Distributional semantics: A Montagovian view

Raffaella Bernardi
DISI, University of Trento

May 14, 2013

Contents

Abstract

This paper describes the current status of research in Distributional Semantics looking at the results from the Montagovian tradition stand point. It considers the main aspects of the Montagovian view as binoculars to observe those results, in particular: compositionality, syntax-semantics interface, logical words and entailment. To this end, it reviews some work that aims to tackle those issues within the Distributional Semantics Models and tries to highlight some open questions formal and distributional semanticists could address together.

Credits: Some of the material in the background section is based on distributional semantics talks by Marco Baroni, Stefan Evert, Alessandro Lenci and Roberto Zamparelli.

1 Introduction

This paper is not a research paper, no new results are reported. Its aim is to bridge two research communities working on related questions using different but compatible methods in order to profit of each other results. The main question they share is how we can formally capture natural language semantics. In other words, how can a computer processes linguistic expressions like "*Two men play a game*", "*Some people play chess*" and "*Some people play music*" and realize that the second sentence is semantically similar to the first, but not to the last one – e.g. the first two sentences can be the descriptions of the same image whereas the last one describes a different event, even though it shares several words with the other sentences. To answer this question, formal semanticists employ a logic framework and exploit its reasoning apparatus, whereas distributional semanticists look at how natural language is used by inducing statistical based representations and exploiting vector semantic space tools. Of course, none of the two communities has reached a final answer, but both have discovered interesting aspects of natural language that can possibly converge within an integrated enterprise. To reach our aim, we will first briefly introduce the core concepts at the heart of the two approaches (Section 2) and then look at distributional semantics with the eyes of formal semanticists (Section 3).

2 Background

In this section, we describe our standing point by briefly introducing the core concepts of Logic and its application to natural language analysis. We will then look at Distributional Semantics from these formal semantics pillars.

2.1 From logic to language: The Montagovian pillars

In Logic, the interpretation of a complex formula depends on the interpretation of the parts and of the logical operators connecting them (*compositionality*). The interpretation of the logical operators determines whether from a set of propositions a given proposition follows: $\{\psi_1, \ldots \psi_n\} \models \phi$. The entailment $\models$ is said to be *satisfiable* when there is at least one interpretation for which the premises and the conclusion are true; *falsifiable* when there is at least one interpretation for which the premises are true and the conclusion is false; and *valid* when the set of interpretations for which the premises are true is included in the set of interpretations for which the conclusion is true (*logical entailment.*) These two aspects have been used to formalize natural language meaning too. The starting point has been Frege's solution to the following puzzle: There is the star a called "venus", "morning star" and "evening star" that are represented in First Order Logic (FOL) by $\texttt{venus}', \texttt{morningst}', \texttt{eveningst}'$: $[\![\texttt{venus}']\!] = a$, $[\![\texttt{morningst}']\!] = a$ and $[\![\texttt{eveningst}']\!] = a$. a is the meaning (*reference*) of these linguistic signs. Checking whether it is true that (i) "*the morning star is the morning star*" or that (ii) "*the morning star is the evening star*" ends

up checking that (i) $[\![\texttt{morningst}']\!] = [\![\texttt{morningst}']\!]$ and (ii) $[\![\texttt{morningst}']\!] = [\![\texttt{eveningst}']\!]$, both of which reduce to checking $a = a$. But checking whether (i) "*the morning star is the morning star*" or that (ii) "*the morning star is the evening star*" cannot amount to the same operation since (ii) is cognitively more difficult than (i). Frege solved this puzzle by claiming that a linguistic sign consists of a *Bedeutung* (reference), the object that the expression refers to, and a *Sinn* (sense), mode of presentation of the reference. Moreover, he claimed that natural language meaning can be represented by a logical language.

Following Frege, formal semanticists' aim has been to obtain FOL representations of natural language expressions compositionaly. A crucial contribution to this research line has come from Montague (Montague, 1970), hence we can refer to the general framework as the Montagovian view. Formal semanticists have wondered what the meaning representation of the lexical words is, and which operation(s) put the lexical meaning representation together. The most largely shared view takes syntax to drive the order of composition. In particular, to assemble the syntactic structure Montague employed Categorial Grammar (CG) in which syntactic categories are seen as functions – $A\backslash B$ (or B/A), a function that wants an argument A on the left (resp., on the right) to return an expression of category B – and their composition as function application. The Categorial Grammar view has been further elaborated into a Logical Grammar by Lambek (1958), the general framework is known as Type Logical Grammar (Moortgat, 1997; Morrill, 1994). In it the connection between syntax and semantics has been tied up at both lexical and grammatical level as we will better see in the sequel. In brief, the core components of the Montagovian framework are:

Compositionality The meaning representation of a phrase depends on the meaning representation of its parts and the way they are put together.

Syntax-semantics interface The meaning representation assembly is guided by the derivational structure and a tight connection must be established between domains of interpretation and syntactic categories. This connection can be captured by defining a mapping between semantic types and syntactic categories.

Logical words and entailment Entailment between phrases consisting only of content words is model dependent (it corresponds to satisfiability), entailment between phrases consisting also of logical (grammatical) words is model independent (it corresponds to validity.)

In the following, first we introduce the general background at the heart of Distributional Semantics Models (DSMs), then we zoom into those models that account for compositionality in the light of the main issues summarized above

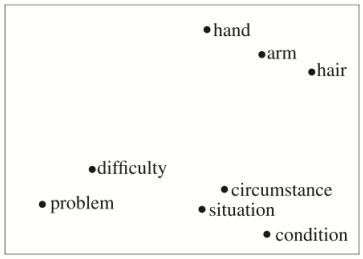
2.2 Distributional Semantics Models

As with any framework, in order to understand and appreciate the results achieved, the main research questions of the people working on the framework should be clear. For DSMs we can say the key questions have been the following ones: 1.) What is the sense of a given *word*?; 2.) how can the sense be induced and represented? and 3.) how do we relate word senses (synonyms, antonyms, hyperonym etc.)?[1] Well established answers supported by several evaluations are 1.) The sense of a word can be given by its use, viz. by the *contexts* in which it occurs; 2.) it can be induced from (either raw or parsed) corpora and can be represented by *vectors* (viz., tensors of order one); 3.) vector *cosine similarity* captures synonyms (as well as other semantic relations).

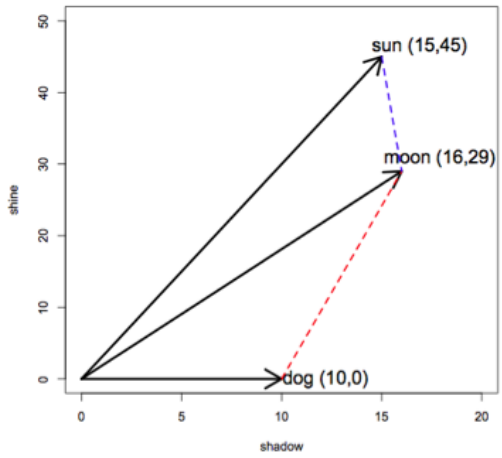
Today DSMs found their inspiration in ideas of the Fifties: First of all, Wittgenstein (1953) claims that word usage can reveal semantics flavor; Harris (1954) observed that words that occur in similar (linguistic) context tend to have similar meanings, Weaver (1955) looked at the applied side of these ideas by considering co-occurrence frequency of the context words near a given target word to be important for word sense disambiguation in machine translation tasks; and the famous slogan of the framework "you shall know a word by the company it keeps" is due to Firth (1957). Finally, Deerwster et al. (1990) put these intuitions at work. To easily capture the main intuition behind Firth's slogan, we can consider the example by McDonald & Ramscar (2001) who show how everyone can get the meaning of a made-up word like *wampimuk* by looking at the contexts in which it is used, for instance "*He filled the* wampimuk *with the substance, passed it around and we all drunk some*" and "*We found a little, hairy* wampimuk *sleeping behind the tree*" would suggest that *wampimuk* is a liquid in the first case and an animate thing in the second. Based on these kinds of observations, people have developed formal DSMs, implemented and evaluated them on several semantic tasks.

Definition A *Distributional Semantics Model* is a quadruple $\langle B, A, V, S\rangle$, where: B is the set of "basis elements" – the dimensions of the space; A is a lexical association function that assigns co-occurrence frequency of target words to the dimensions; V is an optional transformation that reduces the dimensionality of the semantic space; and S is a similarity measure. The results of the model can be depicted for instance by the picture below taken from Mitchell & Lapata (2010).

