Integrating Type Theory and Distributional Semantics: A Case Study on Adjective–Noun Compositions

Nicholas Asher*
CNRS

Tim Van de Cruys*
CNRS

Antoine Bride*
Université Paul Sabatier

Márta Abrusán*
CNRS

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

In this article, we explore an integration of a formal semantic approach to lexical meaning and an approach based on distributional methods. First, we outline a formal semantic theory that aims to combine the virtues of both formal and distributional frameworks. We then proceed to develop an algebraic interpretation of that formal semantic theory and show how at least two kinds of distributional models make this interpretation concrete. Focusing on the case of adjective–noun composition, we compare several distributional models with respect to the semantic information that a formal semantic theory would need, and we show how to integrate the information provided by distributional models back into the formal semantic framework.

1. Introduction

Formal semantics (FS) has provided insightful models of composition and recently has addressed issues of how composition may in turn affect the original meanings of lexical items (Pustejovsky 1995; Partee 2010; Asher 2011). Type Composition Logic (TCL; Asher 2011) provides a detailed formal model of the interaction between composition and lexical meaning in which the composition of two words w and w' may shift the original meanings of w and w'. For example, consider the case of an adjective like *heavy* and a noun like *traffic*. TCL assigns a logical form to the adjective–noun combination *heavy traffic*, $\lambda x.(\mathcal{O}(\text{heavy})(x) \wedge \mathcal{M}(\text{traffic})(x))$, where $\mathcal{O}$ is a functor induced by the noun that outputs a meaning paraphrased as *heavy for traffic*. The $\mathcal{M}$ functor does something

* IRIT, Université Paul Sabatier, 118 Route de Narbonne, F-31062 Toulouse Cedex 9.
E-mail: {nicholas.asher, tim.vandecruys, antoine.bride, marta.abrusan}@irit.fr.

Submission received: 10 April 2015; revised version received: 26 July 2016; accepted for publication: 8 August 2016.

doi:10.1162/COLI_a_00264

© 2017 Association for Computational Linguistics

similar for the noun. Different types of adjectives will interact differently with the meaning of the noun; for example, non-subsective adjectives like *fake* in *fake dollar bill* output a meaning whose denotation has an empty intersection with the denotation of the original noun but shares surface properties with things that are (e.g., [real] dollar bills). TCL thus decomposes an adjective–noun combination into a conjunction of two properties representing the contextual contributions of the noun and adjective. This *decomposition property* allows TCL to predict non-trivial logical entailments just from the form of adjective–noun compositions (in contrast to Montague's higher order approach, which requires meaning postulates), while also capturing the shiftiness of lexical meaning, something that most formal semantic theories do not consider.

However, neither TCL nor the other FS theories mentioned provide a method for constructing such functors or lexical meanings. In this article we develop two distributional semantic (DS) models able to provide such a method, in virtue of (i) TCL's distinction between internal or conceptual content and external or referential content and (ii) the close correspondence between the way TCL and these models treat composition—in particular, the fact that these models share with TCL the decomposition property we just mentioned. We show that such methods can furnish the appropriate TCL functors, provided we take one big step: We identify TCL's internal content with vectors, which distributional methods use to represent word meaning. Functors introduced by TCL for composition then correspond to vector transformations within distributional models. We also show how to translate the results of these transformations back into TCL logical forms. TCL logical forms will then entail non-trivial inferences based on DS lexical information, while keeping the structural and conceptual advantages of a FS based logical form.

We illustrate our approach with adjective–noun compositions because they are simpler and better understood than other compositions. Such compositions do not typically introduce scope-bearing elements like quantifiers, unlike the construction of verb phrases, for instance. Also, the range of variation in adjective–noun composition is better understood, than, say, the effects of composition in verbal predications, which also involve more parameters that can potentially affect the composition.

2. Towards an Integration of DS and FS: The Formal Framework TCL

TCL Asher (2011) has three advantages compared with other FS theories for studying the interactions between FS and DS: its use of types and its notion of internal content, its commitment to the actuality of meaning shifts during composition, and its formal model of meaning shift. However, TCL does not supply detailed information about particular types, which is crucial to determining meaning shifts. This is where we turn to DS for help.

