chase* is missing, and *dog* plays the role of the object of the relative clause verb *chase* while being the subject of the main clause verb *runs*.

Besides the forward and backward rules used so far, CG (or more exactly, the logical version of it) is endowed with another kind of rule. We have seen that the forward and backward rules correspond to the application of a function to an argument; the third rule type corresponds to the abstraction of a variable from a term, in other words, to *hypothetical reasoning*. In the parsed linguistic structure we are considering, the verb *chase* requires an object (a DP) to its right, but no object is provided next to it.The parser, reasoning by hypothesis, assumes there is a DP juxtaposed to *chase* and composes the verb with this hypothetical DP, then it continues its composition process by composing the verb phrase thus obtained with the subject *dogs* found to its left. These steps are represented in (13), where we mark the hypothetical object by *hyp*, and we label the nodes with proof terms as done so far. The clause with the hypothetical object, *"dogs chase hyp"*, is represented by the proof term $(Z@X)@Y$.

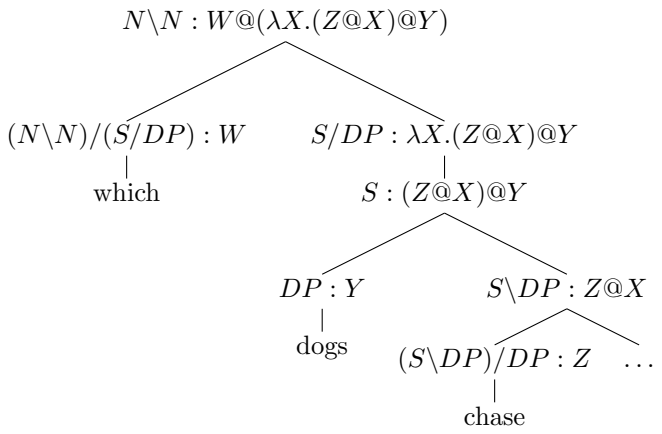
(13)



As explained above, in denotational semantics the relative pronoun *which* acts as a modifier of the noun *cat*, restricting the set of cats to those that dogs chase. Hence, it composes first with the property *"dogs chase"* and then with the noun *cat*. As we have already seen, it has semantic type $(e \to t) \to ((e \to t) \to t)$—a function from properties to properties to truth values. The CG category that expresses the same be-

haviour at the syntactic level is the higher-order two-argument category (N\N)/(S/DP) which requires a sentence missing a DP on the rightmost position to return the category N\N. Hence, the parser encounters a category mismatch: It has the task of composing (N\N)/(S/DP) (*which*) with the tree of category S corresponding to *"dogs chase hyp"*. The tree of category S, however, contains an hypothesis of category DP—it would be a sentence if a DP had been provided. The parser can now withdraw the hypothetical DP, as illustrated in (14), and build the tree of category S/DP. The rule that allows this step is the one-branch rule encoding hypothetical reasoning. The lambda calculus goes step by step with this hypothetical reasoning process. Besides the function application rules we have used so far, it consists of the *abstraction* rule that abstracts from the term $(Z@X)@Y$ (namely the term assigned to the S tree –hence, a term of type t), the variable X assigned to the hypothetical DP (hence, a term of type e), building the lambda term $\lambda X.(Z@X)@Y$ (a term of type $(e \rightarrow t)$). The next step is again the application of a function (W of type $(e \rightarrow t) \rightarrow ((e \rightarrow t) \rightarrow t)$) to an argument (the lambda term of type $(e \rightarrow t)$ we have just built).

(14)



How do we deal with the lambda abstraction step if we replace proof terms with DSM representations instead of their denotational counterparts? We sidestep this question by suggesting a solution that implicitly employs the rule of *associativity*. Through this rule, when the parser fails to find the DP object on the right of the relative clause verb, it can replace the category of the verb $(S\backslash DP)/DP$ with $(S/DP)\backslash DP$ since the two categories are equivalent *modulo* associativity.[64] This change of

[64] Associativity causes over-generation problems. However, its application could be controlled by employing the multi-modal version of CG (Moortgat, 1997). Since our

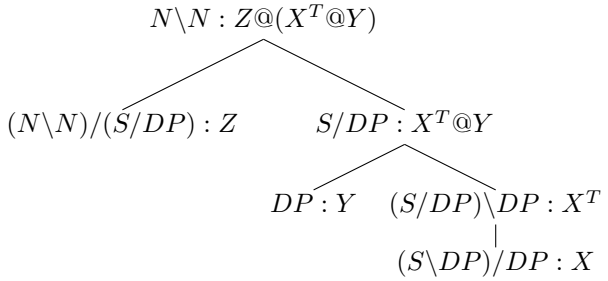
the category allows the verb to combine first with the subject on its left to return a predicate that looks for the object to its right. As a consequence, the tree corresponding to *"dog chase"* is already of the correct S/DP category to be taken as argument by the relative pronoun *which*.

One reason for suggesting this solution is that we have a nice, ready-to-use solution on the tensor composition side. The proof term recording the steps of the tree structure assigned to *which dogs chase* (15) mostly consists of function applications of the kinds discussed so far; the 'one branch step', which changes the category of the transitive verb, is the only new one. This syntactic step has a natural distributional semantic counterpart in the tensor transposition operation introduced at the end of Section 3.3, which gives a general procedure to generate a transposed tensor such that:

$$(\mathcal{T} \times \mathbf{v}) \times \mathbf{w} = (\mathcal{T}^T \times \mathbf{w}) \times \mathbf{v}$$

This comes in very handy, since it allows us to transform a (pre-trained) transitive verb tensor that would normally be multiplied by an object and then a subject into the transposed form, that can take the subject first, and the object later, producing the same result. In the tree, we represent this semantic rule as taking the term X and yielding the term X^T. Now we can replace the proof term with the actual distributional representation, obtaining $\mathcal{WHICH} \times (\mathcal{CHASE}^T \times$ **dogs**). This can later modify the vector representing *cat*.

(15)



Among the challenging class of non-local phenomena, we have restricted our attention to relative clauses involving object abstraction from its outmost position to the main sentence that immediately dominates the relative, a phenomenon that, as far as we are aware, is already more linguistically complex than anything studied so far by the compositional

focus is on the composition of the distributional semantic representations involved in such constructions, we will overlook the syntactic issues. Our semantic analysis or a close variation of it could be connected to different syntactic proposals in the literature.

distributional semantics research community. We will of course have to verify whether our current solution, besides being conceptually simple, is also supported by empirical evidence, and whether it can be extended to other non-local dependencies. Note that in this transposition-based approach the relative clause changes its shape as a result of abstraction: it goes from being a vector to being a matrix, so the heads that could subcategorize for it in one shape cannot straightforwardly do so in the other. This means, for instance, that the *that* in (16-a) cannot have the same type as the one in (16-b) (the first would take a sentence vector, the second a matrix, from the transposed verb). The grammatical rules used so far (function application, abstraction and associativity) derive the type-shifting rule that maps the first *that* into the second (syntactically, from an (S/S) to an $((S/DP)/(S/DP))$). They derive also the rules needed for the verb *claim* in (16-b) to take a clause missing the object instead of the complete sentence it takes in (16-a). The derived shifting of types would allow *who* in (16-b) to work just as *which* in (14). We have not investigated whether for these type-shifting rules there exists a ready transformation as for the case discussed above, if this is the case, we will consider extending our fragment to these more complex long-distance dependencies.[65]

(16) a. People claim that [girls love this boy].
b. A boy who people claim that [girls love __] arrived.

It is also possible that a general solution for long-distance cases will come not from the pure DSM composition and transposition rules, but by reconstructing the arguments in their base position and calculating 'localized' vectors to be used as modifiers. For instance, the long distance relative in (16-b) could be rendered by first computing the vector for those tokens of the lemma *boy* which appear in the context of (17) (and possibly, related sentences, to ease data sparcity).

(17) People claim that girls love DET **boy(s)**

Next, this localized vector would be combined (e.g., by intersective modification) with the general vector for *boy* in the sentence *a boy arrived*, yielding an approximation of the meaning of (16-b). Exploring

[65] Although our transposition approach could be a solution for relatives with a gap in an *argument* position, as far as we can see, it would not work when all the arguments of the verb are saturated. For instance, it would be insufficient to deal with, e.g., *the way in which Marco runs*, where the pronoun is extracted from a manner modifier. A conceivable solution would be to take modifiers to be additional arguments of the verb. A verb with a missing "manner argument" would in this case be transposable, though its dimensionality would increase.

this and similar mechanisms will be a task for future research.

4 Current empirical support for the functional approach to distributional semantics

The aim of this paper is to sketch a program for compositional distributional semantics based on function application and to show how it could be applied to an extended fragment of English that includes grammatical words. But there is already evidence that the approach is empirically feasible, at least when it comes to modeling phrases and simple sentences made of content words. In this section, we briefly review the relevant empirical work.

4.1 DSM implementation and composition methods

We start by providing a succinct summary of the technical aspects of the compositional DSMs used in the experiments that follow; please refer to the original publications for more detail. These DSMs are harvested from a corpus of almost 3 billion running words containing all the documents from the British National Corpus (see Section 2.1 above), a dump of the English Wikipedia and a large sample of other Web documents. The distributional space in which the vectors live is defined by co-occurrence within sentence boundaries with the 10,000 most common lemmatized content words in the corpora (Baroni and Zamparelli, 2010, and Vecchi et al., 2011, include nouns, adjectives and verbs; Grefenstette et al., 2013, and Boleda et al., 2012b, also include adverbs). The raw counts are transformed into Mutual-Information-based association scores (see Section 2.1) and the 10,000 original components are compressed into 300 with dimensionality reduction techniques (again, refer to Section 2.1 above).[66]

Across all experiments, the functional model we are proposing is contrasted with the (unweighted) additive and multiplicative models of Mitchell and Lapata. Vecchi et al. (2011) and Boleda et al. (2012b) also test the model proposed by Guevara (2010) (see Section 6 below). Boleda et al. (2012b) also implement the dilation model of Mitchell and Lapata (2010) (see footnote 24 above). Grefenstette et al. (2013) include, in the transitive tests, the Kronecker model that was the best performer in Grefenstette and Sadrzadeh (2011b) (and that is not defined for intransitive verbs).

[66]For technical reasons, the multiplicative model requires a different method of dimensionality reduction than the other models (Grefenstette et al., 2013), or no reduction at all (the other studies except Vecchi et al., 2011).

4.2 Intransitive and transitive sentences

Mitchell and Lapata (2008) introduced a data set of similarity judgments for 120 intransitive sentence pairs. The pairs, rated by 49 subjects on a 1-7 scale, were constructed to maximize the similarity or dissimilarity between the sentences by exploiting verb ambiguity. For example, one of the pairs that received high similarity ratings was *"the child strayed"–"the child roamed"*; one pair with low ratings was *"the child strayed"–"the child digressed"*. By replacing *child* with *discussion*, the opposite intuitions are obtained. As already discussed in Section 2.3 above, these stimuli tap into systematic polysemy, and in particular co-compositional aspects of verb meaning. All sentences were presented with the definite determiner and in simple past format to the subjects, but the constant determiner and tense are ignored by the composition models.[67] Grefenstette et al. (2013) implemented the approach to composition with intransitive verbs that we sketched in Section 3.4 above. Given the concrete DSM we described in the previous subsection, in which all vectors live in 300-dimensional space, they derived 300-component sentence vectors by multiplying 300×300 matrices representing the verbs by the 300-component vectors representing the subject nouns (verb matrices were trained on example input and output vector pairs as illustrated in Section 3.5 above). Performance was measured by the correlation of the cosines (see Section 2 above) between sentence vectors produced by a model with the subjects' rating for the corresponding sentence pairs (if the model is good, when subjects assign high ratings the cosine should be high, indicating high sentence similarity, and *vice versa* for low ratings). Correlation scores range from 0 for no correlation whatsoever to 1 for perfect correlation (or -1 for a perfect *inverse* correlation). To put things into perspective, the inter-subject correlation is of 0.40; and measuring the similarity between corpus-extracted vectors representing the verbs only (thus ignoring the contribution of subject nouns to meaning) achieves a correlation of 0.06. The functional model outperformed all other composition methods with a correlation of 0.23 (still well below that of human beings), with the multiplicative model coming a close second with a correlation of 0.19.