[1] The use of "sense" (as in Frege terminology) is not standard and may found opponents, but we believe it's useful to highlight the different perspective natural language is looked at within distributional and formal semantics models.



Toy example To better understand the main points, let us take as toy example vectors in a 2 dimensional space, such that $B = \{shadow, shine\}$; A= co-occurency frequency; and S the Euclidean distance. Let's take as target words: *moon*, *sun*, and *dog* and consider how often they co-occur with the basis elements:



The Euclidean distance shows that *sun* is "closer" to *moon* than to *dog*. The two dimensional space representation give $\overrightarrow{moon}$=(16,29), $\overrightarrow{sun}$= (15,45), $\overrightarrow{dog}$=(10,0) that live together in a space representation (a matrix, dimensions $\times$ target-words):

$$\begin{bmatrix} 16 & 15 & 10 \\ 29 & 45 & 0 \end{bmatrix}$$

The most commonly used representation is the transpose matrix: target-words $\times$ dimensions:

	shine	shadow
$\overrightarrow{moon}$	16	29
$\overrightarrow{sun}$	15	45
$\overrightarrow{dog}$	10	0

The dimensions are also called "features" or "contexts".

Standard DSMs In standard DSMs, words are taken to be all in the same space; the space dimensions are the *most k frequent words*, minus the "stop-words", viz. high-frequency words with relatively low information content, such as grammatical words (e.g. *of*, *the*, *and*, *them*, ...). Hence, they may be around 2k-30K or even more; and they can be plain words, words with their part of speech (PoS), words with their syntactic relation. Hence, a text needs to be: tokenized, normalized (e.g., capitalization and stemming), annotated with PoS tags (N, J, etc.), and if required also parsed (to extract the syntactic relations). Instead of plain counts, the values can be more significant weights of the co-occurrence frequency: *tf-idf* (term frequency (tf) $\times$ inverse document frequency (idf)): an element gets a high weight when the corresponding term is frequent in the corresponding document (tf is high), but the term is rare in other documents of the corpus (df is low, idf is high.) (Jones, 1972); or *PMI* (pointwise mutual information): measure how often two events x and y occur, compared with what we would expect if they were independent (Church & Hanks., 1989). Finally, the many dimensions can be reduced so to obtain a matrix of a lower dimensionality (a matrix with less – linearly independent – dimensions) by either Singular Value Decomposition (SVD) that generalizes over sparser surface dimension by capturing "latent dimensions" or Random Indexing that improves efficiency by avoiding constructing too large matrices when not necessary.

Observation I We can say that formal semanticists focused on "meaning" as *Bedeutung*, whereas distributional semanticists studied "meaning" as *Sinn*.

Observation II We can say that formal semanticists have been interested in entailment as validity (entailment driven by logical words), whereas distributional semanticists have looked more to entailment as satisfiability (ISA-relation between content words).

2.3 Distributionality from the Montagavian view

Below we review some state-of-the art approaches to DSM beyond lexical meaning in the light of the Montagavian pillars and the type-logical view that has been developed based on it. In particular, we will briefly look at Mitchell & Lapata (2010); Coecke et al. (2010); Socher et al. (2012) and more in depth at Baroni et al. (In press). The reader interested in a complete overview of DSMs is referred to Turney & Pantel (2010).

Compositionality All the work under consideration tackle the issue of compositionality and adopt the assumption that the meaning of the whole depends on the meaning representation of its parts. Mitchell & Lapata (2010) take all constituents to be represented by vectors which combine together to produce a new vector and investigate possible vector composition operations; they focus their evaluation of such operations by looking at small phrases consisting of a head and a modifier or complement, and consider the class of additive and

multiplicative operations to carry out the composition. As we will see below, the others instead use also tensors of order higher than vectors for capturing how a word can act on the word it composes with.

Syntax-Semantics interface The importance of taking the relation between the semantic composition and the syntactic structure into account is also discussed in all the work, though it is implemented in different ways and the strength of the connection varies from the very soft relation presented in Mitchell & Lapata (2010) and Socher et al. (2012) to the very tight one considered in Coecke et al. (2010); Baroni et al. (In press).

Mitchell & Lapata (2010) take syntax into account at least theoretically by presenting a flexible framework that covers under its umbrella several proposals. They consider the possibility of making the composition operation to depend on the syntactic relation. Formally, they define the result of the composition to be $\mathbf{p} = f(\mathbf{u}, \mathbf{v}, R, K)$ where R and K stand for the syntactic relation and the background knowledge, respectively. However, to simplify the implementation of the model, in practice they ignore K as well as the variety of function compositions based on the different syntactic relations. Moreover, they assume that the value vector $\mathbf{p}$ lies in the same space as $\mathbf{u}$ and $\mathbf{v}$. This essentially means that all syntactic categories correspond to semantic space of the same dimensionality. As the authors notice, the simplification may be too restrictive as it assumes that verbs, nouns, and adjectives are substantially similar enough to be represented in the same space, but it makes the implementation computationally simpler and the approach more feasible. Theoretically, they mention the possibility of considering the composition function to be asymmetric, for instance, as the action of a matrix, $\mathbf{U}$, representing one constituent, on a vector, $\mathbf{v}$, representing the other constituent: $\mathbf{p} = \mathbf{Cuv} = \mathbf{Uv}$ – as the authors notice, this is essentially Baroni & Zamparelli (2010)'s approach to adjective-noun composition to which we return below. Similarly, Socher et al. (2012) take a softer approach to the syntax-semantics interface and consider all words to have the same type of representation: a matrix and a vector. The matrix component expresses the ability of (any) word to act on another when composing with it, each matrix word is composed with the lexical vector of the other word, the result of such composition is still a pair of a vector and a matrix; the vector is obtained by projecting the two matrix product results to the lexical vector space and the matrix is produced by projecting the pairing of matrices back to the matrix space. Hence, the role of the syntax is reduced to the minimum, it just provides the structure of the composition. We will look at how Coecke et al. (2010) and Baroni et al. (In press) handle the syntax-semantics interface in Section 3.1

Logical words and entailment As emphasized by P. Bosch in his ESSLLI '08 talk, research on DSMs has traditionally focused on content words (open word class) whereas logical words (closed word class), like determiners, coordination, modals, prepositions have been neglected. However, if we aim to reach a compositional DSM able to capture the distributional meaning of sentences,

we might need to encode the contribution of e.g. *every* and *no* in the sentences "*few dogs chase cats*" vs. "*no dog chases cats*". Ed Hovey in his IWCS '11 talk discusses his intuitions regarding logical words, like negation of content words (*not hard*) and modal alteration of them (*possibly hard*) and claims that these expressions cannot be represented by tensors harvested from corpora but that they should be considered as operators: e.g. negation should negate the values of the tensors it composes with. Like Bosch and Hovey, we believe that researchers working on DSMs should go beyond lexical meaning and consider also phrases and sentences (a challenge that has been taken up by several research groups in the last few years), and we also believe that in this perspective it is time to consider grammatical words too (a challenge that is mostly overlooked); however, we raise doubts on Hovey's claim that considers grammatical words as pre-defined operators. In our view, the new question the DSM community might have to answer is whether from a distributional stand point there is a real distinction between grammatical and content words and if so to what extend. Do we really need to consider content words as given by their distribution and the grammatical words as pre-defined or do we instead need to change the distributional contexts to look at for capturing the meaning of the grammatical ones? We believe that a correct way to think of the issue should come from the observation of what leads a speaker to use for instance a determiner instead of another when expressing similar quantities (for instance, *few* vs. *a few*, *many* vs. *several*.) In the sequel (Section 3.2 and Section 3.3), we will review some preliminary work on this class of words within DSM.

3 The Montagovian pillars within DSM

3.1 Syntax-Semantics interface

Following the type logical view to the syntax-semantics interface, the connection between the two natural language levels needs to be captured by both the vocabulary and grammar rules; below we look at these two levels within DSMs.

3.1.1 Syntactic categories and semantic types

In the type-logical view, a first step to establish the tight formal connection between syntax and semantics is achieved by defining a mapping between syntactic categories and semantic types, based on the assumption that expressions belonging to the same syntactic categories find their meaning in the same denotational domains and hence receive meaning representations of the same types. For instance, if one assumes that determiner phrases (category: DP) denote in the domain of entities (type: e), and sentences (category: S) denotes in the domain of truth values (type: t), viz. $\texttt{Type}(DP) = e$ and $\texttt{Type}(S) = t$, and that $\texttt{Type}(A\backslash B) = \texttt{Type}(B/A) = \texttt{Type}(A) \to \texttt{Type}(B)$, then $\texttt{Type}(DP\backslash S) = e \to t$, and $\texttt{Type}((DP\backslash S)/DP) = e \to (e \to t)$.