2.1 Types and TCL's Notion of Internal Content

In TCL, each word has a model-theoretic meaning that determines appropriate extensions for expressions (at points of evaluation). This is TCL's notion of external content, which is the usual notion of content in FS theories. In addition, however, each word in TCL has a type. Types are semantic objects and encode the "internal meaning" of the expression associated with it. So, for instance, the external semantics or extension of the word *wine* is a set of wine portions at some world and time, while the type or internal meaning of wine is given by the features we associate with wine (e.g., it is a liquid, a beverage, has alcohol, and has a particular taste). Internal semantics can also make use

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

of multi-modal information; thus olfactory and gustatory features can also play a role. These features enable speakers to correctly judge in normal circumstances whether an entity they experience falls under the extension of a term, though these judgments are not always completely reliable. The notions of internal and external meaning and the particular conception of how they interact are unique to TCL.

Types and internal content play an important role in TCL. They model selectional restrictions and are used to guide composition. An irresolvable clash between the type of a predicate and the type of its argument implies that the predication is semantically anomalous. Types also guide TCL's account of meaning shifts in predication. In TCL types encode the correct usage of a term. This is very similar to what DS methods do. However, TCL makes use of the lambda calculus for composition, a well-known and well-understood formalism; and doing so depends upon a particular interpretation of internal content based on a notion of justification. The correct usage and a speaker's mastery of the content of a term involves, among other things, an ability to justify, when asked, the use of that term in a particular context, and *mutatis mutandis*, an ability to justify the assertion of a predication in which a predicate applies to an individual characterized in a certain way. In the case of a speaker's assertion of a predication, such a justification explains why the speaker takes the assertion to be true. Such justifications encode the features that speakers use to identify extensions of terms. The reason these justifications are a part of linguistic mastery of expressions is that they are a reliable guide to determining extensions. Such justifications constitute internal content in TCL.

Modern type theories like TCL exploit a deep relation between proofs and types, known as the Curry-Howard correspondence (Howard 1980). The Curry-Howard correspondence shows that the notions of proof and types in the lambda calculus are isomorphic, allowing one to identify types with proofs or proof schemas. TCL exploits this correspondence with justifications and types for natural language expressions: Types and justifications are structurally isomorphic and so types are formally indistinguishable from justifications. In light of such a correspondence, the particular type assigned to a sentence like *this is wine* is identical to its justification, which is a defeasible proof of the truth of the proposition that the object the speaker demonstrates is wine. This identification of types with justifications not only theoretically clarifies the internal content of terms, but it also allows us to exploit the notion of composition in the typed lambda calculus as a method for composing internal contents without modification.

To clarify this core concept of TCL, we sketch a recursive definition of internal meanings $\|.\|$ for a language fragment with just nouns (N) and adjectives (A) and assuming, for illustrative purposes, a Montague-like composition rule for the two. We assume each individual, recognizable object has a name e; and to each such name we assign an individual type. We will take as primitive the set I of individual types. We identify these with an **individual justification rule** r that can be used to recognize the object denoted by e. To give an example of an individual justification rule, if a speaker A, who has used the demonstrative *this* to refer to a particular object in her environment, is asked to justify her use of the demonstrative (e.g., another speaker B asks, *what do you mean 'this'?*), the speaker will resort to an individual justification rule for determining the referent of *this*. Linguistically, such a rule is expressed as a definite description for the denotation—for example, to explain her use of *this*, speaker A might say: *the stuff in the glass that I'm holding*. Besides I, we will also take as basic the type PROP, the type of closed formulas or sentences. PROP is a set of **justifications**, which are, given the Curry-Howard correspondence, defeasible proofs for the truth of formulas. Note that

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

PROP also contains the empty set $\emptyset$ in case a formula has no justification. We specify the types and internal contents $\|.\|$ for nouns and adjectives as shown here:

- $\|N\| : \text{I} \rightarrow \text{PROP}$. That is, noun types are functions from individual justification rules $r \in \text{I}$ into a set of justifications that an individual satisfying r is of the type $\|N\|$ (or $\emptyset$, if there is no such justification).
- For adjectives A , $\|A\|: \|N\| \rightarrow \|N\|$. That is, an adjective meaning takes a noun meaning and returns another noun meaning.