Turning to transitive cases, Grefenstette and Sadrzadeh (2011a) constructed a data set of simple transitive sentences with criteria similar to those used by Mitchell and Lapata for intransitives. Their data set

[67] Given how this and the following task are set up, we ignore the distinction between nouns and DPs, and, following Grefenstette and colleagues, we speak of nouns when it would be probably more appropriate to speak of DPs.

contains 200 subject-verb-object sentences rated by 25 subjects. An example of a high-similarity pair is *"map shows location"–"map pictures location"*, whereas *"map shows location"–"map expresses location"* is low-similarity (compare to: *"table shows/expresses results"*). Grefenstette et al. (2013), following the functional approach, model transitive verbs as third-order 300×300×300 tensors, estimated from example data using the procedure illustrated in Figure 3 above. The tensor is first multiplied by a 300-component object vector, giving a matrix corresponding to the VP, which is then multipled by a 300-component subject vector, to return the 300-component vector representing the sentence. In this task, inter-subject correlation is at 0.62, and using verb vectors only achieves a correlation of 0.08. Again, the functional model outperforms all the rivals, with a correlation of 0.32. The second best is Grefenstette and Sadrzadeh's state-of-the-art Kronecker model (0.25) followed by the multiplicative approach (0.23).

4.3 Adjective-noun constructions

Vecchi et al. (2011) studied a sample of the adjective-noun constructions (ANs) that never occur in the 3 billion word corpus we described above. These were rated as semantically acceptable or not by two linguists, resulting in a data set containing 280 acceptable and 413 "deviant" ANs (examples of acceptable: *blind trader*, *coastal mosquito*, *ethical trademark*; example of deviant: *blind numeral*, *coastal subtitle*, *ethical sunset*). Vecchi and colleagues hypothesize that the (model-derived) vectors representing acceptable ANs will inhabit areas of the semantic space that are more densely populated (with vectors of nouns, adjectives and corpus-attested ANs) than those inhabited by the deviant ANs: We might have not encountered *coastal mosquitoes* yet, since this is a sensible concept, there are many close concepts (*river mosquitoes*, *coastal bugs*...) that we are familiar with. Consequently, the model-derived vector for the *coastal mosquitoes* AN should be close to many corpus-derived vectors. On the other hand, *coastal subtitles* are not just unheard of, it is not even clear what a related concept should be. Operationally, Vecchi and colleagues measure the *neighborhood density* of a vector as the average cosine of the vector with its 10 nearest neighbours in distributional space (the denser the neighborhood, the higher this average cosine). Under the functional view, ANs are 300-component vectors derived by multiplying a 300×300 adjective matrix (trained from corpus examples) by a 300-component noun vector. Vecchi and colleagues found that the functional method, as well as the additive and multiplicative models (but not Guevara's method), correctly predicted a significant difference in density between acceptable and deviant ANs.

The functional method, moreover, predicts the largest difference.[68]

As discussed in Section 3.4, it is possible to measure the similarity of matrices and higher-order tensors just like we compare vectors, e.g., using the cosine method. Do these higher-order representations capture lexical similarity as well as traditional distributional vectors do? A small experiment reported by Baroni and Zamparelli (2010) looks at the specific case of (attributive) adjectives (that, in their implementation, are, again, 300×300 matrices mapping noun vectors to AN vectors). The results suggest that the higher-order representations of adjectives derived by matrix estimation are comparable and even *better* than vector representations directly extracted from the corpus. The task is to group 19 adjectives into 4 classes: color (*white, red...*), positive evaluation (*nice, excellent...*), time (*recent, new...*) and size (*big, small*). Baroni and Zamparelli use a standard clustering algorithm that assigns the adjectives to classes based on the similarity of their matrix or vector representations. Both the traditional vector-based and the matrix-based representations of adjectives achieve clustering performance significantly above chance, with matrices being better than vectors (0.74 vs. 0.68 in *purity*, a measure of how good the automated clustering is compared to the real classification, ranging from 1 for perfect clusters to values centered at 0.46 for random clustering).

Boleda et al. (2012b) tackle what is probably the most sophisticated linguistic issue that has been addressed with compositional DSMs until now, namely that of characterizing three kinds of adjectival modification: intersective (*white dress*), subsective (*white wine*) and intensional (*former criminal*) modification. Both intersective and subsective constructions are restricted to those involving color terms. From a denotational perspective, intersective adjectives (more precisely: intersectively-used adjectives) are those that produce the intersection of the set of entities defined by the adjective with the set of entities defined by the noun (a *white dress* is a *dress* and a *white thing*). Subsective (subsectively-used) adjectives cause the inference that the property denoted by the noun holds of the entities being described, whereas the property denoted by the adjective is just a proxy for a more descriptive property, and might or might not hold in a literal sense, with the resulting AN denoting a subset of the noun set (a *white wine* is a yellow-ish sort of *wine*; *brown bear* refers to the species

[68] Vecchi and colleagues also consider a length-based heuristic to measure acceptability. However, they estimate the adjective matrices with a length-insensitive criterion of fit between predicted and example output vectors. Not surprisingly, the length-based cue (that works for the additive and multiplicative methods) does not produce good results when using functional composition.

Ursus arctos, not to just any type of bear which is brown; a *white paper* is an exhaustive report, which might or might not be white). Intensional adjectives do not describe entities but rather complex operations that act on the intension of the noun they modify (a *former criminal* was a *criminal* in a past state of the world, but not in the current one). Boleda and colleagues first show that corpus-extracted vectors for ANs instantiating the different kinds of modification show global patterns in accordance with linguistic intuition. For example, intensional AN vectors tend to be significantly closer than the other types to their head noun vectors, since "intensional adjectives do not restrict the descriptive content of the noun they modify, in contrast to both the intersective and subsective [adjectives]" (Boleda et al., 2012b, p. 1230). Boleda and colleagues then proceed to test how well the compositional models mimic linguistically sensible patterns displayed by corpus-extracted ANs. They find that the functional approach (where, again, adjectives are 300×300 matrices estimated from corpus examples) provides the best approximation (the multiplicative method also does fairly well). Moreover, the functional approach produces composed vectors that are nearest their corpus-extracted counterparts, not only in the case of intersective and subsective ANs (where the additive model also does well), but also for the more difficult intensional adjectives (that is, only the functional approach is able to compositionally predict the corpus-attested distribution of, say, *former criminal*). Finally, qualitatively, the functional approach generates vectors that have the most sensible nearest neighbors. For example, the nearest neighbours of the functionally composed *artificial leg* vector include *artificial limb, artificial joint* and *scar*.

The experiments we reviewed, taken together, confirm that the functional approach to composition is empirically viable, and better than the ML mixture models and other state-of-the-art methods (although the multiplicative model performs fairly well across the board). Still, current data sets do not allow us to test some of the most important predictions we made in the previous section, for example that the functional model will be able to handle composition involving grammatical words, to take word order into account and to rightly compare full sentences with different structures (the intransitive and transitive sentences in the pairs used in the reviewed experiments have always exactly the same structure, and indeed share the same subject and object, with only the verb changing). Clearly, a high priority for the field – and one we are actively pursuing – is to build larger and more varied test sets, to truly explore the potential of more advanced compositional

models. For the time being, equipped with the theoretical framework we developed in the previous sections and the empirical results just discussed, we move on to motivating the compositional aspect of the distributional semantics enterprise.

5 Motivating compositional distributional semantics

The previous sections have laid out a method to go from distributional representations of words to distributional representations of phrases and sentences, and presented preliminary evidence of the empirical viability of this method. We have however not yet sufficiently motivated *why* we would need distributional representations of *phrases* and *sentences*, as opposed to just *words*. We are now in the position to take this step with the proper background.

There is ample evidence that distributional vectors capture many aspects of word meaning and play an important role in lexical semantics. Should we also expect that larger constituents are represented in distributional space? We will start by arguing that, once you assume that words have distributional representations, it is hard to avoid the conclusion that phrases and sentences have distributional representations as well. We will then proceed to discuss some semantic challenges where such representations might prove their worth, showing that they are not only "unavoidable", but also very useful.

5.1 The unavoidability of vector representations of constituents above the word

One could take the view that distributional semantics is a theory of *lexical* semantics, and compositional semantics should be handled by other means. A nicely spelled-out proposal of this sort was recently presented by Garrette et al. (2013). Garrette and colleagues assign vectors to content words (nouns, verbs, adjectives, some adverbs), but use a (probabilistic) logical formalism to capture sentential aspects of meaning, such as entailment between utterances. The vectors representing content words (contextualized using word-meaning-in-context techniques along the lines of those presented in Section 2.3) provide evidence exploited by inferential processes involving the sentences containing them, but no distributional representation of constituents above the word level is constructed. Grammatical words are seen as logical operators and they are not provided with a distributional representation, neither as part of phrases –since phrases are not distributionally represented– nor by themselves. Garrette and colleagues use, for example, distributional vectors to compute the contextualized similarity between *sweeping* in "*A stadium craze is sweeping the country*" and

covering in "*A craze is covering the nation*", and feed the resulting similarity score, together with Discourse Representation Theory representations of the sentences, to their probabilistic logical inference system, that uses these various sources of evidence to decide if there is entailment.

While the approach of Garrette et al. (2013) is extremely interesting, we find the restriction of distributional semantics to the representation of content words too limiting. First, if we assume a distributional representation for single words, it is strange that combinations of such words would have just a completely different logical-form representation. It would mean, for example, that it is meaningful to measure the degree of similarity between, say, *showering* and *bathing* (two content words, both with distributional representations enabling the similarity comparison), but not between *showering* and *taking a shower* (a content word, with distributional representation, and a phrase, not represented distributionally).

Indeed, the standard analysis of idioms as "phrases that behave as words", in the sense that they are stored in the lexicon with their compositionally unpredictable meaning, comes very natural if we assign distributional representations to phrases: An idiomatic phrase such as *red herring* is stored in the lexicon with a special vector that is different from the one that can be obtained compositionally. On the other hand, if phrases do not normally have distributional representations, the birth of an idiom would correspond to a big shift in the representation of the corresponding phrase, from logical form to vectorial representation. This "catastrophic" view of idiom formation does not sit well with the common observation that idiom formation is a gradual process, with different kinds of multiword expressions spread on a cline of idiosyncrasy: Compare the perfectly transparent *kick the ball* to the semi-opaque *kick the habit* to the completely idiomatic *kick the bucket* (see, e.g., Sag et al., 2002, and references there). While the details remain to be worked out, a view in which words and phrases have the same kind of semantic representation promises to handle the lexicalization cline of semantically opaque phrases better than a view in which words and phrases are very different objects, semantically speaking.

Garrette and colleagues limit distributional representations not only to *single* words, but to single *content* words. This is also problematic, given the fuzziness of the boundary between content and grammatical words. Everybody agrees that *car* is content and *the* is grammatical. But how about *several* and *various*? Syntactic tests suggest that *several* is a determiner (**the several friends*) –and hence a grammatical word– and *various* an adjective (*the various friends*) –and hence a content

word. However, the meaning of the two terms does not look dramatically different and, again, it seems artificial to assume that *several* is (only) a logical operator, while *various* comes with a distributional vector.

Or think of adverbs. It seems reasonable to consider *very* a grammatical word, perhaps to be formalized as an intensifying logical operator. However, take a *-ly* adverb such as *massively*. Intuitively, we want to assign similar analyses to *a very dirty look* and *a massively dirty look*, so *-ly* adverbs should also be treated as logical operators. But then, should we provide a (manually crafted?) interpretation for the potentially infinite set of *-ly* adverbs seen as logical operators? Alternatively, if their semantics is to be derived from the corresponding adjectives (that are certainly content words with distributional representations), how does the process of taking a vector and returning a logical operator work?

Another argument to treat content and grammatical words in the same way comes from the fact that often the meaning of a single content word is synonymous of a phrase containing one or more grammatical terms: e.g., *bachelor* can be paraphrased with *man and not married*. Note that we are not making the controversial claim that *not* is part of the semantic representation of *bachelor* (Fodor et al., 1980), but simply observing that it is meaningful to compare a content word (*bachelor*) to a phrase containing grammatical words (*not married*), which is problematic if grammatical words are just logical operators working above the lexical level.