This idea has been imported into the DSM realm in Clark et al. (2008) (and in the extended version in (Coecke et al., 2010)) where the authors assign

to a lexical entry the product between the pregroup category and the vector in the tensor space, using a mathematical structure that unifies syntax and semantics. The use of pre-groups as grammar to analyse linguistic structure traces back again to Lambek (Lambek, 1999, 2004). Clark (2012) discusses the same 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; Clark, 2012) 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 clearly explained in (Clark, 2012), 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. Several implementations of this framework have been proposed, e.g., (Grefenstette & Sadrzadeh, 2011a,b; Coecke et al., 2012; Kartsaklis et al., 2012), but the connection between the syntactic categories and semantics types has been maintained only in Grefenstette et al. (2013).

The mapping between syntactic categories and semantic type is fully emphasised and employed in Baroni et al. (In press). In the remaining of the paper, we will focus on this work. The authors generalize the distinction discussed in Baroni & Zamparelli (2010) between vectors (atomic categories, e.g., nouns) and matrices (one-argument function, e.g., adjectives) starting, as in the type-logical view, from defining a mapping from syntactic categories to semantic types, as specified below.[2]

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

In denotational semantics the semantic types indicate the type of the domain of denotation (for instance, *john* is of type e: it denotes in the domain of entities, D_e, whereas *walks* is of type $e \rightarrow t$ and denotes in the corresponding domains of functions from entities to truth values, $D_{e \rightarrow t}$); in distributional semantics Baroni et al. (In press) take types to stand for the semantics space in which the expression lives, namely the contexts or context transformations. Words that live in the space of vectors have an atomic type, whereas functional types are assigned to words that act as space mappings (context transformations): matrices (that is, second order tensors) have first order 1-argument functional types, third order tensors have first order 2-argument functional types,

[2] Baroni et al. (In press) adopt the alternative notation: $\texttt{Type}(B\backslash A) = \texttt{Type}(B/A) = C_A \rightarrow C_B$.

etc. In general, they assume that words of different syntactic categories live in different semantic spaces. As it is the case in formal semantics, where nouns and verb phrases are both functions from entities to truth values, one could decide that two different syntactic categories are mapped to the same semantic types – live in the same semantic space. Baroni et al. (In press) take as atomic categories N (noun), DP (determiner phrase) and S (sentence); their types are indexed to record the number of dimensions of the corresponding semantic space: $\texttt{Type}(N) = C_{n_i}, \texttt{Type}(DP) = C_{dp_j}, \texttt{Type}(S) = C_{s_k}$ – where C stands for context – whereas the types of the complex categories are obtained by the definition above.

Again following Montague, Baroni et al. (In press) consider a fragment of English that represents the variety of tensor composition the DSM should be able to cover both theoretically and practically. As vocabulary, they consider words in the syntactic categories listed in the table below. For sake of clarity, in the table next to the syntactic category we indicate also the corresponding semantic type as well as the order of the corresponding DS representation. In the sequel, following the standard practice, we will be using boldface lowercase letters, e.g., $\mathbf{a}$, to represent a vector, boldface capital letters, e.g., $\mathbf{A}$, to represent a matrix and Euler script letters, e.g., $\mathcal{X}$, to represent tensors of order higher than two.

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

Lexicon			
Syn Cat	CG Cat	Semantic Type	Tensors
N	N	C_{n_i}	I vector (1st ord.)
NNS	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	$DP\backslash S$	$C_{dp_j} \to C_{s_k}$	$K \times J$ matrix (2nd ord.)
TV	$(DP\backslash S)/DP$	$C_{dp_j} \to (C_{dp_j} \to C_{s_k})$	$(K \times J) \times J$ (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	$(N\backslash N)/(DP\backslash S)$ $(N\backslash N)/(S/DP)$	$(C_{dp_j} \to C_{s_k}) \to (C_{n_i} \to C_{n_i})$	$(I \times I) \times (K \times J)$ (higher ord.)

Table 1: *Syntax-Semantics interface of an English Fragment*

Before going to look at how the relation between syntax and semantics is captured at the grammar rules level, we will still report some observation regarding CG categories within DSMs.

Empirical Coverage Paramita (2012) has analysed two large corpora, Wikipedia and ukWaC[3] parsed with a Combinatorial Categorial Grammar (CCG) parser (Clark & Curran, 2007; Honnibal et al., 2007) aiming to understand which type of categories, hence tensors, are more frequent in natural language structures. From this analysis it results that in Wikipedia there are around 902M (ukWaC: around 1.8M) tokens belonging to an atomic category (vector); around 632M (ukWaC: around 1.4M) tokens belonging to a one-argument function category (matrices); around 114M (ukWaC: around 282M) tokens belonging to a two argument function category (3rd order tensor), and around 189M (uKaWac: 469M) tokens belonging to a tensor higher than 3; hence the large majority of tokens (around 90% in Wikipedia and 40% in ukaWaC) would be represented by a tensor of the order discussed in Baroni et al. (In press) and reviewed above.

Learning the vocabulary The vector representations of words belonging to atomic categories are obtained as explained above by harvesting the co-occurrence frequency and possibly converting them by means of some weighting schema. For the distributional functions, Baroni & Zamparelli (2010) propose to use regression methods. They look at adjective noun phrases, ADJ N, which again belong to the category of nouns and hence are represented by vectors as the modified noun. In other words, the distributional function is learned from examples of its input and output vectors extracted from the corpus; for instance, the matrices **RED** will be learned from vector pairs like (**army**, **RED army**), (**apple**, **RED apple**), etc. Standard machine learning methods are used to find the set of weights in the matrix that produces the best approximations to the corpus-extracted example output vectors when put together with[4] the corresponding input vectors. This method has been generalized to work with n-argument functions in Grefenstette et al. (2013). In particular, when a function returns another function as output (e.g., it acts on a vector and generates a matrix) we need to apply a multiple-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. Grefenstette et al. (2013) have worked on transitive verbs. A transitive verb such as *eat* is a third-order tensor (e.g. $(2 \times 4) \times 4$ tensor, that takes an object, a DP represented by a 4-dimensional vector (e.g., *cake*) to return the corresponding VP (*"eat cake"*, a 2×4 matrix). To learn the weights of such tensor, Grefenstette et al. (2013) 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 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"*), they use pairs of corpus-derived object vectors and the corresponding verb-object matrices estimated in

[3] Wikipedia English articles: around 820 million words, and ukWaC: around 2 billion words.
[4] We will see that the composition operation used is the product.

the first step as input-output examples in a second regression step, where the verb tensor components are determined.

CG category based DSMs As we have mentioned above, the dimensions of standard DSMs have been taken to be words tagged with PoS tags or words labeled with dependency relations. Differently from this tradition, Paramita (2012) exploits the richness of the CG categories to build a DSM model harvested from the large corpora parsed with the CCG parser mentioned above. We briefly report the results obtained. The model (CCG-DSM) has the 20K most frequent CG categories tagged words as dimensions, and the 10K most frequent nouns, 5K most frequent verbs, 5K most frequent adjectives as target words. The co-occurrence matrix harvested from the corpus has been converted by means of different weighting schema and reduced to 300 dimension by SVD. The model has been evaluated against a noun and verb clustering task as proposed in (Baroni et al., 2008). Interestingly, the CCG-DSM model outperforms both the one based on plain PoS-tagging and the one based on dependency relation-tagging in clustering verbs. The data-set, used for the evaluation, contains 45 verbs divided into five classes/clusters, viz. cognition: 10, motion: 15, body: 10, exchange: 5, change state: 5. The clustering has been done using CLUTO and evaluated with the standard clustering measures of entropy (clusters' level of disorder) and purity (proportion of the most frequent class in the cluster). The best performing results have been obtained with the Exponential Point-wise Mutual Information (epmi) weighting schema and the 2 window context (the 2 words on the left and the 2 words on the right of the target word). The measures are: entropy 0.305 (CCG-DSM) vs. 0.556 (dependency-DSM), purity 0.756 (CCG-DSM) vs. 0.667 (dependency-DSM). These results on the one hand confirm that the syntactic structure (encoded in the CG categories) plays a role in the distributional meaning of words, and on the other show that CG categories do carry important semantic information too.