To illustrate, a justification rule for *wine* must provide particular features such that if something satisfying a particular individual justification type r has these features, then we can defeasibly conclude it is wine. The justification rule for *wine* will appeal to olefactory, gustatory, and visual features (clear liquid of either yellow, red, or pink color) that are typical of wine. As an example of an adjectival meaning, $\|white\|$ is a function from a justification rule like that of the noun type $\|wine\|$ to a noun type justification rule for something being $\|white\ wine\|$. As the internal content of a noun N is a function from individuals to propositions, it is of the right type to be assigned as a meaning to the λ term λxNx, the usual representation for a common noun in formal semantics. The internal content of an adjective also has the requisite structure to reflect the standard type of adjectives, and this enables composition using the lambda calculus. This means we can compose internal contents using the same method with which we compose external contents.

TCL's characterization of internal content yields a natural link between internal content and external, model-theoretic content. The internal semantics "tracks" the external semantics, in that in the majority of cases or in normal circumstances, the internal semantics determines appropriate truth conditions for sentences. The internal content given by the types does not determine the expression's extension in all cases, as philosophical, externalist arguments show (Putnam 1975; Kripke 1980). But assuming speaker competence, internal content should normally yield the correct extensions for expressions. For instance, Nicholas's olfactory and gustatory capabilities are reasonably good at distinguishing different kinds of white wine. They are not infallible; and so they cannot determine the extension of the predicate *Chardonnay from the Corbières*. But they do often work correctly and would constitute his justification for his asserting that something is a Chardonnay from the Corbières. A justification for a predicative expression should in normal circumstances identify elements in that predicate's extension; otherwise it would not be a justification. Similarly, an individual justification rule r_t for using a referring term t would not be a justification if r_t did not pick out normally what t refers to. Composing these justifications and similar ones for other parts of speech together to get a justification for a whole sentence will then also normally deliver the correct truth value in a circumstance of evaluation. Because these justifications tell us what the truth conditions of the sentence would be *in the normal case*, they are in effect a *modal* characterization of those truth conditions.

2.2 TCL and Meaning Shifts

Meaning shifts occur often when composition occurs. We call meaning shifting compositions **co-compositions**, following Pustejovsky (1995). There are several kinds of co-composition. One kind is easily explained using TCL's system of types. An ambiguous word may be made less ambiguous when it combines with other words. Consider for

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

instance the word *traffic*. It is ambiguous at least between the senses of denoting a flow of vehicles or information. However, when combined with a modifier like *Internet* or *New York City* in the phrases *Internet traffic* and *New York City traffic*, this ambiguity vanishes: The modifier selects or at least prefers one of the senses of *traffic*. TCL and other type theoretic lexical theories represent the different senses of ambiguous words with the use of disjoint types. For example, *traffic* would have the disjoint type INFORMATION $\vee$ VEHICLE. TCL models the disambiguation in the phrases above with an inference that is logically sound: The predicate, by selecting one of the disjoint types to satisfy its selectional restrictions, makes the other types in the disjoint union non-applicable, thus conferring a more specialized meaning to the argument.

But the composition of a predicate and its argument can exhibit other sorts of meaning shifts as well, which pose challenges for type theories other than TCL. Consider the following adjective–noun compositions:

(1) a. heavy appliance
b. heavy rain
c. heavy sea
d. heavy bleeding
e. heavy smoker

In these examples the head noun affects the meaning of the modifier. If these data are well known, formal analyses for them are not. We could assume that adjectives are wildly ambiguous, roughly one sense for each noun with which they can combine. And we could model their internal content in terms of the disjoint union of their possible precisifications (the unambiguous senses). But that would miss or obscure certain logical relations. For instance, a heavy physical object does have something in common with *heavy rain*, and even with *heavy smoker* and *heavy bleeding*; in each case some dimension of the denotation of the head noun is modified towards an extreme, saturated end of the scale (Mel'cuk 2006). A disjoint union type is right for homonymously ambiguous expressions (such as *bank*) but not for logically polysemous ones—expressions whose senses have some logical or metaphysical connection.