Finally, consider the historical process of *grammaticalization* (Hopper and Traugott, 1990), whereby the same word starts its life as a full content word and progresses to become a grammatical element: Again, such gradual progression is hard to account for if content and grammatical terms have completely different semantic representations. Once more, it is simpler to assume that *all* words have (also) distributional representations.[69]

We have an argument, then, for the view that all words, including grammatical elements, have (also) distributional representations. In Section 3, we have argued at length that the right way to handle (most) grammatical words in distributional terms is as distributional compo-

[69]The same fuzzy-boundary arguments can be used to argue that, if there are reasons to represent grammatical elements in a logical formalism, then content words should also have a logical-form representation. We have no qualms about this conclusion. More specifically, we find it appealing to conjecture that the logical representations of content words are radically underspecified, with details about the conceptual knowledge they convey encoded in their distributional vectors.

sition functions. Putting the two conclusions together, if grammatical words have distributional representations, they are distributional composition functions, which in turn implies distributional representations for the phrases they construct.[70]

We conclude from the previous arguments that the position of Garrette et al. (2013) that only single (content) words, and not phrases of any complexity, have distributional representations is not tenable. Another very interesting recent contribution that promotes distributional semantics while rejecting composed vectors is that of Turney (2012). Although it is not directly relevant to compositionality, to fully understand Turney's account of the latter we must first introduce his domain and function spaces.[71] Turney represents the meaning of each word in *two* distributional semantic spaces, whose dimensions are populated with different kinds of co-occurrence counts. The dimensions of the *domain* space are meant to capture domain similarity: *Carpenter* is domain-similar to *wood* because both concepts belong to the domain of *carpentry*. The dimensions of the *function* space capture function similarity: *Carpenter* is functionally similar to *mason* because the two roles have the same function within different domains. Consequently, for each word pair we can compute two separate similarity scores.

Turney's proposal regarding phrases and sentences is that, instead of composing vectors representing these larger constituents and then measuring their similarity, we should first compute similarities between the words in the phrase (or in the sentence), and then *compose the similarities* to derive a single similarity score comparing the larger constituents. For example, to measure the similarity of *dog house* to *kennel*, Turney first computes the domain similarity of *dog* to *kennel*, and both the

[70] In the approach sketched in Section 3 above, there is a clear-cut distinction between words that act as *arguments* (vectors) and *functions* (matrices or higher-order tensors). This is a different cutoff from the one between content and grammatical words. For example, adjectives and verbs are content words that act as functions, and pronouns, being DPs, are grammatical words that should be treated as arguments. Just as in formal semantics, it might be difficult for specific combinations to decide which element acts as the functor and which as the argument (with type shifting operations possibly allowing both analyses for the same word), but the resulting competing theoretical proposals will still have a clear separation between functions and arguments, there is no "argument-function continuum". More importantly, all linguistic objects, whether functions or arguments, are represented by distributional tensors, so that there is not a big ontological leap from one type to the other. For example, a change-of-category rule, e.g., the one associated to a nominalizing suffix such as *-ness*, is easily modeled as a function from matrices (adjectives) to vectors (nouns).

[71] Turney's study also connects compositionality and relation analogy modeling. We do not discuss this aspect of his work here.

domain and function similarities of *house* and *kennel*. Then, Turney uses the geometric average of the resulting scores as his estimate of the similarity between the phrases. Different similarity composition functions are employed for phrases or sentences with different syntactic structures. For example, to compare *dog house* to *shelter for cats*, we would use a function that takes into account the fact that in this case we want to measure, among other things, both the function similarity between *dog* and *cats* and the one of *house* with *shelter*. Clearly, there is an explosion of possible similarity composition functions (we need to define, at least, a distinct function for each pairing of possible syntactic structures). Turney speculates that automated methods could be used to discover the right function for a certain pair of sentences or phrases (or, more generally, for a pair of syntactic structures, we suppose).

Turney's method is competitive against Mitchell and Lapata's additive and multiplicative models on the tasks of picking the right paraphrase for a composite nominal expression (e.g., *kennel* as the right paraphrase of *dog house*) and predicting similarity judgments about pairs of noun, verb and adjective-noun phrases. Turney's approach dramatically outperforms the Mitchell and Lapata models if the tasks are run on modified versions of the evaluation sets that take word order into account: Addition and multiplication will assign the same similarity to the pairs *dog house–kennel* and *house dog–kennel*, which is obviously wrong. In Turney's approach, on the other hand, similarity is sensitive to order (in one case, the overall similarity is a function of the domain similarity of *dog–kennel* and of both domain and function similarities of *house–kennel*; in the other, of the domain similarity of *house–kennel* and of domain and function similarity of *dog–kennel*).

The "dual space" idea is certainly worth exploring, and we also find the proposal of composing similarity scores very appealing. However, we do not see why the input to similarity composition should be limited to single word comparisons. Besides the huge number of similarity functions that need to be defined, this misses obvious generalizations. For example, different similarity functions are required to compare "*dogs sleep in kennels*" with "*dogs sleep in woody kennels*" vs. "*dogs sleep in kennels made of wood*", and yet another set of functions are required if the first sentence is replaced by "*domestic dogs sleep in kennels*". In an approach in which phrases have also distributional representations, we could instead define a single similarity composition function accounting for the previous sentences and many other structures by comparing, in each case, the subject noun phrase, verb and prepositional phrase vectors (and/or directly verb phrase vectors, that include the prepositional phases).

Turney conjectures that grammatical words could be either treated just like content words, and incorporated in the similarity calculations, or used as cues to guide the derivation of the right similarity composition functions. Regarding the first option, just as with the ML vector mixture approaches discussed in Section 3.1 above, it is not clear that vectors extracted from all contexts in which words such as *a* or *in* occur will carry any distinctive information. Moreover, we do not see, in most cases, how they would enter the similarity computations: When comparing *the dog sleeps in a kennel* to *dogs sleep in kennels*, which element of the second sentence should *a* be compared against? If we construct phrasal vectors, we can instead incorporate the contribution of the determiner in the computation of the similarity between the prepositional phrases *in a kennel* and *in kennels*.

But the second route (grammatical words guiding the construction of similarity composition functions) is even less appealing, since the contribution of grammatical words to meaning is reduced to signaling which content words are compared to which, and in which of the two spaces, and this is a very limited contribution. A reasonable role for *with* in the pairwise comparison N_1 *with* N_2–N_3 N_4 (e.g., *mansion with windows–terrace house*) would be to tell us that we must compare, domain- and function-wise, N_1 to N_4 and N_2 to N_3. However, when comparing N_1 *without* N_2 to N_3 N_4, all we can say about *without* is that it leads to exactly the same comparisons as *with*, which makes the two prepositions identical! Again, an approach in which *with* and *without* act as distinct distributional functions used to construct the distributional representations of different prepositional phrases is more appealing.

Turney also presents and extends an argument originally put forth by Erk and Padó (2008) against using vectors of the same size to represent sentences of all possible lengths. While the following is not quite the same argument that Turney presents (that is somewhat more technical and based on information-theoretic considerations), we think it captures its main point. In abstract mathematical terms, each component of a vector can contain an infinite number of real numerical values, and hence there is an infinite number of distinct vectors. However, when vectors are encoded on a psychical device such as a computer or a brain, the range of possible values that can be distinguished in a single component is finite, which makes the number of possible distinct vectors also finite. But a finite set of vectors cannot represent the meaning of the infinite number of possible sentences in a language.

In replying to this argument, we contend, first, that, just as vectors are infinite *in theory*, so sentence meanings are infinite *in theory*.

No single brain (or computer) will ever be capable or need to encode anything but a small finite subset of this infinity of possible meanings. Indeed, humans have problems keeping distinct, in their heads, the meanings of very long sentences that differ in just a few words. Second and more importantly, the argument is based on the unwarranted inference from the premise that sentences have vector representations to the conclusion that sentence meaning is represented by these vectors and these vectors *only*. We (and, we suspect, most proponents of compositional DSMs) agree with the premise, but strongly disagree with the conclusion. If we build the sentence vector compositionally from distributional representations of its parts, we do not see why these intermediate representations should be thrown away once the top node is reached. We find it more natural to assume that any semantic operation that refers to the distributional meaning of a sentence can access the vector representing the whole sentence as well as the vectors (or tensors) representing all its sub-constituents, down to the word level. And, going beyond distributional meaning, we do not dispute either that sentences will also have one, or indeed many, logical-form representations (a point we shall shortly return to in the conclusion). Thus, we are interested in arguments against distributional representations of sentences (and long phrases) that show that such vectors are not *necessary*; we do not need to be persuaded of the fact that they are not *sufficient*.

We conclude this section by observing that, while we hope to have argued convincingly for the extension of distributional representations beyond single content words, we do not know whether *every* word, phrase and sentence should have a distributional representation. There might be good reasons to represent determiner and verb phrases distributionally (e.g., to insure that phrases such as *dog house* and *kennel* or *taking a shower* and *bathing* are directly comparable), but the motivations to assign distributional representations to larger expressions are not so clear (for example, when discussing Turney's "composition of similarity scores" idea above, we suggested to replace the comparison of single words with the comparison of *phrases*, rather than whole sentences). It's far from clear where the line between the constituents that need distributional representations and those that don't should be drawn. Ideally, we will want to strike an acceptable balance between what DSM representations "do for you", and what they cost (in time spent creating and applying them). Perhaps the distributional representation for some or even most sentences will be something vague, perhaps just a hint that the sentence sounds "formal", "threatening", "odd", "funny" or "positive". Knowing this much would be useless for

drawing inferences, but it would be valuable information if the task is to decide whether the sentence can be embedded under *complained that*, *boasted that* or *joked that*. So, for some purposes, there could be reason enough to keep around even vague DSM representations for higher constituents, for others there might not. For the time being, we assume that all linguistic constituents up to sentence nodes have a distributional representation, and leave it to future work to look for principled ways to determine the upper syntactic bound on distributional representations, possibly on a task-dependent basis.

5.2 The usefulness of vector representations of constituents above the word

The arguments we presented in the previous section in favour of distributional representations for larger constituents are negative in nature: If you accept that content words are associated to distributional vectors (and there is ample lexical-semantic evidence for the usefulness of the distributional representation of content words), then it's difficult to deny distributional representations to function words and larger constituents. However, once we have such representations, what can we do with them? How can distributional representations of phrases and sentences aid and/or complement the classic truth-functional representation of utterances?

Note that this is a more specific question with respect to the more general issue of whether distributional semantics can help (compositional) formal semantics, a topic we briefly addressed in Section 2.6. For example, in formal semantics it is typically (tacitly) assumed that word meanings are disambiguated before composition applies. Contextually disambiguated vectors (see discussion and references in Section 2.3 above) can help solve the mystery of how such disambiguation takes place. This is a way in which distributional semantics can help compositional semantics, but it does not require distributional representations above words and perhaps simple phrases.[72] Similarly, in Section 2.6 we (very tentatively) hypothesized that the distributional representation of a sentence might help to pick up the right reference for the sentence in the outside world. This would be of great help to compositional semantics, but we suspect that the level of words or simple phrases is

[72]Note, however, that, as we conjectured in Section 2.3, a compositional DSM might largely sidestep issues of polysemy and disambiguation by implicitly disambiguating terms as part of the composition process. The (relative) success of the intransitive and transitive sentence experiments reported in Section 4.2 above, where handling verb ambiguity plays a crucial role in getting the right similarity scores, suggests that this is an empirically viable approach to polysemy.

better suited to perform the matching with real-world percepts than full sentences. The anchoring of sentence "*a black dog is barking*" to the outside world might proceed by matching a (multimodally-enhanced) *black dog* vector to vectors representing objects in the current scene, in order to scan for candidate black dogs, and a (multimodally-enhanced) *barking* vector to current auditory events, in order to scan for candidate barkings, rather than by matching a single holistic vectorial representation of the sentence to a holistic vector representing all the current percepts together.

We focus here on the usefulness of *compositional distributional semantics*, and not on the just discussed potential contributions of word-level distributional models to compositional semantics. We want, moreover, to look at uses of phrasal and sentential vectors that should be of direct interest to purveyors of semantic theory, rather than aimed at engineering applications, although effective distributional representations of sentences *are* of considerable practical interest. In particular, such representations can be employed to measure sentence similarity in order to detect paraphrases (informally, paraphrases are sentences that mean approximately the same thing: we will get back to them shortly). Paraphrase detection, in turn, is useful for information retrieval tasks such as finding semantically equivalent ways to query a data-base or the Web, or avoiding search results that overlap with the query in lexical terms but have very different meanings. Another natural application for paraphrase detection[73] pertains to the evaluation of machine translation systems, where we must check if the translation provided by a system is just a rephrasing or it is significantly different from a benchmark manual translation. Other practical tasks helped by paraphrase detection include text summarization, question answering and shallow forms of text understanding, such as recognizing whether a short text entails a certain conclusion (Dagan et al., 2009). Another application domain where compositional distributional semantics has already proved its worth is in predicting degrees of positive or negative evaluation expressed by sentences, where it is important to look not just at single words, but also at how they are combined: *Very bad* is a more negative assessment than *bad*, but *very good* is more positive than *good* (Socher et al., 2012).

Assessing sentence similarity

After the previous brief excursus on applications, which we included for the benefit of potential industrial funders, let us turn to more theoretical concerns. Just as with words, the main function of sentential

[73]Suggested to us by Stephen Clark.