3.1.2 Lambek's lesson: Function application and also Abstraction

As we explained earlier natural language expressions can correspond to first order or higher-order functions and can require one or more argument. Moreover, at the syntactic level, functions are directional ($A\backslash B$ vs. B/A), since in natural language function-argument order matters. Hence, CG and the type-logical grammar based on it consist of 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.)

Function application has been the main focus of several work aiming at combining CG-like syntax with DSMs. As mentioned above Clark et al. (2008); Coecke et al. (2010); Baroni & Zamparelli (2010) have been among the pioneers of such enterprise. As anticipated earlier, Baroni & Zamparelli (2010) look at how an adjective modifies a noun by employing the matrix-by-vector product (see below) that allows a matrix (the adjectives, ADJ) to act on a vector (the noun, N) resulting in a new vector (a new noun, ADJ N). Interestingly, the

authors show a great advantage of the DSM over the Formal Semantics one when dealing with composition of content words, namely they show that the same adjective modifies their argument differently accordingly to which is the noun it composes with (for instance, "*red apple*" vs. "*red army*".) However, the authors, by focusing on the adjective-noun constructions, do not consider the variety of syntactic-semantics constructions natural language exhibits. Baroni et al. (In press) extend the approach to further cases generalizing the matrix-by-vector composition to handle n-argument functions and follow the type-logical framework exploiting the correspondence between Lambek and Lambda calculi. Again, we will report on this work and refer the reader to the cited bibliography for related work.

One of the earlier contribution of Lambek mathematical view to the natural language parsing problem is the discovery of the inverse rule of function application, namely *abstraction*. Lambek highlighted that if a structure can be composed, it can also be de-composed, in other words if one knows that $w_1 : B, w_2 : B\backslash A$ yields $w_1\ w_2 : A$ she also knows that e.g. $w_1 : A/(B\backslash A)$. Hence, the Lambek calculus, in the natural deduction format, consists of both elimination (function application – composition) and introduction (abstraction – de-composition) rules of the implicational operators ($\backslash$ and $/$). A restricted version of abstraction (type raising) is also present in the CG combinatory version, CCG (Steedman, 2000) together with other function composition rules.[5]

In the type-logical framework, the syntactic trees (derivations) are labelled with lambda terms that record the operational steps and are therefore called "proof terms". Once the proof term of a parsed sentence is built, it can be replaced with the corresponding semantic representation of the lexical bits in the linguistic structure parsed. In a Montagovian view they will be replaced with λ-terms standing for the denotation of the words, in Continuation Semantics they would be replaced with λ-terms that take context into account (see Bernardi & Moortgat (2010); Barker & Shan (2006)). In DSM, Baroni et al. (In press) propose to replace them with the corresponding tensors. Below we will see this system at work on some toy examples.

Function application in DSM Baroni et al. (In press) propose to use "generalized matrix-by-vector multiplication" to account for function application

[5] From the empirical coverage study conducted in Paramita (2012) it results that most of the sentences in the Wikipedia and ukWaC need only forward application (Wikipedia: around 3M and ukWaC: around 2.6M), backward application (Wikipedia: around 233K and ukWaC: around 391K), or combination of them: Wikipedia: 25M (uKWaC: 48M); hence totally around 28M sentences in Wikipedia (ukWaC: 51M) would require the generalized matrix-by-vector composition Baroni et al. (In press) in a rather straight-forward way; around 2.5M (ukWaC: 4.7M) sentences are parsed also with function composition (forward or backward) and around 5.8M (ukWac: 15M) sentences require also backward crossed composition. Furthermore, there are 18M sentences in Wikipedia and 40M in ukWac that require the conjunction rule, 149K sentences in Wikipedia and 494K sentences in ukWaC that require generalized backward crossed composition, and 800K sentences in Wikipedia and 3M sentences in ukWaC that require the type-raising rule. Of course these numbers are subject to possible mistakes of the parser.

defined as below and explained by means of examples in the sequel. 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:

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

The term of the operation is used by the authors to underline the fact that the general product operation they assume 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, they 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). They 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). The simple matrix-by-vector multiplication is used straight forwardly to apply a first-order function to an argument:

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

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. This is the rule used in Baroni & Zamparelli (2010) to account for the composition of an adjective with a noun. Let us assume 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). See the toy example in Table 2: suppose *old* is associated to the toy matrix and applied to the *dog* vector, it returns the vector for *old dog*:

As observed in Baroni et al. (In press), 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

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

Table 2: *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).*

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.

We can now look at the function application cases required by the fragment of English whose vocabulary is presented in Table 1.

(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*)

For instance, when parsing the expressions *dogs bark*, *dogs chase cats* and *which chase cats*, CG produces the structures and terms in the trees of Figure 1. To help reading the proof term, we use the @ symbol to indicate the application of a function to an argument ($f@a$).

Baroni et al. (In press) replace the variables with the corresponding DSM representations obtained as described above and compute the vectors representing the sentences. In particular, in the (a) tree X and Y are replaced with the matrix **BARK** and the vector **dogs**, respectively, giving **BARK** $\times$ **dogs**; whereas in the (b) tree 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 ($\mathcal{CHASE} \times$ **cats**) $\times$ **dogs**. Similarly, for (c1) $\mathcal{WHICH} \times (\mathcal{CHASE} \times$ **cats**). Once we have built the meaning representation of the sentence, we can compute its meaning by means of generalized matrix-by-vector multiplication introduced above.

Below, we simplify the problem by using a toy scenario in which sentences live in a two dimensional space (C_{s_2}), determiner phrases in a four dimensional space (C_{dp_4}), and nouns into a three dimensional space (C_{n_3}). As said above, we have three cases to consider.

(a) Matrix vector composition Matrix vector composition can be exemplified by the operation composing a determiner phrase and an intransitive verb, as in the sentence before *dogs bark*. The CG labeled syntactic tree of this sentence

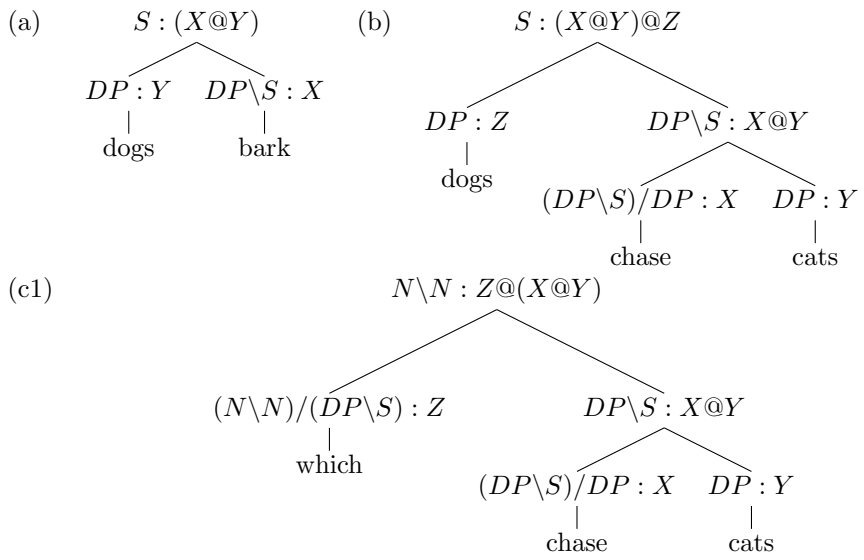


Figure 1: *Proof terms: Function application*

gives us **BARK dogs**. Since in our toy semantic space scenario the semantic type of an intransitive verb is $C_{dp_4} \to C_{s_2}$ and of a determiner phrases is C_{dp_4}, we take **BARK** and **dog** to be a 2×4 matrix and a 4-dimensional vector, respectively; these terms are composed simply by function application which returns a 2-dimensional vector standing for the meaning of the whole sentence.

Below we represent the *dp* contexts as dp1, dp2, dp3, dp4 and similarly the two *s* contexts as s1, s2.