To analyze logical polysemy, TCL appeals to functors that shift the meaning of the predicational relation itself. Although TCL motivates the functor view based on an analysis of coercion, it also uses it for co-composition. In TCL an expression has a type presupposition which must be satisfied in the predicational environment; a failure to satisfy such a type either leads to semantic anomaly or coercion effects. Type presuppositions are very general types like EVENTUALITY, PHYSICAL-OBJECT, or INFORMATIONAL-OBJECT. But an expression also has a more specific, "fine-grained" type that encapsulates the internal content specific to the term, the sort of content we discussed before. It is this fine-grained content that TCL exploits in co-composition.

TCL's approach to adjective–noun co-composition is quite different from a standard Montagovian approach. In standard semantic treatments, an adjectival meaning is a functor taking a noun meaning as an argument and returning a noun phrase meaning; composition is a matter of applying the adjective meaning as a higher-order property to the noun meaning. In TCL the noun and adjective meanings affect each other, and the output of an adjective–noun composition is the conjunction of a modified adjectival meaning and a modified noun meaning, which are both first order properties and apply to individuals, as in Schema (2). It introduces functors that potentially modify both the adjective and the noun's internal content in co-composition and then conjoins the modified contents. In the *adjective–noun composition* Schema (2), A is the adjective, N the

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

noun, $\mathcal{O}_A$ the functor on the noun given by the adjective, and $\mathcal{M}_N$ the functor on the adjective induced by the noun:

(2) $\lambda x\,(\mathcal{O}_A(N(x)) \wedge \mathcal{M}_N(A(x)))$

For subsective adjectives,[1] which include the vast majority of adjectives in most languages, $\mathcal{O}_A$ selects a subtype or constituent type of N, if they shift the meaning of N at all. Thus, the individuals satisfying $\mathcal{O}_A(N(x))$ will necessarily be a subset of the denotation of N at any point of evaluation. TCL thus predicts an influence of the adjective on the noun's denotation when we have a subsective modifier—in particular, an ambiguous noun may be disambiguated by the modifier's meaning. Those few adjectives that are not subsective, like *former* in *former prisoner*, support inferences that subsective adjectives do under the scope of some sort of modal or temporal operator; for example, *former prisoners* were once prisoners, *possible difficulties* are difficulties in some epistemic alternative, and *fake guns* and *stone lions* appear to be or look like guns and lions.

TCL also predicts an often non-trivial shift in the meaning of a modifier as it combines with various nouns or vice versa. This coincides with our findings in distributional semantics for adjective–noun compositions in Section 4. For instance, non-intersective adjectives are predicted to undergo a modification that relativizes their denotation. Consider a non-intersective adjective like *small* in the sentence *that's a small elephant*. The functor $\mathcal{M}_{\text{elephant}}$ should shift to select those things in the denotation of *elephant* that are small on a scale suitable for elephants. Adjective–noun compositions analyzed with functors thus immediately yield interesting inferences; *that's a small elephant* entails that that is an elephant and that it was small for an elephant.

According to TCL, adjective–noun compositions should thus be decomposable, in the sense that it should entail that there is an object of type N as modified by the adjective and that it has some properties given by the modified sense of the adjective. This formalizes observations made by other researchers as well (Kamp and Partee 1995; Partee 2010).

2.3 Types as Algebraic Objects

TCL tells us about the general form of composition, and the TCL equation in Example (2) imposes useful constraints on the functors essential to this process. But to build appropriate functors for individual words like *heavy* in the context of *storm*, for instance, TCL does not provide any method. DS offers us the promise of giving us the functors we want in a systematic and automatic way.