(or phrasal) vectors is to measure the degree of semantic similarity between sentences (or phrases). This is probably not of immediate help to determine the truth conditions of sentences. To know that "*a dog is barking*" is very similar to "*one canine creature arfs*" won't (directly) help you establish under which conditions the first utterance is true. However, there are other important aspects of meaning that similarity might be better suited to handle than truth conditions.

First and most obviously, humans do have strong and reasonably consistent intuitions about phrase and (simple) sentence similarities (as shown, for example, by the relatively high inter-subject sentence similarity correlations in the benchmark of Grefenstette and Sadrzadeh 2011a discussed in Section 4.2 above). The notion of a *paraphrase*, in particular, seems psychological robust, and difficult to capture in truth-functional terms.[74] The "*one canine creature arfs*" sentence above strikes us as a rather close paraphrase of "*a dog is barking*", but it's hard to characterize this intuition in terms of truth conditions. The sentences are not tautologies (the canine creature could be a coyote, arfing is not quite barking), and to simply claim that they are not contradictory is too weak a condition for paraphrase status. It might be possible to capture paraphrasing in terms of possible worlds (something along the lines of a requirement that paraphrases must share truth values in a certain proportion of worlds of a certain kind), but this seems a rather torturous way to account for something that can be modeled very straightforwardly by compositional distributional semantics, as already shown empirically by the studies we reviewed in Section 4.2. Moreover, the very notion of "close paraphrasing" (as used, for example, by lawyers to assess plagiarism claims) suggests that being a paraphrase is not an all-or-nothing property: There exist closer ("*a dog is barking*"–"*one canine creature arfs*") and more distant paraphrases ("*a dog is barking*"–"*a small mammal is making sounds*"). This gradient property follows naturally from the view that paraphrases are neighboring sentences in distributional space, but it is difficult to capture in a logical formalism.

Semantic anomaly detection

Semantic anomaly is another important aspect of meaning that, we believe, can be captured more appropriately using distributional rep-

[74]Note that paraphrasing is not just a metalinguistic ability, but it's likely to play a role in many unconscious everyday linguistic tasks, such as deciding the best way to communicate a thought or quickly determining whether a piece of news brings new information. More speculatively, paraphrasing could be used as a fast surface-y way to reformulate a statement in a form that is better suited for deeper logico-semantic analysis.

resentations of sentences.[75] Chomsky's (1957) famous "*colorless green ideas sleep furiously*" example demonstrates how a sentence can be at the same time perfectly grammatical and completely nonsensical. Chomsky used the sentence as part of an argument against statistical models of language (that would fail to distinguish between this unattested but syntactically well-formed sentence and equally unattested but grammatically ill-formed ones).[76] However, ironically, the kind of purely semantic ill-formedness illustrated by this sentence resists an account in terms of the formal models of meaning developed within the paradigm of generative grammar. The natural way to tackle semantic ill-formedness with the classic apparatus of formal semantics is by adopting a very rich and granular inventory of semantic types (see Asher, 2011, for a very interesting recent proposal in this direction). However, to capture violations such as that ideas cannot be green or that sleeping cannot be performed in a furious manner, one would need a very rich ontology made of thousands of types, and it is not clear how such ontology could be learned from data (recall that the logical approach, unlike distributional semantics, lacks practical algorithms for large-scale induction of semantic knowledge from naturally occurring data). But even with a rich type ontology, the extended theory of semantic types might not be the right instrument to characterize semantic anomaly. First, anomaly is highly context-dependent (*green ideas* sound good in "*green ideas are dominating the global warming debate*"), and accounting for context-dependency will make the theory of types even more complex. Second, semantic ill-formedness judgments are not sharp like syntactic ones, but are rather spread on a cline of acceptability from the completely natural ("*dogs bark*") to the utterly nonsensical (the Chomsky sentence) via various degrees of semantic plausibility: "*?cats bark*", "*??closets bark*", "*???preferences bark*".

Collecting large amounts of lexico-semantic knowledge from data, handling context dependency and modeling graded judgments are, however, core properties of the distributional approach to meaning. And this is clearly a job for *compositional* distributional semantics: We cannot see how you could measure the semantic plausibility of a sentence or a phrase using just the distributional representations of the component words, without combining them. In Vecchi et al. (2011) (briefly reviewed in Section 4.3 above), we implemented and tested simple methods to

[75] A system able to predict degrees of semantic anomaly will be able to perform many linguistically and practically important tasks, such as checking if an argument satisfies the selectional preference of the verb it depends from.

[76] See Pereira (2000) for an interesting discussion of how modern statistical models can address Chomsky's challenge.

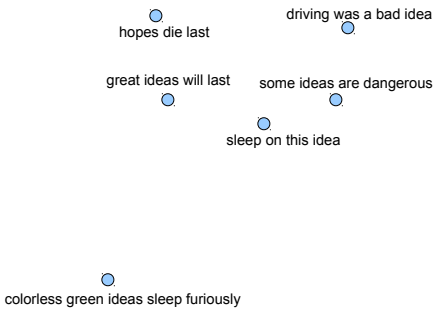


FIGURE 4 Nonsensical sentences might be isolated in semantic space.

measure the degree of semantic acceptability of phrases using compositional DSMs. Extending ideas from that work to sentences, we hypothesize that properties of the semantic space neighborhood inhabited by a sentence will provide us with information about the plausibility of the sentence. One simple hypothesis in this direction (with preliminary support from the work by Vecchi and colleagues) is that semantic acceptability correlates with the density of a sentence neighborhood. Figure 4 is a cartoon illustration of this hypothesis. A meaningful sentence such as "*some ideas are dangerous*" will have many neighbours, that is, sentences that state related things. On the other hand, the nonsensical "*green ideas*" sentence will be out there in semantic space, without any (meaningful) sentence stating related facts to keep it company.

An approximation to this approach would involve constructing a large set of (meaningful) sentence vectors (e.g., taking relatively frequent sentences from a corpus), and measuring how populated the area surrounding an arbitrary point (corresponding to a compositionally-derived sentence) is, or what are the closest neighbours of a given point. Yet, if we want to follow this idea to its full consequences we will need a different, more ambitious and more interesting method, which at present we can just sketch.

The compositional DSM enterprise, if fully successful, would allow us to build a meaningful vector for any meaningful and syntactically correct sentence. The motivation for doing this compositionally is by now very familiar: the space of possible sentences is infinite. Suppose, now, that we want to reverse our task: Given a vector $\mathbf{s}$ produced by

the distributional composition of a sentence, we want to ask which are the sentences that, if fed through the same process, would produce the vectors closest to **s**. In the best of possible worlds, if **s** has been created from S (let's say that **s** is a 'distributional composition' of S, **s** =CO(S)), the "noisy" inverse function GEN[77] applied to **s** should give back a set containing the closest paraphrases of S within certain bounds of length, complexity, etc. (including S itself). The distance of the generated phrases to S would then be measured by applying CO to them to obtain the corresponding vectors, and measuring the cosines of the latter to **s**. In particular, it should be the case that cos(CO(S), **b**) ≈ 1.

An empirically effective formulation of GEN would have great practical importance, since without it we can only ask whether two candidate phrases are similar, but we have no way to *generate* similar cases, which is a fundamental step of the approach to semantic anomaly we just sketched and for some of the linguistic applications discussed below. We must however leave the construction of (approximations to) GEN to further work.

Alternative classes

There are several phenomena in formal semantics where the truth of a statement can be established only on the basis of a set of alternatives. The best-known case is probably Mats Rooth's (1985, 1995) seminal analysis of *association with focus*. Consider (18), where uppercased represents sentential stress and square brackets the focused constituent.[78]

(18) The candidate only [shared a CAB]$_F$ with the mafia boss.

For this sentence to be true, the candidate must have, of course, shared a cab with the mafia boss, and, to capture the semantics of *only*, he must have not done any of a set of alternative possible things he could have done with the mafia boss.

The problem is what the set of 'alternative possible things' amounts to. Obviously, (18) cannot be saying that the candidate could not have, say, shared a restaurant with a boss, or seen the mafia boss on TV.

[77]The name "GEN" should remind familiar readers of the formally similar problem of *candidate generation* in Optimality Theory (OT, Prince and Smolensky, 2004). In OT, a set of ordered violable constraints can decide which of a set of candidates 'wins', i.e., satisfies the most important constraints. OT has no independent way of generating the candidates to be evaluated, and this task is left to an unspecified function also named GEN.

[78]Due to the mechanism of focus percolation, the area affected by focus is typically larger than the stressed (sub)constituent.

Intuitively, we understand the sentence as claiming that the most compromising relation the candidate had with the mafia boss was sharing a cab, which is maybe not particularly compromising. (18) thus excludes relations like 'being friend' or 'going on vacation' with him, 'paying him regular visits' and the like.

Unfortunately, formal semantics has very little to say on what the alternative set actually contains: Its content is assumed to be dependent on context, and is typically left to an underspecified 'pragmatic module'. However, without knowing the actual content of the alternative set, and indeed without having a measure of the extent to which an assertion counts as a valid alternative to the focused constituent, we have no way to explain why the replies in (19-a-e) sound progressively less convincing as rejections of (18).

(19)
a. That's false: he also worked for the mafia boss!
b. That's false: he also shared a house with the mafia boss!
c. That's false: he also ate in the same pizzeria as the mafia boss!
d. That's false: he also ate at the same time as the mafia boss!
e. That's false: he also lived in the same country as the mafia boss!

The problem is pervasive since alternative sets are part and parcel of many semantic operations. They appear with what is sometimes called metalinguistic negation (20); in generic sentences (21); with scalar operators like *even* (22), etc.

(20) John didn't just [play a PRACtical joke]$_F$ on his colleague! He positively TERRified her!

(21)
a. In San Petersburg, [OFFicers]$_F$ always escort ballerinas (not low-rank military personnel)
b. In San Petersburg, officers always escort [balleRInas]$_F$ (but not other artists)

(22) John even [SANG]$_F$ at the party (not just took part in it)

The problem is always the same: not all possible alternatives count (for instance if a ballerina is sometimes escorted by her husband, this doesn't seem to contradict (21-a); but if she was escorted by simple soldiers, it does.).

We believe that DSMs could actually offer a principled way to address this problem.[79] Very sketchily, the idea is to produce a tensor for the focused constituent in context, then use this tensor to decide which constituents would be sufficiently similar, i.e., which one would produce similar-enough tensors. The acceptability of the cases in (19) would thus have to be evaluated by computing the tensors for the constituents corresponding to the focus element of (18) (*"share a cab"*) and measuring the distance from its tensor. In other cases, e.g., to generate the continuation *"not low-rank military personnel"* in (21-a), we would need to use the function GEN to produce examples of actual close alternatives (excluding, of course, those that are entailed by the focused assertion, like *"officers escort ballerinas"*).

Note, finally, that while focus on single elements (e.g., *John didn't actually [KISS]$_F$ Mary*) could still be handled by word-level DSMs, any case in which focus spans a VP or a complex nominal would require the full power of compositional DSMs.

Generic information

We already discussed in Section 2.6 how DSMs, reflecting statistical trends from large corpora, will capture generic rather than episodic knowledge. Indeed, modeling the acceptability of *generic sentences* is another big challenge for a truth-functional view of meaning (Krifka et al., 1995, Cohen, 2004) where distributional semantics might help. Generic statements about regularities in whole classes of objects, such as *"birds fly"*, *"mammals do not fly"* or *"lions have a mane"* are found acceptable despite the presence of (sometimes large and systematic) exceptions (*penguins*, *bats*, *lionesses*), which makes them difficult to handle in straightforward logical quantification terms.

Two aspects of generics suggest that distributional semantics might have something to contribute to this task. First, sentences such as *"lions have a mane"* express a facet of our general knowledge about a concept – in this case, that of a lion (Carlson, 2009, draws an explicit connection between generics and conceptual knowledge). Not surprisingly, DSMs derived from large corpora are good at extracting general world knowledge about concepts (Almuhareb and Poesio, 2005, Baroni et al., 2010, Kelly et al., 2012), so it is reasonable to expect their compositional extension to capture valid statements about properties of concepts.