$DP\backslash S$: matrix

bark	dp1	dp2	dp3	dp4
s1	n_{11}	n_{12}	n_{13}	n_{14}
s2	n_{21}	n_{22}	n_{23}	n_{24}

DP: vector

	dogs
dp1	k_1
dp2	k_2
dp3	k_3
dp4	k_4

S: vector

	dogs bark
s1	$(n_{11}, n_{12}, n_{13}, n_{14}) \cdot (k_1, k_2, k_3, k_4)$
s2	$(n_{21}, n_{22}, n_{23}, n_{24}) \cdot (k_1, k_2, k_3, k_4)$

$=$

S: vector

	dogs bark
s1	$(n_{11} \times k_1) + \ldots + (n_{14} \times k_4)$
s2	$(n_{21} \times k_1) + \ldots + (n_{24} \times k_4)$

(b) 3rd order tensor composed with two vectors An example of this case is provided by the composition of a transitive verb with its object and subject. For instance, for the sentence *dogs chase cats*, CG produces the labeled syntactic tree seen above which gives us the DS representation $(\mathcal{CHASE} \times \textbf{cats}) \times \textbf{dogs}$. Hence, we need to apply step-wise the 3rd order tensor, the transitive verb,

to two vectors, the object and the subject. In our toy example, the transitive verbs have semantic type $C_{dp_4} \rightarrow (C_{dp_4} \rightarrow C_{s_2})$. Hence, the DS meaning representation of *chase* is a $(2 \times 4) \times 4$ tensor; we can think of it as tensor of four slices of one 2×4 matrix each.

chase

	slice 1: dp1	dp2	dp3	dp4
s1	n^1_{11}	n^1_{12}	n^1_{13}	n^1_{14}
s2	n^1_{21}	n^1_{22}	n^1_{23}	n^1_{24}

$\ldots$

	slice 4: dp1	dp2	dp3	dp4
s1	n^4_{11}	n^4_{12}	n^4_{13}	n^4_{14}
s2	n^4_{21}	n^4_{22}	n^4_{23}	n^4_{24}

The application of *chase* to *cats* gives the following 2×4 matrix:

chase cats	dp1	$\ldots$	dp4
$s1$	$(n^1_{11}, n^1_{12}, n^1_{13}, n^1_{14}) \cdot (k_1, k_2, k_3, k_4)$	$\ldots$	$(n^4_{11}, n^4_{12}, n^4_{13}, n^4_{14}) \cdot (k_1, k_2, k_3, k_4)$
$s2$	$(n^1_{21}, n^1_{22}, n^1_{23}, n^1_{24}) \cdot (k_1, k_2, k_3, k_4)$	$\ldots$	$(n^4_{21}, n^4_{22}, n^4_{23}, n^4_{24}) \cdot (k_1, k_2, k_3, k_4)$

which can then be applied to the 4 dimensional vector representing *dogs* yielding a 2 dimensional vector representing the whole sentence.

(c) Higher order-tensor matrix composition The only higher-order tensor in Table 1 is the one representing a relative pronoun. From a formal semantic point of view, a relative pronoun creates the intersection of two properties. e.g. $[\![\text{dog}]\!] \cap [\![\text{chase cats}]\!]$; in distributional semantics, we can look at it as a 4th order tensor whose first argument is a verb phrase, hence a matrix, and its second argument is a noun, hence a vector, and it yields a modified noun, hence again a vector. In our toy example, it lives in a $(3 \times 3) \times (2 \times 4)$ space, it is of semantic type $(C_{dp_4} \rightarrow C_{s_2}) \rightarrow (C_{n_3} \rightarrow C_{n_3})$ and can be applied to a 2×4 matrix. For instance *which* can be applied to the matrix obtained above *chase cats*. As explained in Baroni et al. (In press), this operation can be reduced to the simpler one considered above, namely to the application of a 3rd order tensor to a vector. To this end, Baroni et al. (In press) apply the *unfolding* method that transforms a tensor into one of lower order by reordering its elements. There are several ways of reordering the elements, for instance, a $(2 \times 3) \times 4$ tensor can be arranged as a 6×4 matrix. Which mode is chosen is not important as far as across related calculations the same mode is used. Going back to our linguistic example, the relative pronoun and the VP-matrix could be unfolded into a $(3 \times 3) \times 8$ tensor and a 8 dimensional vector, respectively. To understand the unfolding method, let us look at how it could transform the 2×4 VP-matrix into a 8 dimensional vector and let us take as unfolding mode the concatenation of its elements as illustrated below. Let us assume the matrix representing *chase cats* represented abstractly above is instantiated as below; by unfolding we obtain the corresponding vector as following:

chase cats	dp1	dp2	dp3	dp4
$s1$	1	3	5	7
$s2$	2	4	6	8

$\rightsquigarrow_{\text{unfolding}}$ $(1, 2, 3, 4, 5, 6, 7, 8)$.

The evaluation carried out so far using generalized matrix-by-vector operation for function application has obtained encouraging results. See Vecchi et al. (2011); Grefenstette et al. (2013) for the evaluation of the compositional distributional models for adjective-noun and transitive verb-object composition, respectively and Pham et al. (2013) for an evaluation of (Baroni et al., In press)'s approach at sentential level. No evaluation has been carried out yet on relative sentences, these constructions are going to be object of investigation of the COMPOSES project.[6]

Abstraction in DSMs Abstraction is used mostly for two cases: long distance dependency and inverse scope. The latter is more a challenge for the formal grammar researchers than for the semanticists: once the grammar provides the right representations of an ambiguous sentence the semantic operations should be able to compute the proper meaning straight forwardly. Hence, in the following we will look only at long distance dependencies as instances of abstraction, and in particular at the case of relative pronoun that extracts the object of the relative clause sentence.

As far as we know, the first attempt to handle cases of long distance dependencies within the compositional distributional semantic framework is presented in Baroni et al. (In press) where the authors discuss the dependency of a main sentence subject from the transitive verb of a relative clause, e.g., "*A cat which dogs chase runs away*": the object of the relative clause is missing and its role is played by "*A cat*" thanks to the presence of the relative pronoun *which*. The lack of the object can be marked by a trace "*(A cat (which dogs chase ...)) runs away*". The type-logical view on the composition of this sentence can be represented by the tree below (Figure 2) that starts by assuming an hypothetical object (hyp), builds the sentence *dogs chase hyp* (Figure 2, tree on the left) and then withdraws the hypothesis building a tree without it (Figure 2, tree on the right) using abstraction. The application of abstraction is governed by the presence of the higher-order two-argument category (N\N)/(S/DP) assigned to the relative pronoun; it 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 and build the tree of category S/DP. The rule that allows this step is the one-branch rule encoding hypothetical reasoning. The derivation can then proceed by function application. 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

[6] `http://clic.cimec.unitn.it/composes/`

of a function (W of type $(e \to t) \to ((e \to t) \to t)$) to an argument (the lambda term of type $(e \to t)$ we have just built).

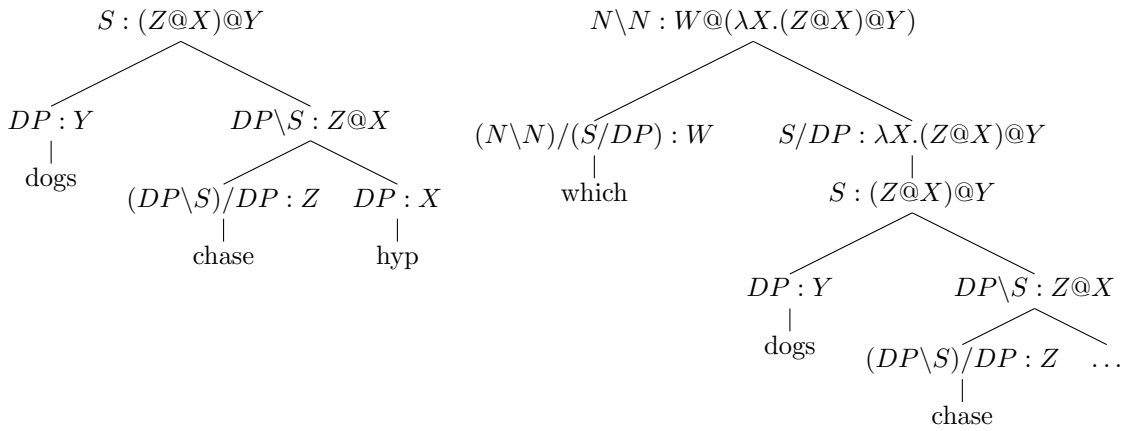


Figure 2: *Proof terms: function application and abstraction: hypothetical reasoning*

Syntactically, these constructions challenge any formal grammars, hence they have attracted the attention of researchers and several solutions have been proposed within the CG framework. Baroni et al. (In press) build the syntactic tree in the way more straightforwardly linked to the distributional semantics analysis even though the rule employed (Figure 3), namely associativity, would cause over-generation problems.[7]