We will model each word's type or TCL internal content with a suitable algebraic object from DS. In most versions of DS, each basic word meaning is a vector in some space V whose dimensions are contextual features or a more abstract set of latent features. This is not quite the sort of defeasible justifications discussed in Section 2.1, but it is a place to start and it contains information pertinent to justification. Thus, individual word types will be modeled as vectors in a finite dimensional space V, whose dimensions reflect aspects of the context of use. The DS counterpart of a TCL functor is a transformation of $v \in V$ into a vector $v' \in V$, where v's values

1 A subsective adjective, A, in an adjective–noun combination AN is one that validates the inference from $AN(x)$ to $N(x)$. An intersective adjective, A, validates the inference from $AN(x)$ to $A(x) \wedge N(x)$. For more information, see Partee (1995).

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

on certain dimensions differ from those of v because the context has been filled in slightly. We want our functors to output a new type; so in the algebraic setting, a type is any vector in V. More general types—appropriate for type presuppositions and selectional restrictions—can be represented as functions of lower level types. Such an identification allows us to construct selectional restrictions for predicates automatically, which extends TCL's coverage dramatically.

In identifying types with vectors, we must take care that the type is captured in the right way so as to link with the "logical" type required for composition by FS. For composing adjectives and nouns, TCL's functor approach and the co-composition schema in Example (2) tells us that an adjective's contextually given type must depend on the fine-grained noun type it combines with and return a common noun type $\|N\|$, whereas the noun type must be a function from the fine-grained adjective types it combines with to common noun types — i.e., $(\|N\| \rightarrow \|N\|) \rightarrow \|N\|$.[2] As we saw in Section 2.1, in light of the Curry-Howard correspondence, it suffices to assign the right types to expressions, to have a compositional story with internal content. Once you specify the type of an expression, you have specified the form of its justification, its internal content; and that is all that is required to get composition to work. But once we have identified types with vectors in order to supply them with rich information, we need to revisit the issue, because vectors by themselves do not have the structure of TCL types or justifications. To exploit co-composition, the DS algebraic meaning for adjectives must reflect the *contextual modification* of that word's unmodified distribution due to a noun it combines with, and it must do something similar for nouns. In addition, a DS method must provide an algebraic meaning for nouns and adjectives that eventually provides a justification of the right type (e.g., a justification of type $\|N\|$).

We provide vector meanings for nouns and adjectives using DS methods by proceeding in several steps. First, we will provide a vector for the individual word, be it adjective or noun, within a space that takes the *syntactic/semantic dependencies* of that word into account. These include direct syntactic dependencies but more long distance semantic dependencies as well. In a second step, we exploit a space of latent dimensions to calculate compositional effects on these vectors. This second step adapts these vectors to the local predicational context. The noun vector is weighted by the dimensions that are most prominent in the adjective's latent representation, and the adjective's vector is similarly adjusted to take into account the meaning of the noun with which it is paired. Building the modified meanings in this way will enable us to output a meaning of the right type for the co-composition. The process, which we detail in Section 3.1, outputs two predicates of type $\|N\|$ that we can conjoin together to get the meaning of the adjective–noun combination.

2.4 Discussion

Some might wonder why we consider it necessary to mix statistical and logical information in one system. Would it not be possible to just use the statistical information provided by vectors, without recourse to types? We think a system like TCL has some attractive FS features—like the use of variables or discourse referents, scope bearing operators, and so forth—that will be difficult to reproduce with algebraic methods on their own (Garrette, Erk, and Mooney 2011). Further, convinced by the arguments

2 In order to express this co-dependence formally, we must assign a higher functional type to nouns than the one given to nouns in Montague grammar. See Asher (2011) for details.