Second, acceptability judgments about generic statements are not sharp. *"Lions live in Africa"* is perfect, *"lions live in Europe"* sounds

[79]The relevance of compositional DSMs for the creation of alternative sets was first pointed out to us by Jacopo Romoli (p.c.).

funny but it is not nearly as bizarre and obviously false as *"lions live on the moon"*. Again, it is easier to model this sort of gradience in the geometric framework of distributional semantics than in truth-functional terms. In particular, there might be a relation between the way in which we just proposed to handle semantic acceptability and anomaly in general and the case of generics in particular. Not by chance, Chomsky's "*green ideas*" and almost all the examples we discussed above when speaking of semantic anomaly are in the form of generic sentences with bare plural subjects. And, again, density and other properties of the neighborhood of a generic statement might turn out to predict its degree of acceptability.[80]

6 A cursory look at some further relevant work

The last years have seen an enormous increase in the amount of published work on how distributional meanings can be composed; we refer to Clark (2013b), Erk (2012) and Mitchell and Lapata (2010) for overviews. In previous sections, we have already discussed in depth some of the closest related work (see in particular Section 3.1 for a discussion of Mitchell and Lapata, 2008, and Section 5.1 for our view of Garrette et al., 2013 and Turney, 2012). Here, we limit our discussion to work that share our goal and that uses approaches either very close or radically different from ours.

Before going into the details of this work, it is worth mentioning that there is a rich tradition of corpus-based statistical semantics methods producing compositional representations that are different from the classic logic-based ones, but are not distributional in our sense. This line of research includes the corpus-based induction of semantic parsers suitable for question answering (Liang et al., 2011), role labeling (Titov and Klementiev, 2012) and modeling semantic and syntactic acquisition (Kwiatkowski et al., 2012). The output of such semantic parsers could be tried as an alternative to the purely syntactic CG input used in our current work.

Recently, there also has been much interest in higher-order tensors for distributional semantics (e.g., Baroni and Lenci, 2010, Giesbrecht, 2010, Turney, 2007, Van de Cruys, 2010, Widdows, 2008). However,

[80] A proper characterization of generics must take into account not only conceptual mismatches and world knowledge, such as the incompatibility of the class denoted by the subject with the property denoted by the predicate (lions are not the sort of things that live on the moon), but also grammatical aspects, such as the way in which the subject is overtly quantified: *"lions live in Africa"* sounds "true", but *"all lions live in Africa"* isn't; *"lions live in Europe"* sounds "false", but it is certainly the case that *"some lions live in Europe"*.

even when this line of work tackles the issue of compositionality, it looks at tensors as a way to represent larger structures that result from composition, rather than taking the view we propose here of using tensors to encode composition functions.

Our view on the syntax-semantics relations traces back to the traditional type-logical approach introduced by van Benthem in 1986, following which we have exploited the Curry-Howard correspondence between logical rules and lambda-calculus rules to obtain a proof term for a parsed structure. The proof-term gives the backbone to be filled in with the interpretation of the linguistic signs composed. In our case, we have a clear-cut division of the workload. The grammar and the Curry-Howard correspondence with the lambda-calculus take care of building the proper structure by taking into account the expressivity issues, over-generation and under-generation problems as well as the scope possibilities that sentences with logical operators may display, and the distributional model accounts for lexical meaning and meaning composition.

Clarke (2011) studies the algebraic properties a vector space used for representing natural language meaning needs to have; the author claims the composition operation has to be bilinear (components of meaning persist or diminish but do not spontaneously appear; e.g., both *red* and *herring* must contain some components relating to the meaning of *red herring* which only come into play when these two words are combined in this particular order), associative and distributive. By taking this abstract view, Clarke manages to define a general framework that covers under its umbrella our approach together with, for instance, the one of Mitchell and Lapata (2008), discussed in Section 3.1, and Clark et al. (2008), which we discuss below. Moreover, the author identifies possible ways to account for degree of entailment between distributional representations, proposing to exploit the partial order of the defined algebraic structure.

Besides the general abstract framework, Clark et al. (2008) (and the extended version in Coecke et al., 2010) share also our view of the syntax-semantics interface of natural language and of its formal models. Similarly to us, they capture compositionality by exploiting a morphism between the syntactic and semantic building systems. Differently from us, they define a morphism directly between a grammar (a pre-group) and a vector space model without going through the intermediate step of the lambda-calculus. The choice of the pre-group is due to the authors interest in the category-theoretic perspective under which pre-groups share a common structure with vector spaces and tensor products. A different view is taken in Clark (2013a) where the

author discusses the framework in terms of multi-linear algebra providing a more concrete and intuitive view for those readers not familiar with category theory. At the level of lexical and phrasal interpretation, Clark et al. (2008), Coecke et al. (2010) and Clark (2013a) import Frege's distinction into DSMs by representing "complete" and "incomplete" expressions as vectors and as higher-order tensors, respectively, and consider the syntax-semantics link established between syntactic categories and semantic types. For instance, a transitive verb has syntactic category $DP^r \cdot S \cdot DP^l$ (that corresponds to the functional CG category $(DP\backslash S)/DP$) and semantic type $N \otimes S \otimes N$, since expressions in DP and S are taken to live in the semantic space of type N and S, respectively, and the transitive verb relates these vector spaces via the tensor product ($\otimes$): its dimensions are combinations of those of the vectors it relates. As explained in Clark (2013a), the verb vector can be thought of as encoding all the ways in which the verb could interact with a subject and object in order to produce a sentence, and the composition (via inner product) with a particular subject and object reduces those possibilities to a single vector in the sentence space. Different implementations of this framework have been proposed by, e.g., Grefenstette and Sadrzadeh (2011a), Grefenstette and Sadrzadeh (2011b), Coecke et al. (2013), Kartsaklis et al. (2013). The one closest to us is the one by Grefenstette et al. (2013), where the authors further develop the framework discussed above. In particular, they exploit a tensor contraction operation that guarantees an equivalence between tensor order and semantic type. Tensor contraction is closely related to our generalized matrix-by-vector product and indeed, the framework of Grefenstette and his colleagues can be seen as an abstract formalization of the one we are proposing here. Grefenstette and his colleagues (that, not by chance, include one of the authors of the current paper) also bring together the Coecke et al. formalism with ours by adopting the regression-based learning method we explained in this article (the empirical results of Grefenstette et al., 2013, are reviewed in Section 4.2 above).

We share the idea of learning composition functions by regression on corpus-extracted examples of their inputs and outputs with Guevara (2010), who, however, treats all linguistic expressions as vectors without distinguishing them into atomic and functional types. The importance of exploiting input and output training data for building compositional distributional semantic models is also stressed by Zanzotto et al. (2010), who present a model similar to the one of Guevara, but cleverly exploit dictionary definitions to extract both positive and negative training examples.

Another model that is closely related to ours is that of Socher et al. (2012), who also implement function application in terms of operations on matrices and vectors. However, differently from us, they treat each word equally, as both a vector and a matrix. Distributional composition is doubled – each word matrix is composed with the lexical vector of the other word in a phrase – and the result is still a pair of a vector and a matrix. Since Socher et al. (2012) do not use tensors higher than matrices, all composition is pairwise, whereas we have presented a model of composition permitting functions of larger arity. Finally, Socher and colleagues estimate the weights of their models by direct optimization of a specific semantic task, thus requiring hand-labeled examples of the intended output of the task, and producing different representations of the same linguistic expressions depending on the intended task. Direct empirical comparison of our approach to the one of Socher and colleagues is an important item in our future work agenda.

Rather different and quite interesting points of view are assumed in Garrette et al. (2013) (already discussed in Section 5.1) and Copestake and Herbelot (2012). Garrette et al. (2013) adopt a formal semantics framework to build First Order Logic (FoL) representations of sentences, lexical distributional semantic representation for computing word similarities and weighting FoL clauses, and Markov Logic Networks for reasoning on such clauses. Hence, they exploit the FoL logical operators, like negation, existential and universal quantifiers, to draw inferences involving their relations, and DSM representations to draw inferences involving lexical information.

The integration of logical and distributional models is studied in Copestake and Herbelot (2012). Here too, the authors distinguish the interpretation of closed and open class words by adopting a view closer to the logical one for the former and the distributional one for the latter, but they propose more drastic changes and an integration of the two models. On the one hand, they import in the logical model the idea that the meaning of a content word is given by the contexts in which it occurs, hence they replace the denotational sets of entities with distributional sets of contexts. On the other, they take the contexts to be logical rather than linguistic ones, namely the semantic space components are not words, but their logical representation. For instance, *jiggle* would be said to co-occur with the logical representation `table'(x)` if the corpus, from which co-occurrence information is extracted, contains the sentence "*the ball on the table jiggled*". As in standard distributional semantics, here too there are different possibilities for choosing the logical context to be considered. For instance, logical contexts can be those predicated of the same entities of the tar-

get word, or those related by paths of a certain length to it, reaching sets of logical contexts corresponding to a very fine-grained notation of semantic features. Finally, each distributional set records also the connection of the formal logic representation of the sentence to the situation in which the sentence was uttered, as well as the entities that are arguments of the logical forms in it. This rich information allows to preserve the idea of extension as well as to distinguish words' intensions (sets of logical forms) even when their extension is the same.

7 Open issues and conclusions

In closing this paper, we want to touch on some of the open issues we see along the road to a full-sized DSM semantics, and return to the general system architecture.

One of the great advantages of DSMs is that they hold the promise to handle *polysemy* gracefully. We do not need to have a separate entry for *brown* in each of the *brown N* phrases cited in the Introduction (*"brown cow"*, *"brown book"*, etc.): The *brown* matrix will produce a sensible semantic value for (almost) all the nouns it is applied to. *Homonymy*, like the fact that *page* refers to a piece of paper or a person, is a different matter. The difference between polysemy and homonymy is notoriously difficult to draw, but the existence of a distinction seems indisputable. For instance, most people need some linguistic training to note that in co-predication cases like (23) *lunch* must mean different things (an object can be tasty, an event cannot), but any speaker is aware of the fact that *page*, *bank* or *bass* can each mean very different things.[81] With true homonyms co-predication is impossible (*"*the page was written in Latin and knew this language"*).

(23) Lunch was tasty but lasted forever.

Based as they are on word forms, DSMs tend to overlook homonymy, resulting in vectors that conflate all the different senses of a lexical expression. This is certainly a problem, but one which has been amply addressed at the lexical level in the computational community (see McCarthy, 2009, and Navigli, 2009, for recent surveys, and the discussion in Section 2.3). Word sense disambiguation techniques have lim-

[81] Words can be both polysemous and homonymous: *page* in the paper sense could refer to the object or the text in it, in the human sense, to the person or the position (*"he was nominated page"*). This suggests that homonymy and at least regular polysemy should not be seen as opposite values on a single scale. See Copestake and Briscoe (1995), Boleda et al. (2012a) for discussion of various approaches and Frisson (2009) and Klepousniotou and Baum (2007) for experimental results on the psychology of this distinction.

ited success in fine-grained sense distinctions—fortunately, those that a DSM approach seems to be best at handling—but perform well on true homonymy, where there is no (synchronic) relation between the various senses. The output of these algorithms could thus be a semantic tagging of words in context, which distinguishes, say, between $page_{paper}$ and $page_{person}$. Suppose that tokens that cannot be disambiguated with confidence are left unlabeled, and that each labeled sense receives its own vector. When a compositional function like the adjectives *written* encounters an unlabeled ambiguous noun like *page* it could look up the vector for each of its possible meanings, then apply semantic anomaly detection, described in Section 5.2, applied to the output of *written(*$page_{person}$*)* and *written(*$page_{paper}$*)*, to guess which meaning is probably correct for *"written page"*. If the ambiguity cannot be resolved locally it will be carried up the tree until it can be resolved by further compositions, up to the sentence (and potentially, the discourse) level. This will require a mechanism for storing multiple disjunctive meanings, but such a system would also be necessary if we want to store both compositional and idiosyncratic DSM-meaning for a constituent (to be able to process idioms in their literal and figurative use; see discussion in Section 3.5). It will make the system more complex, but it should not pose any special theoretical challenge, except for the general problem of deciding which words should be treated as homonyms. In the worst case, we run the risk of incorrectly treating a polysemous word as a case of homonymy, creating n possible lexical items with n vectors, where one should have been sufficient. Note however that in this case the vectors for the various senses will be quite similar, so even if the disambiguation system fails to distinguish one sense from the other, the harm done to the global system should be very limited.

A boundary case of 'lexical ambiguity' to which we will need to devote special attention is that of words with multiple argument structures. Recall that in our compositional DSMs the number of argument a word takes determines its shape, and that objects of different shapes cannot be directly compared (see Section 3.4). But many nouns can optionally take arguments: we have *mother*, but also *mother of twins*. Most nouns derived from transitive verbs can take PP arguments that correspond to the direct object of the verb (*"the end (of the paper) is near"*; *"the discovery (of America) was surprising"*, etc.).[82] Thus, it

[82]Treating all these cases as PP-adjuncts, i.e., cases where the preposition *of* takes the DP and the N to form a modified N, i.e., (N\N)/DP, does not capture the fact that these are true arguments of the nouns, as they are of the corresponding verbs. Adjuncts are not selected; they can be attached to any noun, not just to relational ones. Moreover, arguments and adjuncts can combine, but only in one

would appear that the same nouns are sometimes vectors, sometimes tensors, i.e., functions from the space of their arguments to nominal vectors.[83] The problem is also found with those verbs that have a transitive and an intransitive version (*John ate his lunch*/*John ate*): the VPs would be comparable (both matrices), the Vs would not. We reach the counterintuitive conclusion that the *end* in *"the end is near"* and *"the end of the paper is near"*, or the transitive and intransitive usages of *eating* are completely different and incomparable linguistic objects.