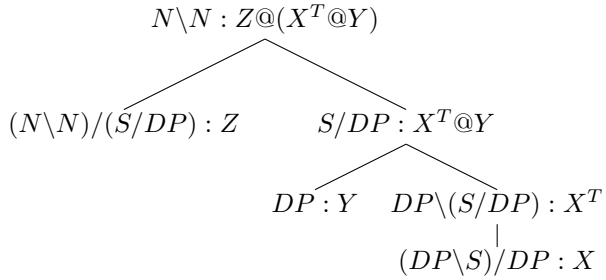


Figure 3: *Proof terms: associativity (syntax) and transformation (semantics)*

The proof term consists of just function application of the kinds we have discussed above. The only step here to introduce is the one that transforms a transitive verb category from $(DP\backslash S)/DP$ into $DP\backslash(S/DP)$. Syntactically, this step corresponds to associativity, semantically it corresponds to the tensor

[7]Its application could be controlled by employing for instance the multi-modal version of CG Moortgat (1997), but since our focus is on the composition of the distributional semantic representations involved in such constructions, we will overlook the syntactic issues. This semantic analysis or a variation of it could be connected to the different syntactic proposals in the literature.

transformation rule which establishes in general terms how the elements of a tensor can be switched:

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

This rule allows 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, this semantic rule is presented as taking the term X and yielding the term X^T. Now the proof term can be replaced with the actual distributional representation, obtaining $\mathcal{WHICH} \times (\mathcal{CHASE}^T \times \mathbf{dogs})$, which can later modify the vector representing *cat*.

Let's now assume as toy example that a transitive verb is a $(3 \times 2) \times 2$ matrix; for instance, *chase* consists of the two slice tensor below; it can be transformed by switching the second and first column of the first and second slice, respectively. This guarantees that ($\mathcal{CHASE}$ **dogs**) **cats** is equivalent to ($\mathcal{CHASE}^T$ **cats**) **dogs** as the reader can test by herself.

$\mathcal{CHASE}$	slice 1:		slice 2:		$\rightsquigarrow$	$\mathcal{CHASE}^T$	slice 1:		slice 2:	
	1	4	7	10			1	7	4	10
	2	5	8	11			2	8	5	11
	3	6	9	12			3	9	6	12

The use of the tensor transformation rule above avoid having to use the counterpart of abstraction, and does not provide a way for handling structure de-composition, as a more general solution might require. Hence, we wonder whether as Lambek highlighted the possibility of extending CG with abstraction, DSM framework can be extended with an equivalent rule.

We conjecture that a solution to long distance dependency could come from the categorical view presented in Preller & Sadrzadeh (2009), where the authors discuss the "eta maps" showing that they create Bell states that produce an extra space allowing for "teleportation", in other words the eta maps enable the information to flow between the quantum states that are not locally close. For instance, in the case of a negative transitive sentence "*John does not like Mary*" *does* and *not* are vectors that act as identity and as base swapper, respectively. The eta maps move the subject vector *john* to be the argument of the transitive verb so that the *does* and *not* vectors act on the representation of the positive transitive sentence swapping its bases, namely making it true if it was false, and vice versa.

Alternative solutions could come from the regression method discussed above. Following Montague's intuition regarding type lifting, namely that an expression like a personal name can be represented either as an object (a constant of type e) or as a set of those properties that hold for that object (a second order function of type $((e \to t) \to t)$), we could assume that an expression of syntactic category DP can be represented semantically either by a vector or

by a higher order tensor obtained from the corresponding vector by means of some tensor transformation rule. This rule could be learned by regression: we could induce from the corpus the vectors of a quantifier phrase (DET N, e.g., "*some dog*"), learn the tensor representing the same expression by regression, e.g. given the input pair (*runs*, "*some dog runs*"), learn the function representing "*some dog*", and then learn the transformation rule from the induced vector representing "*some dog*" to its higher order tensor representation learned by regression; or we could find ways to exploit the CCG parsed corpus for learning the type-raising rule from those structures in which it has been used.

Question In short, the challenge we are posing to the compositional distributional semantic community is to handle the semantic composition of not-juxtaposed expressions.

3.2 From logical entailment to entailment in DSM

In the brief review of the core logical concepts behind formal semantics we have highlighted the difference between satisfiability (an entailment that holds in a particular interpretation) and validity (an entailment that holds across all the interpretation). The latter is due to the presence of logical words, like determiners, whose meaning is independent from a specific interpretation. Hence, in the following before reviewing the work on entailment within DSMs, we are going to present DSMs views on logical (grammatical) words.

3.2.1 Grammatical words as pre-defined logical operators

Garrette et al. (2011) proposes an interesting hybrid framework combining the complementary strengths of FOL and DSM, namely the expressivity of the former and the flexibility of the latter. The authors use FOL for representing the logical structure of a sentence, and DSM for capturing the content words meaning and project lexical inferences from the vector space to logical form. In a nutshell, based on the idea that distribution similarity between expressions A and B corresponds to substitutability of B in the place of A, they generate inference projection rules like $\forall x.\texttt{Car}(x) \rightarrow \texttt{Vehicle}(x)$ using WordNet as a filter to validate such axioms. Then, FOL sentence representations are obtained using Boxer (Curran et al., 2007). Given a pair of sentences (Prem, Hyp), they check whether Hyp could follow from Prem (Prem $\rightarrow$ Hyp) from a DS view by checking (i) if there are inference projection rules between the words occurring in the two sentences and (ii) contextualising such rules by assigning them a weight: given a rule that relates a word w_1 in Hyp with a word w_2 in Prem, the weight is obtained by computing the similarity of w_1 with the sum of the vectors of the words co-occurring with w_2 in Prem (the context of w_2.) In the simplified setting described in Garrette et al. (2011), the authors generate a potential alignment between any pair of words, within the two sentences, that are related (synonymous or hyponym/hypernym up to a certain distance) in

WordNet, which also means that they have to be of the same syntactic category. Moreover, they currently only deal with single-word paraphrases. Finally, they use Markov Logic Networks for reasoning on FOL weighted clauses. For instance, given the projection rule (1) $\forall x.\texttt{Car}(x) \rightarrow \texttt{Vehicle}(x)$, by FOL they can infer that given $\neg\exists x.\texttt{Vehicle}(x) \wedge \texttt{Own}(x)$ and the rule (1) above, then $\neg\exists x.\texttt{Car}(x) \wedge \texttt{Own}(x)$. The rule (1) receives a high weight in the given contexts since *car* is similar to *own* (the context of *vehicle*). Following Natural Logic, to handle inferences involving sentences containing nested propositions, they mark the polarity of the position of the embedded proposition.

The view of considering logical words, in particular negation, as pre-defined and close to their truth-value meaning is present in other work too (see Clark et al. (2008); Widdows (2003) among others), though in a full vector space model approach. Clark et al. (2008) take sentences to be in the space spanned by a single vector ($\vec{1}$), identified with "true" and the origin to be "false" ($\vec{0}$). So a sentence like "*John likes Mary*" is represented by the vector $\vec{1}$ if the sentence is "true" and by $\vec{0}$ otherwise, moreover the authors leave open the possibility of considering degree of sentence meaning instead of just the two truth values. Within the DSM framework of Clark et al. (2008), Preller & Sadrzadeh (2009) takes negation to be a base swapper operator. The details of this approach are further explained in Coecke et al. (2010). Similarly, Widdows (2003) starts from the intuition that unrelated meanings should be orthogonal to one another, which is to say that they should have no features in common at all. Hence, he takes negation to generate a vector representation that is completely orthogonal to the negated term. In the following, we will report on some preliminary results carried out on grammatical words, more in particular on determiners and determiner phrases, that suggest the possibility of undertaking a truly distributional analysis of these words too.