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

in Kripke (1980) and Putnam (1975), we believe that algebraic methods, and indeed any purely internal semantics, cannot capture the external aspects of meaning. Both of these are crucial to determining truth conditions, a long-standing goal of theories of formal semantics (cf., e.g., Frege 1985; Montague 1974; Davidson 1967b; and Lewis 1970). We believe that a proper theory of meaning needs an external semantics as well as an internal one. The external part of the semantics is tied to a theory of truth, and the internal one to the content that expressions come endowed with by virtue of how they are used and how that use is justified. And as internal content characterizes truth conditions modally, our DS construction of TCL functors should ultimately affect our characterization of truth conditions. But to do that, we have to bring the information encoded in the modified vectors back into the symbolic system, via expressions that DS associates with a target expression. Ideally, the characterization of the output of the functors applied to an adjective or noun should be something like a defeasible justification for using the particular adjective or noun in that particular predicational context. Although statistical distributions that DS methods offer do not offer this directly, we investigate in the following sections how cosine similarity might capture information relevant to the functor definition and to its output, although other approaches might offer improved results (Roller, Erk, and Boleda 2014).

The identification of vectors and types changes TCL and its approach to basic linguistic phenomena. Whereas in TCL, semantic well-formedness was originally a binary decision, semantic well-formedness now becomes a matter of degree. We could provide a score to each predication depending on how close the fine-grained types are to matching type presuppositions. The closer the distances, the better or more prototypical the predication. Thus TCL's binary view of semantic well-formedness would morph into a more graduated scale, which might more accurately reflect the intuitions of ordinary speakers (Magidor 2013). A further change to TCL is the nature of the space of types. Although type spaces in most type theories are discrete, the space of types given our new assumptions is a compact metric space. This allows us to apply more constraints to meaning shift that can give the account some more meat. For instance, the TCL functors are constrained by the types they modify. One cannot just shift a type anywhere in type space. If types are points in a metric space, we can make this restriction precise by, for example, using a Lipschitz condition.[3] Such a constraint requires that functors should treat similar types similarly.

3. Distributional Models for Constructing Internal Contents and Their Composition

To incorporate information from distributional semantics into TCL's functor approach, our distributional models need to provide modified vectors, in our case study, both for adjectives (as modified by nouns) and nouns (as modified by adjectives). This section provides an overview of two distributional models that are able to provide us with such vectors. Section 4 contains the results of our case study, where we apply the models to the case of adjective–noun composition. In Section 5, we then sketch how the information that comes from distributional models might be incorporated into a TCL logical form.

We consider two different methods for computing a contextual weighting of adjective and noun vectors. The first method, *latent vector weighting*, is based on a matrix

3 A function f obeys the **Lipschitz condition** iff $\forall x, y \in \mathbb{R}^n,\ \|f(x) - f(y)\| \leq C\|x - y\|$, where $\|.\|$ is a suitable norm for the vector space and C is some constant.

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

factorization technique in which a latent space is constructed that is shared between different modes. The second technique, based on tensor factorization, makes use of a latent "core tensor" that is able to model multi-way interactions between the latent factors of different modes. Neither method is associative or commutative, and so are a priori plausible candidates for general composition methods in DS.

Note that these are not the only models one might use in order to compute the modified vectors we are after; we chose to illustrate our approach with these two models because they provide a straightforward way to compute the contextified vectors that we need for integration with TCL's functor approach. However, some models are not suitable. Additive or multiplicative methods for combining meanings (Mitchell and Lapata 2010) do not yield unique decomposition. For instance, an additive method produces a vector that could be the result of any number of sums of vectors.

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

3.1 Latent Vector Weighting

The main idea of **latent vector weighting** (LVW) is that the adjective (noun) that appears with a particular noun (adjective) defines a distribution over latent semantic factors, which is subsequently used to adapt the general vector representation of the noun (adjective), shifting the vector towards the correct meaning. As a first step, a factorization model is constructed in which words, together with their window-based context words and their dependency relations, are linked to latent dimensions. The factorization model then allows us to determine which dimensions are important for a particular expression, and adapt the dependency-based feature vector of the word accordingly. The model uses non-negative matrix factorization (Lee and Seung 2000) in order to find latent dimensions. We use **non-negative matrix factorization** (NMF), because of the property that its dimensions each give a more interpretable component of meaning (Lee and Seung 1999), and because it has an efficient learning algorithm (Lee and Seung 2000). A detailed description of the method can be found in Van de Cruys, Poibeau, and Korhonen (2011).