We see two linguistically informed solutions for this problem. First, we could treat verbs like *run* or *eat* as uniformly transitive, training them on their objects when they have one; when they do not, we could apply the transitive function to an 'internal object', built by averaging the vectors of the most frequent actual objects of the transitive version.[84] We surmise that the same approach could be applied to nouns.

A second approach would be to use corpora to learn a mapping from the version of a word with an argument to the version without, or vice-versa. In this case, we would collect vectors for occurrences of *eat* or *end* without complements, vectors for occurrences of the same words when they have arguments, then use regression to establish a mapping (a third-order tensor if we map from a matrix to a vector). This approach might be feasible, but it is too general: nothing would prevent the function from being applied also to nouns that never take arguments.

Both routes are worth exploring if we want to handle a different and more productive case, that of *active/passive* alternations (24). In passives, the external argument of a transitive verb becomes syntactically optional and can be expressed by a *by*-phrase.

(24) a. Mary kissed John.
b. John was kissed (by Mary).

The two approaches (providing a null, 'average' argument, or learning

order (in *"a mother of twins of high social status"* only the second *of*-PP is an adjunct; *"*a mother of high social status of twins"* is impossible, at least with a neutral intonation).

[83]Following an old but still popular analysis (Stowell, 1981), we could simplify things a bit taking the *of* that introduces nominal arguments to be a pure case marker. Semantically, this *of* would denote the identity function, which simply returns the vector of its DP argument. Relational nouns would then be matrices that map vectors in DP-space to vectors in N-space.

[84]This would attempt to capture the common observation that the understood object of the intransitive versions has to be somehow prototypical. If John is *"eating the dust"*, *"running a risk"* or *"drinking poison"*, it is odd to say that he is *eating, running* or *drinking*.

a transitive-intransitive mapping) have different strengths and weaknesses. The first essentially reduces (24-b) to *"someone kissed John"*, but is computationally straightforward (after training *kiss* on active sentences alone one would be able to compose the DSM for a passive sentence with no additional training). The second (learning from the corpus a mapping from active to passive verbs or vice-versa) seems potentially superior at capturing the fact that active and passive voice might be used in different contexts—their information structure is not the same (see, e.g., Lambrecht, 1996). Neither methods, however, could recover the agent within *by*-phrases; this would probably require a more structural approach, akin to the tensor transposition we saw for relative clauses. In principle, cases like the causative alternation (*"the missile sank the ship"*/ *"the ship sank"*) or the middle construction (*"the shop sells the book"*/ *"the book sells well"*) could be handled in a similar way, though note that in these cases, unlike in passives, the necessary DSM-manipulations could not be associated with the presence of an affix (this was the spirit of the decompositional morphology hinted at in Section 3.4).

Operations that take existing forms and generate variations would be the DSM-equivalent of *lexical redundancy rules* in transformational syntax (Chomsky, 1970): *if* a certain linguistic item has been observed with a certain argument structure, *then* we can generate a new argument structure for it according to the functions we have learned. Despite the potential risk of overgeneration, the opportunity to establish interesting mappings for a large set of constructions is wide open: For instance, we could easily imagine a rule, triggered by Subject-Aux inversion, which maps a declarative-sentence vector into a yes/no-question vector.

As discussed in Section 3.5, a key ingredient for the success of a full-scale DSM-distributional approach is representational efficiency. This means, among other things, having more compact and efficient methods for representing and learning tensors, as well as the possibility to recognize and exploit the similarity of different linguistic objects in the learning phase. The latter would be crucial at the lexical and phrasal level, to learn rare forms. Recognizing, for instance, that *indigo* names a color, we could use whatever knowledge we have about other colors to extrapolate a part of its semantics.

A different case in which similarity of structures might play a role is the way we could handle generalized *coordination*. This extremely pervasive operation has many properties that set it aside from other constructions in grammar (see, e.g., Zamparelli, 2011, for a review). One is its promiscuous categorial behavior: *and* can join any pair of syntactic

categories (in CG, we can define the type of *and* as ((X\X)/X), X a variable over categories). Since these categories can themselves be quite complex (e.g., transitive verbs or VP modifiers, higher-order tensors), a potential objection to our approach is that conjunction of all but the most basic cases will be impossible to implement, or train.

We believe that the situation is actually a lot better than it seems. Even setting aside the discussion at the end of Section 3.5, which suggests that the computational problem could be reduced by a careful implementation, we could follow the approach in Rooth and Partee (1983) and propose that conjunction never applies to higher-order tensors. In this influential approach (see also Winter, 1996, 2001) the scope of coordination is not what it appears to be: V-conjunction in (25-a) is semantically converted into sentential conjunction (25-b), an operation on vectors in our system. Similarly, (26-a) would become (25-b), again vector conjunction.

(25) a. Dogs [chase and pester] cats.
b. [Dogs chase cats and dogs pester cats]

(26) a. A [long and fat] hot dog.
b. A [long hot dog and fat hot dog]

Additional reasons for optimism come from the observation that, when applied to predicates, conjunction in formal semantics boils down to a very simple operation—set intersection.[85] The DSM version of coordination could be equally simple; so simple, in fact, that we might discover that (some variation on) Mitchell and Lapata's componentwise vector multiplication might approximate it well enough to make training superfluous for those coordinated categories that do not involve tensed elements.[86]

If coordination became, in fact, an 'easy' case, the road would

[85] By predicates we mean, pretheoretically, any category that can appear after a copula (adjectives and non-quantificational noun phrases), and modifiers (attributive adjectives, PPs). The behavior of conjunction with DPs is more complex, and has prompted some linguists to distinguish between an intersective (Boolean) and a non-intersective (non-Boolean) conjunction. Again, see Zamparelli (2011) for references.

[86] As Mitchell and Lapata discuss, componentwise vector multiplication produces a sort of "component intersection" —only those dimensions that are significantly different from 0 in both input vectors will be significantly different from 0 in the output. Tensed cases like *"John arrived and Mary left"*, or *"John took the car and went to school"* imply a temporal sequence of two events, while conjunction of statives do not: *"John likes spaghetti and Mary loves sushi"*. We do not see how vector-mixture models could approximate these cases, and especially their difference: if they are symmetrical, they might at best get the second, if not, the first.

be open to a *decompositional approach* to those constructions that have conjunction as one of their subcomponents (examples are relative clauses, correlatives, adpositional structures), factoring out conjunction and dealing with the hopefully straightforward residual part. For instance, consider a relative clause, as in (27): in formal semantics, the brackets in this example would denote the intersection between the set of dogs and the set of cat-chasers.

(27) The [dogs which chase cats] barked.

Now, in many languages, relative clauses can function as full nominals ("free relatives", e.g., the constituents marked in (28)).

(28) [Whoever came] knew [what Bill feared most].

This suggests an alternative treatment for *which* with respect to the one we propose in Section 3, where this word is a function from matrices (verb plus object or transposed verb plus subject) to N vectors. To deal with free relatives, the same operation that in *"dogs chase cats"* applies to the nouns *dogs* and *cats*, turning them into full DPs (see Section 3.6), would now map the free relatives in (28) onto the DPs needed by *knew*. But (27) would involve combining the noun *dog* with the "pseudo-noun" *"which chase cats"* by means of a standard N conjunction (which in this case would be entirely structure-driven: no *and* is present between *dogs* and *"which chase cats"*). The DSM meaning of the relative pronoun would be quite simpler and the conjunction could be the same as the one trained for, say, *"a friend and colleague"*. The same *divide et impera* methodology could be applied to other cases, e.g., adverbial modifiers.

When DSMs were first developed for single words, they were tested on fairly basic lexical tasks, such as simulating word similarity judgments or spotting synonyms. However, in the two decades since these first experiments, the very same models have been applied to much more complex and arguably linguistically interesting tasks, such as predicting the selectional preferences of verbs or the qualia roles of nominal concepts (see Section 2.2). Analogously, we are currently testing our early compositional DSMs on relatively 'simple-minded' tasks such as paraphrase detection, but we are confident that the imagination of future researchers will find applications for these models in increasingly ambitious and linguistically interesting domains of semantics.

This paper is the beginning of a long journey. We hope that you, patient reader, will forgive us if its ambitious course still rests on empirical foundations that we and many others are just starting to verify. In the years to come, we will devote our energies to chart the land and to trim

the paths, many of which undoubtedly lead nowhere. We hope that if you found the ideas presented here stimulating enough to accompany us until this last paragraph, you will also join us in the exploration of this brave new world.

Acknowledgments

We owe some fundamental ideas to Edward Grefenstette and Emiliano Guevara. Edward came up with the stepwise method to estimate higher-order tensors and suggested tensor transposition to implement the rule we use for relative clauses in Section 3.6. Emiliano first proposed to use regression to learn compositional functions. We also thank Chris Barker, Gemma Boleda, Johan Bos, Peter Bosch, Stephen Clark, Georgiana Dinu, Katrin Erk, Stefan Evert, Eugenie Giesbrecht, Dimitri Kartsaklis, Graham Katz, Alessandro Lenci, Louise McNally, Stefano Menini, Ray Mooney, Sebastian Padó, Massimo Poesio, Martha Palmer, Denis Paperno, Jacopo Romoli, Chung-chieh Shan, Mark Steedman, Anna Szabolcsi, Peter Turney, Dominic Widdows, Fabio Massimo Zanzotto, the LILT editor and reviewers, the members of the inter-continental FLOSS reading group, the audience at CLIN 2012, Linguistic Evidence 2012, the UCL Cognition, Perceptual and Brain Sciences seminar, the EACL 2012 Compositionality in Distributional Semantics tutorial, LSD 2012 and KONVENS 2012 for a mixture of ideas, stimulating discussions, constructive criticism, implementation help and feedback on earlier versions of this paper. Of course, we thank also all the COMPOSES group for so many engaging discussions. Last but not least, we thank Annie Zaenen for her careful proof-reading of our draft. The work described in the article is currently being funded by the ERC 2011 Starting Independent Research Grant nr. 283554 (COMPOSES project).

References

Aitchison, Jean. 1993. *Words in the Mind*. Malden, MA: Blackwell.

Almuhareb, Abdulrahman and Massimo Poesio. 2005. Concept learning and categorization from the web. In *Proceedings of CogSci*, pages 103–108. Stresa, Italy.

Andrews, Mark, Gabriella Vigliocco, and David Vinson. 2009. Integrating experiential and distributional data to learn semantic representations. *Psychological Review* 116(3):463–498.

Apresjan, Yuri. 1974. Regular polysemy. *Linguistics* 142:5–32.

Asher, Nicholas. 2011. *Lexical Meaning in Context: A Web of Words*. Cambridge, UK: Cambridge University Press.

Axler, Sheldon. 1997. *Linear algebra done right, 2nd ed*. New York: Springer.

Bader, Brett and Tamara Kolda. 2006. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. *ACM Trans. Math. Software* 32:635–653.

Baker, Collin, Charles Fillmore, and John Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of COLING*, pages 86–90. Montreal, Canada.

Barker, Chris and Chung-Chieh Shan. 2006. Types as graphs: Continuations in type logical grammar. *Journal of Logic, Language and Information* 15:331–370.

Baroni, Marco, Eduard Barbu, Brian Murphy, and Massimo Poesio. 2010. Strudel: A distributional semantic model based on properties and types. *Cognitive Science* 34(2):222–254.

Baroni, Marco, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-Chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of EACL*, pages 23–32. Avignon, France.

Baroni, Marco and Alessandro Lenci. 2010. Distributional Memory: A general framework for corpus-based semantics. *Computational Linguistics* 36(4):673–721.

Baroni, Marco and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of EMNLP*, pages 1183–1193. Boston, MA.

Basile, Valerio, Johan Bos, Kilian Evang, and Noortje Venhuizen. 2012. Developing a large semantically annotated corpus. In *Proceedings of LREC*, pages 3196–3200. Istanbul, Turkey.

Bernardi, Raffaella and Michael Moortgat. 2010. Continuation semantics for the Lambek-Grishin calculus. *Information and Computation* 208(5):397–416.

Biemann, Chris and Eugenie Giesbrecht. 2011. Distributional semantics and compositionality 2011: Shared task description and results. In *Proceedings of the ACL Workshop on Distributional Semantics and Compositionality*, pages 21–28. Portland, Oregon.