3.2.2 Grammatical words as tensors learned from their use

Menini (2012) studies the distributional behavior of 50 determiners (articles, quantifiers, and cardinals). First of all, the author aims to check how the distributional context changes if at the same nouns are applied different determiners and if similar determiners occur in similar contexts. To this end, he builds two DSMs using a large corpus:[8] (a) one with lemmatized content words as dimension (LDSM), and (b) a second one with inflected content and grammatical words as dimensions (IDSM). For each of the studied determiner, he extracts determiner phrases (DPs) from the corpus choosing the most frequent 20K nouns and their vector representation in the two DSMs mentioned above, and extract the closest neighbour of the DPs vectors. The experiment shows that in the DP vector representations of LDSM the meaning of the nouns emerges over the one of the determiner contrary to what happens in IDSM. Moreover, the use of a noun seems to change according to the determiner used: for instance,

[8] ukWaC, a 2 billion word corpus crawled from the web, British National Corpus, a 100 million word corpus, Wikipedia, about 1.6 billion tokens. The three corpora have been tokenzied and tagged with Treetagger.

whereas *every dog* tends to co-occur with general concepts, usually attributed to dogs in general – *animal*, *tail*, *pet*, *love*, *cleaver* and *friend* – "*that dog*" occurs with more familiar words, usually associated to a single dog, single episodes or everyday situation – *bite*, *owner*, *bad*, *police*, *kill* or *bloody*. Interestingly, the author conjectures that the determiner *that* is usually preferred for describing negative events, creating a distance between the dog and the speaker, whereas other determiners like *this* are used in positive contexts, occurring with words as *rescue*, *show*, *wonderful*, *loving* or *companion*. All in all, the experiment brings evidence to the claim that using a determiner rather than another affects the context in which the DP occurs. The result is confirmed by a second experiment carried out in Menini (2012) based on clustering. In this case, the vector representation of the determiner is computed out of the DPs by calculating their average vector representation. Interesting clusters have indeed emerged (e.g. {*too few*, *too many*, *too much*}, and {*four*, *three*, *two*, *several*} etc.), but further studies in this direction are necessary since no clustering method alone has succeed in the task. The experiment shows that also determiners seem to be characterized by their distributional context, but a DSM more suitable to their role should be built for them.

Finally, Menini (2012) reports on a third experiment in which pairs of DPs are studied. The attention is focused on nine determiners (*all*, *every*, *four*, *that*, *these*, *this*, *those*, *three* and *two*.) A classifier is used for recognizing similar vs. dissimilar DP pairs.[9] The author has carried out the task using different classifiers, the best results have been obtained with a polynomial super vector machine (SVM)[10] that has obtained the following weighted average of the precision and recall (F-measures): 0.8% (LDSM, with raw co-occurrence values) and 0.81 % (inflected IDSM, with lmi weight.) The same experiment has also be tried against unseen determiners, viz., the testing dataset contains one determiner more than the training dataset, but the same nouns already used for the training. The SVM was able to correctly classify up to 68.9% of the 1226 never seen instances.

Before concluding this section, we would like to draw the reader attention on some interesting studies on determiner phrases carried out within the psycholinguistic community. As it has been emphasized in Paterson et al. (2011) quantifiers have been studied in details from the formal semantics angle, but they have been mostly ignored by the empirical based community which has focused on content words. Interestingly, they have been studied in Pragmatics and Psycholinguistics. In Pragmatics, it has been claimed that quantifiers like *no*, *few*, *some* and *all* are scalar expressions: they can be ordered on a scale

[9]Similar are DPs that share the determiners or the noun (e.g., *four countries-four states*, or that have similar determiners and similar nouns (e.g., *two boats-four boats*) or have similar determiners and similar nouns (e.g., *this shirt-that coat*); whereas dissimilar DPs are such that they have different determiners and different nouns (e.g., *this village-every cat*), or different determiners and similar noun (e.g., *two musicians-those painters*) (or viceversa, e.g., *two artists-four ducks*)

[10]The other classifiers used are Naive, J48, SVM Radial Kernel. Interestingly, the need of a polynomial SVM classifier for classifying relations between DPs was shown in an other internal project too on DP entailment.

with respect to the strengths of the information that they convey. As it is well known, their use involves pragmatic inferences called scalar implicature (Grice, 1975) ("the participants in a conversation expect that each will tailor their contribution to be as informative as required but no more informative than is required"). Though, in formal semantics, for instance *some* has just one meaning, in practice it can be used in different ways, see for instance the example below taken from Paterson et al. (2011)

- R: if you ate some of the cookies, then I won't have enough for the party.
- M: I ate some of the cookies. In fact, I ate all of them. [Meaning: "some and possibly all"]
- R: Where are the apples that I bought?
- M. I ate some of them [Meaning: "some but not all")

Moreover, Paterson et al. (2011) shows that quantifiers can have different "polarity" even when denoting the same vague quantity: Quantifiers with positive (negative) polarity, e.g., *a few*, *quite a few*, *many*, (resp. *few*, *very few*, *not many*) are used to encourage (resp., discourage) the speaker to do something. The distinction between positive vs. negative polarity QPs is reinforced by their different behaviour with respect to the set of discourse entities they refer to and they make accessible via anaphora. To this end, the authors distinguish between the "reference set", viz. the set of entities the quantifier operates upon, and the "complement set", viz., the complement of the reference set.[11] Positive polarity QPs put the focus on the reference set while negative polarity QPs put the focus on the complement set. Moreover, the reference set is available for anaphora. Example:

"*(a) A few/(b)Few of the students attended the lecture. They ....*"

people continue (a) by speaking of properties of the reference set (e.g., "*They listen carefully and took notes*") and (b) by speaking of the complement set (e.g., "*They decided to stay at home instead*")

Question The psycholinguistics results on determiners reported above seem to confirm the possibility of studying these words (and maybe the class of logical words in general) from a distributional view. We wonder whether they also suggest that the relevant part of the context of use could be of a different nature than the one considered within DSMs for content words. For instance, instead of just looking at co-occurency frequency within a sentence, we might need to consider the discourse level (see the comment above on anaphora), or we might need to consider instead of the words in isolation, the semantic relation holding within the words in the observed context (see the comment on the choice of the verb phrase above.)

[11] Example: "*many of the students attended the lecture*". The reference set is the set of all the students who were present at the lecture, the complement set is the set of all the students who were absent.

3.3 Entailment in DSM

Clarke (2011) studies the algebraic properties a vector space used for representing natural language meaning needs to have and identifies possible directions to account for degree of entailment between distributional representations proposing to use the partial order of the defined algebraic structure. However, he does not describe the idea in details and does not evaluate it on any empirical ground. Implementations and interesting evaluation results have been carried out at lexical level. For instance, Erk (2009) suggests that it may not be possible to induce hyponymy information from a vector space representation, but it is possible to encode the relation in this space after it has been obtained through some other means. On the other hand, recent studies (Geffet & Dagan, 2005; Kotlerman et al., 2010; Weeds et al., 2004) have pursued the intuition that entailment is the ability of one term to "substitute" for another. For example, *baseball* contexts are also *sport* contexts but not *vice versa*, hence *baseball* is "narrower" than *sport* (*baseball* $\models$*sport*). On this view, entailment between vectors corresponds to inclusion of contexts or features, and can be captured by asymmetric measures of distribution similarity. In particular, Kotlerman et al. (2010) carefully crafted the *balAPinc* measure for lexical entailment. In brief, the balAPinc score is higher if many features are present and the included features are ranked high. Baroni et al. (2012) look at a similar issue but from a different perspective and by going beyond lexical level. The authors do not use a hand-crafted measure, but rather a machine learning based classifier. They use a SVM and show that it can learn the entailment relation between phrases of the same syntactic categories: from a training set of noun pairs it learns the entailment relation between expressions of this category ($\|=_N$) e.g., from training examples like *big dog* $\|=$ *dog*, it learns *dog* $\|=$ *animal*, and from a training set of quantifier phrases, e.g. *all dog* $\|=$ *some dog*, it learns the $\|=_{QP}$ even when the testing data set contains QPs never seen in the training data set. The entailment is specific to the syntactic category and does not generalize across the categories (if the SVM is trained on $\|=_N$ it will obtain bad performance on a $\|=_{QP}$ test dataset, and viceversa if trained on $\|=_{QP}$ it will obtain bad performance on a $\|=_N$ test dataset.)[12] The results reported in Baroni et al. (2012) had been obtained with a cubic polynomial kernel; interestingly, Gutierrez Vasques (2012) shows that a linear classifier will obtain worse results and that a two degree classifier (either homogeneous or inhomogeneous) would perform equally well than the cubic one. These latter results confirm the involvement of features interaction, rather than purely inclusion, in the entailment relation of DSM representations.

4 Conclusions

By reviewing the core aspects of formal and distributional semantics models and by presenting the more recent results obtained within DSMs beyond lexical

[12]This second half of the experiment, training on QPs and testing on Ns has been carried out by Gutierrez Vasques (2012).

level adopting the formal semantics binoculars, we have highlighted some open issues. First of all, we have underlined the importance of considering the correspondence between syntax and semantic both as expressed between syntactic categories and semantic types (types of semantic space) and as captured by the composition rules. As for the latter, we have raised the question of how structures with gaps can be handled with DSMs for which researchers so far have focused only on function application of juxtaposed function-arguments. Moreover, by introducing the concepts of satisfiability and validity, we have focused on the role the logical (grammatical) words play in natural language reasoning from a logic view and compared their role when observed from the language of use perspective. More research needs to be carried out on this class of words to understand whether there is the need of an hybrid system that combines logic and distributional relations or whether the integration of the two approaches would be needed only to take into account the two fregean aspects of meaning, reference and sense.