Using the results of the factorization model, we can adapt a word's feature vector according to the compositional expression it appears in.[4] Intuitively, a modifier that takes part in a compositional expression with the target word (e.g., an adjective modifier that appears with a target noun) pinpoints the important semantic dimensions of the target word, creating a probability distribution over latent factors $p(\mathbf{z}|d_i)$, where d_i is the dependency feature that represents the target word's modifier in the compositional expression.

The resulting probability distribution over latent factors can be interpreted as a semantic fingerprint according to which the target word needs to be interpreted. By combining this fingerprint with the appropriate factor matrix, we can now determine a new probability distribution over dependency features given the context—$p(\mathbf{d}|C)$.

$$p(\mathbf{d}|C) = p(\mathbf{z}|C)p(\mathbf{d}|\mathbf{z}) \tag{3}$$

The last step then is to weight the original probability vector of the word according to the probability vector of the dependency features given the word's

4 Note that the factorization that comes out of the NMF model can be interpreted probabilistically (Gaussier and Goutte 2005; Ding, Li, and Peng 2008). More details are provided in Van de Cruys, Poibeau, and Korhonen (2011).

context, by taking the pointwise multiplication of probability vectors $p(\mathbf{d}|w_i)$ and $p(\mathbf{d}|C)$.

$$p(\mathbf{d}|w_i, C) \sim p(\mathbf{d}|w_i) \cdot p(\mathbf{d}|C) \tag{4}$$

Note that this final step is a crucial one in our approach. We do not just build a model based on latent factors, but we use the latent factors to determine which of the features in the original word vector are the salient ones given a particular context. This last step provides the algebraic counterpart of TCL's functors. In the LVW model, what we do is use two vector spaces, the original vector space V where each word is represented in a space of syntactic/semantic contexts and a vector space V' with reduced dimensions, where lexical meanings have a more topical representation. Computing the conditional probability of each dimension z of V' relative to the vector for the adjective then provides a way of calculating the probability of each element of V given the presence of the adjective. This "slightly more determined context" vector v^* now furnishes the algebraic counterpart of our functor: The functor can be represented as $\lambda v\ v^*.v$, where v^* is the contextually weighted vector $p(\mathbf{d}|C)$, v is the original vector whose values are $p(\mathbf{d}|w_i)$, and $v^*.v$ signifies the point-wise product of the two vectors.

The following example, which uses actual corpus data, illustrates how the approach works. Say we want to compute the distributionally similar words to the noun *device* in the context of example expressions *explosive device* and *electrical device*. First, we determine our semantic fingerprints—$p(\mathbf{z}|explosive)$ and $p(\mathbf{z}|electrical)$, which are provided by our factorization model. Using these probability distributions over latent factors, we can now determine the probability of each dependency feature given the different contexts—$p(\mathbf{d}|explosive)$ and $p(\mathbf{d}|electrical)$—following Equation (3). Our last step is then to weight the original probability vector of the target word (the aggregate of dependency-based context features over all contexts of the target word) according to the new distribution given the argument that the target word appears with, using Equation (4). We can now compute the top similar words for the two adapted vectors of *device* given the different arguments, which, for the first expression, yields {*device*, *ammunition*, *firearm*, *weapon*, *missile*} and for the second expression yields {*device*, *equipment*, *sensor*, *system*, *technology*}.[5]

3.2 Tensor Factorization

Our second approach—based on tensor factorization (TENSOR)—allows for an even richer and more flexible modeling of the interaction between adjectives and nouns, in order to provide an adequate representation of each when they appear in each other's context. The key idea is to factorize a three-way tensor that contains the multi-way co-occurrences of nouns, adjectives, and other dependency relations (in a direct dependency relationship to the noun) that appear together at the same time. A number of well-known tensor factorization algorithms exist; we opt for an algorithm called Tucker factorization, which allows for a richer modeling of multi-way interactions using a core tensor. In Tucker factorization, a tensor is decomposed into a core tensor,

5 We constructed a separate model for adjectives because the dependency relations for adjectives are rather different. This allows us to compute the most similar adjectives to a particular adjective used in context (weighting the original adjective vector consisting of dependency features). Formally, we take these similar adjectives to be simple predicates and so effectively of type $\|N\|$, as required from Section 2.3.