Blacoe, William and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of EMNLP*, pages 546–556. Jeju Island, Korea.

Blei, David, Andrew Ng, and Michael Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Bloom, Paul. 2000. *How Children Learn the Meanings of Words*. Cambridge, MA: MIT Press.

Boleda, Gemma, Sebastian Padó, and Jason Utt. 2012a. Regular polysemy: A distributional model. In *Proceedings of *SEM*, pages 151–160. Montreal, Canada.

Boleda, Gemma, Eva Maria Vecchi, Miquel Cornudella, and Louise McNally. 2012b. First order vs. higher order modification in distributional semantics. In *Proceedings of EMNLP*, pages 1223–1233. Jeju Island, Korea.

Boole, George. 1854. *An Investigation of the Laws of Thought*. London: Walton and Maberly.

Bruni, Elia, Giang Binh Tran, and Marco Baroni. 2011. Distributional semantics from text and images. In *Proceedings of GEMS*, pages 22–32. Edinburgh, UK.

Bruni, Elia, Jasper Uijlings, Marco Baroni, and Nicu Sebe. 2012. Distributional semantics with eyes: Using image analysis to improve computational representations of word meaning. In *Proceedings of ACM Multimedia*, pages 1219–1228. Nara, Japan.

Budanitsky, Alexander and Graeme Hirst. 2006. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics* 32(1):13–47.

Bullinaria, John and Joseph Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods* 39:510–526.

Bullinaria, John and Joseph Levy. 2012. Extracting semantic representations from word co-occurrence statistics: Stop-lists, stemming and SVD. *Behavior Research Methods* 44:890–907.

Burgess, Curt. 2000. Theory and operational definitions in computational memory models: A response to Glenberg and Robertson. *Journal of Memory and Language* 43(3):402–408.

Cacciari, Cristina. 2012. Multiword idiomatic string processing: Many words in one? In J. Järviki, P. Pyykkönen, and R. Laine, eds., *From Words to Constructions: Structural and Semantic Complexity in Representation and Processing*. Berlin, Germany: Mouton de Gruyter. In press.

Carey, Susan and Elsa Bartlett. 1978. Acquiring a single new word. *Papers and Reports on Child Language Development* 15:17–29.

Carlson, Greg. 2009. Generics and concepts. In J. Pelletier, ed., *Kinds, Things and Stuff*, pages 16–35. Oxford, UK: Oxford University Press.

Chomsky, Noam. 1957. *Syntactic Structures*. Berlin, Germany: Mouton.

Chomsky, Noam. 1970. Remarks on nominalization. In R. Jacobs and P. Rosenbaum, eds., *Readings in English Trasformational Grammar*. Waltham, MA: Ginn-Blaisdell.

Cinque, Guglielmo, ed. 2002. *Functional Structure in DP and IP - The Carthography of Syntactic Structures*, vol. 1. Oxford University Press.

Cinque, Guglielmo. 2010. *The Syntax of Adjectives. A Comparative Study*. Cambridge, MA: MIT Press.

Clark, Stephen. 2013a. Type-driven syntax and semantics for composing meaning vectors. In C. Heunen, M. Sadrzadeh, and E. Grefenstette, eds., *Quantum Physics and Linguistics: A Compositional, Diagrammatic Discourse*, pages 359–377. Oxford, UK: Oxford University Press.

Clark, Stephen. 2013b. Vector space models of lexical meaning. In S. Lappin and C. Fox, eds., *Handbook of Contemporary Semantics, 2nd ed.*. Malden, MA: Blackwell. In press; `http://www.cl.cam.ac.uk/~sc609/pubs/sem_handbook.pdf`.

Clark, Stephen, Bob Coecke, and Mehrnoosh Sadrzadeh. 2008. A compositional distributional model of meaning. In *Proceedings of the Second Symposium on Quantum Interaction*, pages 133–140. Oxford, UK.

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

Clark, Stephen and Stephen Pulman. 2007. Combining symbolic and distributional models of meaning. In *Proceedings of the First Symposium on Quantum Interaction*, pages 52–55. Stanford, CA.

Clarke, Daoud. 2011. A context-theoretic framework for compositionality in distributional semantics. *Computational Linguistics* 1(54).

Coecke, Bob, Edward Grefenstette, and Mehrnoosh Sadrzadeh. 2013. Lambek vs. Lambek: Vector space semantics and string diagrams for Lambek Calculus. *Ann. Pure Appl. Logic* 164(11):1079–1100.

Coecke, Bob, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis* 36:345–384.

Cohen, Ariel. 2004. Generics and mental representations. *Linguistics and Philosophy* 27:529–556.

Copestake, Ann and Ted Briscoe. 1995. Semi-productive polysemy and sense extension. *Journal of Semantics* 12:15–67.

Copestake, Ann and Aurelie Herbelot. 2012. Lexicalised compositionality. `http://www.cl.cam.ac.uk/~ah433/lc-semprag.pdf`.

Curran, James, Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of ACL (Demo and Poster Sessions)*, pages 33–36. Prague, Czech Republic.

Curran, James and Marc Moens. 2002a. Improvements in automatic thesaurus extraction. In *Proceedings of the ACL Workshop on Unsupervised Lexical Acquisition*, pages 59–66. Philadelphia, PA.

Curran, James and Marc Moens. 2002b. Scaling context space. In *Proceedings of ACL*, pages 231–238. Philadelphia, PA.

Dagan, Ido, Bill Dolan, Bernardo Magnini, and Dan Roth. 2009. Recognizing textual entailment: rationale, evaluation and approaches. *Natural Language Engineering* 15:459–476.

Dinu, Georgiana and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceedings of EMNLP*, pages 1162–1172. Cambridge, MA.

Dumais, Susan. 2003. Data-driven approaches to information access. *Cognitive Science* 27:491–524.

Dummett, Michael. 1981. *Frege: Philosophy of language, 2nd ed*. Cambridge, MA: Harvard University Press.

Erk, Katrin. 2010. What is word meaning, really? (and how can distributional models help us describe it?). In *Proceedings of GEMS*, pages 17–26. Uppsala, Sweden.

Erk, Katrin. 2012. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass* 6(10):635–653.

Erk, Katrin and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of EMNLP*, pages 897–906. Honolulu, HI.

Erk, Katrin and Sebastian Padó. 2010. Exemplar-based models for word meaning in context. In *Proceedings ACL*, pages 92–97. Uppsala, Sweden.

Eslick, Ian. 2006. *Searching for Commonsense*. Ms thesis, MIT, Cambridge, MA.

Evert, Stefan. 2005. *The Statistics of Word Cooccurrences*. Ph.D dissertation, Stuttgart University.

Fellbaum, Christiane, ed. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Feng, Yansong and Mirella Lapata. 2010. Visual information in semantic representation. In *Proceedings of HLT-NAACL*, pages 91–99. Los Angeles, CA.

Firth, John R. 1957. *Papers in Linguistics, 1934-1951*. Oxford, UK: Oxford University Press.

Fletcher, William. 2004. Making the web more useful as a source for linguistic corpora. In *Corpus Linguistics in North America*, pages 191–205. Rodopi.

Fletcher, William. 2012. Corpus analysis of the World Wide Web. In C. Chapelle, ed., *Encyclopedia of Applied Linguistics*. Hoboken, NJ: Wiley-Blackwell.

Fodor, Jerry, Merrill Garrett, Edward Walker, and Cornelia Parkes. 1980. Against definitions. *Cognition* 8:263–367.

Foltz, Peter, Walter Kintsch, and Thomas Landauer. 1998. The measurement of textual coherence with Latent Semantic Analysis. *Discourse Processes* 25:285–307.

Frege, Gottlob. 1892. Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik* 100:25–50.

Frisson, Steven. 2009. Semantic underspecification in language processing. *Language and Linguistics Compass* 3(1):111–127.

Garrette, Dan, Katrin Erk, and Ray Mooney. 2013. A formal approach to linking logical form and vector-space lexical semantics. In H. Bunt, J. Bos, and S. Pulman, eds., *Computing Meaning, Vol. 4*, pages 27–48. Berlin: Springer.

Giesbrecht, Eugenie. 2010. Towards a matrix-based distributional model of meaning. In *Proceedings of the NAACL HLT 2010 Student Research Workshop*, pages 23–28. Los Angeles, CA.

Glenberg, Arthur and David Robertson. 2000. Symbol grounding and meaning: A comparison of high-dimensional and embodied theories of meaning. *Journal of Memory and Language* 3(43):379–401.

Grauman, Kristen and Bastian Leibe. 2011. *Visual Object Recognition*. San Francisco: Morgan & Claypool.

Grefenstette, Edward, Georgiana Dinu, Yao-Zhong Zhang, Mehrnoosh Sadrzadeh, and Marco Baroni. 2013. Multi-step regression learning for compositional distributional semantics. In *Proceedings of IWCS*, pages 131–142. Potsdam, Germany.

Grefenstette, Edward and Mehrnoosh Sadrzadeh. 2011a. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of EMNLP*, pages 1394–1404. Edinburgh, UK.

Grefenstette, Edward and Mehrnoosh Sadrzadeh. 2011b. Experimenting with transitive verbs in a DisCoCat. In *Proceedings of GEMS*, pages 62–66. Edinburgh, UK.

Grefenstette, Gregory. 1994. *Explorations in Automatic Thesaurus Discovery*. Boston, MA: Kluwer.

Griffiths, Tom, Mark Steyvers, and Josh Tenenbaum. 2007. Topics in semantic representation. *Psychological Review* 114:211–244.

Guevara, Emiliano. 2009. Compositionality in distributional semantics: Derivational affixes. In *Proceedings of the Words in Action Workshop*. Pisa, Italy.

Guevara, Emiliano. 2010. A regression model of adjective-noun compositionality in distributional semantics. In *Proceedings of GEMS*, pages 33–37. Uppsala, Sweden.

Harnad, Stevan. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena* 42(1-3):335–346.

Harris, Zellig. 1954. Distributional structure. *Word* 10(2-3):1456–1162.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning, 2nd edition*. New York: Springer.

Haxby, James, Ida Gobbini, Maura Furey, Alumit Ishai, Jennifer Schouten, and Pietro Pietrini. 2001. Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science* 293:2425–2430.

Heim, Irene and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Malden, MA: Blackwell.

Hills, Thomas. 2013. The company that words keep: comparing the statistical structure of child- versus adult-directed language. *Journal of Child Language* 40(03):586–604.

Hopper, Paul and Elizabeth Traugott. 1990. *Grammaticalization*. Cambridge, UK: Cambridge University Press.

Huth, Alexander, Shinji Nishimoto, An Vu, and Jack Gallant. 2012. A continuous semantic space describes the representation of thousands of object and action categories across the human brain. *Neuron* 76(6):1210–1224.

Jackendoff, Ray. 1990. *Semantic Structures*. Cambridge, MA: MIT Press.

Kamp, Hans. 1975. Two theories about adjectives. In E. Keenan, ed., *Formal Semantics of Natural Languages*, pages 123–155. Cambridge, UK: Cambridge University Press.

Kamp, Hans and Barbara Partee. 1995. Prototype theory and compositionality. *Cognition* 57(2):129–191.

Kartsaklis, Dimitri, Mehrnoosh Sadrzadeh, Stephen Pulman, and Bob Coecke. 2013. *Reasoning about Meaning in Natural Language with Compact Closed Categories and Frobenius Algebras*. Cambridge University Press. To appear.

Kelly, Colin, Barry Devereux, and Anna Korhonen. 2012. Semi-supervised learning for automatic conceptual property extraction. In *Proceedings of the 3rd Workshop on Cognitive Modeling and Computational Linguistics*, pages 11–20. Montreal, Canada.

Kintsch, Walter. 2001. Predication. *Cognitive Science* 25(2):173–202.

Kipper, Karin, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A large-scale classification of English verbs. *Language Resources and Evaluation* 42(1):21–40.

Klepousniotou, Ekaterini and Shari Baum. 2007. Disambiguating the ambiguity advantage effect in word recognition: An advantage for polysemous but not homonymous words. *Journal of Neurolinguistics* 20:1–24.

Kolda, Tamara and Brett Bader. 2009. Tensor decompositions and applications. *SIAM Review* 51(3):455–500.

Kotlerman, Lili, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering* 16(4):359–389.

Krifka, Manfred, Francis Pelletier, Gregory Carlson, Alice ter Meulen, Godehard Ling, and Gennaro Chierchia. 1995. Genericity: An introduction. In G. Carlson and F. Pelletier, eds., *The Generic Book*, pages 1–124. Chicago, IL: University of Chicago Press.