References

Barker, C., & Shan, C.-C. (2006). Types as graphs: Continuations in type logical grammar. *Journal of Logic, Language and Information*, *15*, 331-370.

Baroni, M., Bernardi, R., Shan, C.-C., & Do, N. Q. (2012). Entailment above the word level in distributional semantics. In *Proceedings of EACL 2012.*

Baroni, M., Bernardi, R., & Zamparelli, R. (In press). Frege in space: A program for compositional distributional semantics. *Linguistic Issues in Language Technology*. (Special issues on Semantics for textual inference)

Baroni, M., Evert, S., & Lenci, A. (Eds.). (2008). *Proceedings of the ESSLLI workshop on distributional lexical semantics.*

Baroni, M., & Zamparelli, R. (2010). Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of EMNLP* (pp. 1183–1193). Boston, MA.

Bernardi, R., & Moortgat, M. (2010, May). Continuation semantics for the Lambek-Grishin calculus. *Information and Computation*, *208*(5), 397–416.

Church, K. W., & Hanks., P. (1989). Word association norms, mutual information and lexicography. In *Proceedings of the 27th annual conference of the association of computational linguistics (ACL 1989).*

Clark, S. (2012). Quantum physics and linguistics: A compositional, diagrammatic discourse. In (chap. Type Driven Syntax and Semantics for Composing Meaning Vectors). Oxford University Press. (In press)

Clark, S., Coecke, B., & Sadrzadeh, M. (2008). A compositional distributional model of meaning. In P. Bruza, W. Lawless, K. van Rijsbergen, B. C. D. Sofge,

& S. Clark (Eds.), *Proceedings of the second symposium on quantum interaction* (p. 133-140).

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

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

Coecke, B., Grefenstette, E., & Sadrzadeh, M. (2012). *Lambek vs. Lambek: Vector space semantics and string diagrams for lambek calculus.* (Unpublished)

Coecke, B., Sadrzadeh, M., & Clark, S. (2010). Mathematical foundations for a compositional distributed model of meaning. *Lambek Festschirft, Linguistic Analysis, vol. 36*, *36*. Available from `http://arxiv.org/submit/10256/preview`

Curran, J., Clark, S., & Bos, J. (2007). Linguistically motivated large-scale nlp with c&c and boxer. In *Proceedings of acl (demo and poster sessions)* (pp. 33–36). Prague, Czech Republic.

Deerwster, S., Dumai, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *J. Am. Soc. Information Science and Technology*.

Erk, K. (2009). Supporting inferences in semantic space: representing words as regions. In *Proceedings of IWCS* (pp. 104–115). Tilburg, Netherlands.

Firth, J. R. (1957). A synopsis of linguistic theory 1930-1955. *Studies in Linguistic Analysis*. (Reprinted in F.R. Palmer (ed.), Selected Papers of J.R. Firth 1952-1959, London: Longman (1968).)

Garrette, D., Erk, K., & Mooney, R. (2011, January). Integrating logical representations with probabilistic information using markov logic. In *Proceedings of the international conference on computational semantics* (pp. 105–114). Oxford, England. Available from `http://www.cs.utexas.edu/users/ai-lab/pub-view.php?PubID=127042`

Geffet, M., & Dagan, I. (2005). The distributional inclusion hypotheses and lexical entailment. In *Proceedings of the 43rd annual meeting of the ACL* (pp. 107–114).

Grefenstette, E., Dinu, G., Zhang, Y.-Z., Sadrzadeh, M., & Baroni, M. (2013). Multi-step regression learning for compositional distributional semantics. In *Proceedings of IWCS.* Potsdam, Germany.

Grefenstette, E., & Sadrzadeh, M. (2011a). Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of EMNLP* (pp. 1394–1404). Edinburgh, UK.

Grefenstette, E., & Sadrzadeh, M. (2011b). Experimenting with transitive verbs in a DisCoCat. In *Proceedings of GEMS* (pp. 62–66). Edinburgh, UK.

Grice, H. P. (1975). Logic and conversation. *Syntax and semantics*. (Reprinted in Studies in the Way of Words, ed. H. P. Grice, pp. 2240. Cambridge, MA: Harvard University Press (1989))

Gutierrez Vasques, M. X. (2012). *Quantifying determiners from the distributional semantics view.* Unpublished master's thesis, Free University of Bozen-Bolzano. Erasmus Mundus European Master Program in Langauge and Communication Technologies.

Harris, Z. (1954). Distributional structure. *Word*, *10*(23), 146–162.

Honnibal, M., Curran, J. R., & Bos, J. (2007). Rebanking CCGBank for improved interpretation. In *Proceedings of the 48th annual meeting of acul* (pp. 207–215).

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, *28*(11).

Kartsaklis, D., Sadrzadeh, M., Pulman, S., & Coecke, B. (2012). *Reasoning about meaning in natural language with compact closed categories and frobenius algebras.* (Unpublished)

Kotlerman, L., Dagan, I., Szpektor, I., & Zhitomirsky-Geffet, M. (2010). Directional distributional similarity for lexical inference. *Natural Language Engineering*.

Lambek, J. (1958). The mathematics of sentence structure. *American Mathematical Monthly*, *65*, 154–170.

Lambek, J. (1999). Type grammars revisited. In F. L. A. Lecomte & G. Perrier (Eds.), *Logical aspects of computational linguistics* (pp. 1–27).

Lambek, J. (2004). A computational algebraic approach to english grammar. *Syntax*.

McDonald, S., & Ramscar, M. (2001). Testing the distributional hypothesis: The influence of context on judgements of semantic similarity. In *Proceedings of CogSci* (pp. 611–616).

Menini, S. (2012). *A distributional approach to determiners.* Unpublished master's thesis, University of Trento.

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

Montague, R. (1970). English as a formal language. *Linguaggi nella società e nella tecnica*, 189–224.

Moortgat, M. (1997). Categorial Type Logics. In J. van Benthem & A. ter Meulen (Eds.), *Handbook of logic and language* (pp. 93–178). Cambridge MA: The MIT Press.

Morrill, G. (1994). *Type logical grammar*. Dordrecht: Kluwer.

Paramita. (2012). *Exploiting ccg derivations within distributional semantic models*. Unpublished master's thesis, Free University of Bozen-Bolzano. Erasmus Mundus European Master Program in Langauge and Communication Technologies.

Paterson, K., Filik, R., & Moxey, L. (2011). Quantifiers and discourse processing. *Language and Linguistics Compass*, 1–29.

Pham, N., Bernardi, R., Zhang, Y. Z., & Baroni, M. (2013). Sentence paraphrase detection: When determiners and word order make the difference. In *Proceedings of the IWCS 2013 workshop: Towards a formal distributional semantics*.

Preller, A., & Sadrzadeh, M. (2009). Bell states as negation in natural languages. *ENTCS, QPL, University of Oxford*.

Socher, R., Huval, B., Manning, C., & Ng, A. (2012). Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP* (pp. 1201–1211). Jeju Island, Korea.

Steedman, M. (2000). *The Syntactic Process*. Cambridge, MA: MIT Press.

Turney, P., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, *37*, 141–188.

Vecchi, E. M., Baroni, M., & Zamparelli, R. (2011). (linear) maps of the impossible: Capturing semantic anomalies in distributional space. In *Proceedings of the ACL DISCo workshop* (pp. 1–9).

Weaver. (1955). Machine translation of languages. In W. Locke & D. Booth (Eds.), (chap. Translation). Cambridge, MA: MIT Press.

Weeds, J., Weir, D., & McCarthy, D. (2004). Characterising measures of lexical distributional similarity. In *Proceedings of the 20th international conference of computational linguistics, coling-2004* (pp. 1015–1021).

Widdows, D. (2003). Orthogonal negation in vector spaces for modelling word-meanings and document retrieval. In *Proceedings of the 41st annual meeting on association for computational linguistics - volume 1* (pp. 136–143). Stroudsburg, PA, USA: Association for Computational Linguistics. Available from `http://dx.doi.org/10.3115/1075096.1075114`

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