Kripke, Saul A. 1980. *Naming and Necessity*. Cambridge, Mass: Harvard University Press.

Kübler, Sandra, Ryan McDonald, and Joakim Nivre. 2009. *Dependency Parsing*. San Francisco: Morgan & Claypool.

Kwiatkowski, Tom, Sharon Goldwater, Luke Zettelmoyer, and Mark Steedman. 2012. A probabilistic model of syntactic and semantic acquisition from child-directed utterances and their meanings. In *Proceedings of EACL*.

Lahav, Ran. 1993. The combinatorial-connectionist debate and the pragmatics of adjectives. *Pragmatics and Cognition* 1:71–88.

Lambek, Jim. 1961. On the calculus of syntactic types. In R. Jakobson, ed., *Structure of Languages and its Mathematical Aspects*, pages 166–178. American Mathematical Society.

Lambrecht, Knud. 1996. *Information Structure and Sentence Form: Topic, Focus, and the Mental Representations of Discourse Referents*. Cambridge, UK: Cambridge University Press.

Landauer, Thomas and Susan Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review* 104(2):211–240.

Lenci, Alessandro. 2008. Distributional approaches in linguistic and cognitive research. *Italian Journal of Linguistics* 20(1):1–31.

Lenci, Alessandro. 2011. Composing and updating verb argument expectations: A distributional semantic model. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 58–66. Portland, OR.

Leong, Chee Wee and Rada Mihalcea. 2011. Going beyond text: A hybrid image-text approach for measuring word relatedness. In *Proceedings of IJCNLP*, pages 1403–1407.

Liang, Percy, Michael Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *Proceedings of ACL)*, pages 590–599.

Louwerse, Max. 2011. Symbol interdependency in symbolic and embodied cognition. *Topics in Cognitive Science* 3:273–302.

Lund, Kevin and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods* 28:203–208.

Manning, Christopher. 2011. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In *Proceedings of CICLing*, pages 171–189. Waseda, Japan.

Marcus, Mitchell, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: the Penn Treebank. *Computational Linguistics* 19(2):313–330.

McCarthy, Diana. 2009. Word Sense Disambiguation: An overview. *Language and Linguistics Compass* 3(2):537–558.

McDonald, Scott and Chris Brew. 2004. A distributional model of semantic context effects in lexical processing. In *Proceedings of ACL*, pages 17–24. Barcelona, Spain.

McDonald, Scott and Michael Ramscar. 2001. Testing the distributional hypothesis: The influence of context on judgements of semantic similarity. In *Proceedings of CogSci*, pages 611–616.

Mel'chuk, Igor A. 1987. *Dependency Syntax: Theory and Practice*. Albany: State University Press of New York.

Meyer, Carl. 2000. *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA: SIAM.

Miller, George and Walter Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes* 6(1):1–28.

Mitchell, Jeff and Mirella Lapata. 2008. Vector-based models of semantic composition. In *Proceedings of ACL*, pages 236–244. Columbus, OH.

Mitchell, Jeff and Mirella Lapata. 2009. Language models based on semantic composition. In *Proceedings of EMNLP*, pages 430–439. Singapore.

Mitchell, Jeff and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science* 34(8):1388–1429.

Mitchell, Tom, Svetlana Shinkareva, Andrew Carlson, Kai-Min Chang, Vincente Malave, Robert Mason, and Marcel Just. 2008. Predicting human brain activity associated with the meanings of nouns. *Science* 320:1191–1195.

Montague, Richard. 1970a. English as a Formal Language. In B. V. et al., ed., *Linguaggi nella Societaà e nella Tecnica*, pages 189–224. Edizioni di Comunità. Reprinted in Montague (1974), 188–221.

Montague, Richard. 1970b. Universal Grammar. *Theoria* 36:373–398.

Montague, Richard. 1973. The proper treatment of quantification in English. In K. e. a. Hintikka, ed., *Approaches to Natural Language*, pages 221–242. Dordrecht: D. Reidel.

Moortgat, Michael. 1997. Categorial Type Logics. In J. van Benthem and A. ter Meulen, eds., *Handbook of Logic and Language*, pages 93–178. Cambridge, MA: The MIT Press.

Murphy, Brian, Partha Talukdar, and Tom Mitchell. 2012. Selecting corpus-semantic models for neurolinguistic decoding. In *Proceedings of *SEM*, pages 114–123. Montreal, Canada.

Murphy, Gregory. 2002. *The Big Book of Concepts*. Cambridge, MA: MIT Press.

Navigli, Roberto. 2009. Word Sense Disambiguation: A survey. *ACM Computing Surveys* 41(2):1–69.

Newport, Elissa, Henry Gleitman, and Lila Gleitman. 1977. Mother, I'd rather do it myself: Some effects and non-effects of maternal speech style. In C. E. Snow and C. A. Ferguson, eds., *Talking to children : Language input and acquisition*, pages 109–150. Cambrdige: Cambridge University Press.

Ng, Vincent. 2010. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of ACL*, pages 1396–1411. Uppsala, Sweden.

Nivre, Joakim. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of IWPT*, pages 149–160. Nancy, France.

Padó, Sebastian and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics* 33(2):161–199.

Padó, Ulrike, Sebastian Padó, and Katrin Erk. 2007. Flexible, corpus-based modelling of human plausibility judgements. In *Proceedings of EMNLP*, pages 400–409. Prague, Czech Republic.

Pantel, Patrick and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of KDD*, pages 613–619. Edmonton, Canada.

Partee, Barbara. 2004. *Compositionality in Formal Semantics*. Malden, MA: Blackwell.

Pelletier, Francis Jeffry. 2001. Did Frege believe Frege's principle? *Journal of Logic, Language, and Information* 10:87–114.

Pereira, Fernando. 2000. Formal grammar and information theory: Together again? *Philosophical Transactions of the Royal Society* 385:1239–1253.

Poesio, Massimo, Simone Ponzetto, and Yannick Versley. 2010. Computational models of anaphora resolution: A survey. `http://clic.cimec.unitn.it/massimo/Publications/lilt.pdf`.

Prince, Alan and Paul Smolensky. 2004. *Optimality Theory*. Malden, MA: Blackwell.

Pustejovsky, James. 1995. *The Generative Lexicon*. Cambridge, MA: MIT Press.

Rapp, Reinhard. 2004. A freely available automatically generated thesaurus of related words. In *Proceedings of LREC*, pages 395–398. Lisbon, Portugal.

Reddy, Sravana and Kevin Knight. 2011. What we know about the Voynich manuscript. In *Proceedings of ACL Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH)*, pages 78–86. Stroudsburg, PA, USA: Association for Computational Linguistics.

Reinhart, Tanya. 1976. *The Syntactic Domain of Anaphora*. Ph.D dissertation, Massachusetts Institute of Technology.

Reisinger, Joseph and Raymond Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Proceedings of NAACL*, pages 109–117. Los Angeles, CA.

Riordan, Brian and Michael Jones. 2011. Redundancy in perceptual and linguistic experience: Comparing feature-based and distributional models of semantic representation. *Topics in Cognitive Science* 3(2):1–43.

Roller, Stephen, Michael Speriosu, Sarat Rallapalli, Benjamin Wing, and Jason Baldridge. 2012. Supervised text-based geolocation using language models on an adaptive grid. In *Proceedings of EMNLP*, pages 1500–1510. Jeju Island, Korea.

Rooth, Mats. 1985. *Associations with Focus*. Ph.D. thesis, University of Massachusetts at Amherst.

Rooth, Mats. 1995. A theory of focus interpretation. *Natural Language Semantics* 4.

Rooth, Mats and Barbara Partee. 1983. Conjunction, type ambiguity and wide scope *or*. In D. Flickenger, M. Macken, and N. Wiegand, eds., *Proceedings of WCCFL1*. Stanford, CA: Stanford Linguistic Association.

Sag, Ivan, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In A. Gelbukh, ed., *Computational Linguistics and Intelligent Text Processing*, pages 189–206. Berlin: Springer.

Sahlgren, Magnus. 2005. An introduction to random indexing. `http://www.sics.se/~mange/papers/RI_intro.pdf`.

Sahlgren, Magnus. 2006. *The Word-Space Model*. Ph.D dissertation, Stockholm University.

Sahlgren, Magnus. 2008. The distributional hypothesis. *Italian Journal of Linguistics* 20(1):33–53.

Schubert, Lenhart and Matthew Tong. 2003. Extracting and evaluating general world knowledge from the Brown corpus. In *Proceedings of the HLT-NAACL 2003 workshop on Text Meaning*, pages 7–13. Morristown, NJ.

Schütze, Hinrich. 1997. *Ambiguity Resolution in Natural Language Learning*. Stanford, CA: CSLI.

Schütze, Hinrich. 1998. Automatic word sense discrimination. *Computational Linguistics* 24(1):97–123.

Scott, Gary-John. 2002. Stacked adjectival modification and the structure of nominal phrases. In G. Cinque, ed., *Functional Structure in DP and IP. The Cartography of Syntactic Structures*, vol. 1. Oxford, UK: Oxford University Press.

Searle, John. 1980. Minds, brains and programs. *Behavioral and Brain Sciences* 3(3):417–457.

Silberer, Carina and Mirella Lapata. 2012. Grounded models of semantic representation. In *Proceedings of EMNLP*, pages 1423–1433. Jeju, Korea.

Smolensky, Paul. 1990. Tensor product variable binding and the representation of symbolic structures in connectionist networks. *Artificial Intelligence* 46:159–216.

Socher, Richard, Brody Huval, Christopher Manning, and Andrew Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP*, pages 1201–1211. Jeju Island, Korea.

Steedman, Mark. 2000. *The Syntactic Process*. Cambridge, MA: MIT Press.

Stowell, Timothy. 1981. *Origins of Phrase Structure*. Ph.D. thesis, MIT.

Strang, Gilbert. 2003. *Introduction to linear algebra, 3d edition*. Wellesley, MA: Wellesley-Cambridge Press.

Thater, Stefan, Georgiana Dinu, and Manfred Pinkal. 2009. Ranking paraphrases in context. In *Proceedings of the 2009 Workshop on Applied Textual Inference*, pages 44–47. Suntec, Singapore.

Titov, Ivan and Alexandre Klementiev. 2012. A bayesian approach to unsupervised semantic role induction. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*. Avignon, France.

Tomasello, Michael. 2003. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Cambridge, MA: Harvard University Press.

Turney, Peter. 2007. Empirical evaluation of four tensor decomposition algorithms. Tech. Rep. ERB-1152, NRC.

Turney, Peter. 2012. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research* 44:533–585.

Turney, Peter and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* 37:141–188.

Tversky, Amos and Daniel Kahneman. 1983. Extension versus intuitive reasoning: The conjunction fallacy in probability judgment. *Psychological Review* 90(4):293–315.

van Benthem, Johan. 1986. *Essays in Logical Semantics*. Dordrecht: Reidel Publishing Company.

Van de Cruys, Tim. 2010. A non-negative tensor factorization model for selectional preference induction. *Natural Language Engineering* 16(4):417–437.

Vecchi, Eva Maria, Marco Baroni, and Roberto Zamparelli. 2011. (Linear) maps of the impossible: Capturing semantic anomalies in distributional space. In *Proceedings of the ACL Workshop on Distributional Semantics and Compositionality*, pages 1–9. Portland, OR.

Wang, Chang and Sridhar Mahadevan. 2008. Manifold alignment using Procrustes analysis. In *Proceedings of ICML*, pages 1120–1127. Helsinki, Finland.

Werning, Markus, Wolfram Hinzen, and Edouard Machery, eds. 2012. *The Oxford Handbook of Compositionality*. Oxford University Press.

Widdows, Dominic. 2008. Semantic vector products: Some initial investigations. In *Proceedings of the Second Symposium on Quantum Interaction*. Oxford, UK.

Winter, Yoad. 1996. A unified semantic treatment of singular NP coordination. *Linguistics and Philosophy* 19:337–391.

Winter, Yoad. 2001. *Flexible Principles in Boolean Semantics*. The MIT Press.

Wittgenstein, Ludwig. 1953. *Philosophical Investigations*. Oxford, UK: Blackwell. Translated by G.E.M. Anscombe.

Zamparelli, Roberto. 2011. Coordination. In K. von Heusinger, C. Maienborn, and P. Portner, eds., *Semantics : an international handbook of natural language meaning*, pages 1713–1741. De Gruyter Mouton.

Zanzotto, Fabio M., Ioannis Korkontzelos, Francesca Falucchi, and Suresh Manandhar. 2010. Estimating linear models for compositional distributional semantics. In *Proceedings of COLING*, pages 1263–1271. Beijing, China.