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

multiplied by a matrix along each mode. For a three-mode tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times L}$, the model is defined as

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} \tag{5}$$

$$= \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} g_{pqr} \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r \tag{6}$$

where $\circ$ represents the outer product of vectors. By setting $P, Q, R \ll I, J, L$, the factorization represents a compressed, latent version of the original tensor $\mathcal{X}$; matrices $\mathbf{A} \in \mathbb{R}^{I \times P}$, $\mathbf{B} \in \mathbb{R}^{J \times Q}$, and $\mathbf{C} \in \mathbb{R}^{L \times R}$ represent the latent factors for each mode, and $\mathcal{G} \in \mathbb{R}^{P \times Q \times R}$ indicates the level of interaction between the different latent factors.

Again, we carry out the factorization with non-negative constraints, for the same reasons that were mentioned with our first approach, and we find the best possible fit to the original tensor $\mathcal{X}$ using Kullback-Leibler divergence, a standard information-theoretic measure. We make use of an efficient algorithm for non-negative Tucker decomposition, exploiting the fact that our input tensor is ultra-sparse. More details on the algorithm may be found in Chi and Zhu (2013).

To ensure that the algorithm finds a good global optimum, we initialize the three matrices using data that come from the non-negative matrix factorization of our first approach. Additionally, to strike a balance between the rich latent semantics that comes from the non-negative matrix factorization and the latent multi-way interaction that is provided by our tensor factorization algorithm, we do not make the tensor factorization algorithm converge, but we stop the iterative updates early based on the reconstruction of adjective–noun pairs from a development set (cfr. infra).

We can now compute a representation for a particular adjective–noun composition. In order to do so, we first extract the vectors for the noun ($\mathbf{a}^i$) and adjective ($\mathbf{b}^j$) from the corresponding matrices **A** and **B**. We can now multiply those vectors into the core tensor, in order to obtain a vector **h** representing the importance of latent dimensions given the composition of noun i and adjective j, that is, $\mathbf{h} = \mathcal{G} \times_1 \mathbf{a}^i \times_2 \mathbf{b}^j$. By multiplying the vector representing the latent dimension with the matrix for the mode with dependency relations (**C**), we are able to compute a vector **d** representing the importance of each dependency feature given the adjective–noun composition, namely, $\mathbf{d} = \mathbf{h}\mathbf{C}^T$. The last step is then again to weight the original noun vector according to the importance of each dependency feature given the adjective–noun composition, by taking the pointwise multiplication of vector **d** and the original noun vector **v** (i.e., $\hat{\mathbf{v}}_d = \mathbf{d}_d.\mathbf{v}_d$). Note that we could just keep the representation of our adjective–noun composition in latent space. In practice, the original dependency-based representation provides a much richer semantics, which is why we have chosen to perform an extra step weighting the original vector, as we did with our first approach, latent vector weighting.

As an example, when the computations outlined here are applied to the expressions *muddy bank* and *financial bank*, the top similar words are {*hillside*, *slope*, *ledge*, *cliff*, *ridge*} and {*bank*, *broker*, *insurer*, *firm*, *banker*}, respectively.

3.3 Implementational Details

This section contains a number of implementational details for both our approaches. We used the UKWaC corpus (Baroni et al. 2009), an Internet corpus of about 1.5 billion

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

words, to construct the algebraic structures for both approaches. We tagged the corpus with part-of-speech tags, lemmatized it with the Stanford Part-Of-Speech Tagger (Toutanova and Manning 2000; Toutanova et al. 2003), and parsed it using MaltParser (Nivre, Hall, and Nilsson 2006).

For the LVW approach, the matrices needed for our NMF factorization were extracted from the corpus. We built the model, using 5K nouns (or 2K adjectives), 80K dependency relations, and 2K context words[6] (excluding stop words) with highest frequency in the training set. All matrices were weighted using pointwise mutual information (Church and Hanks 1990). The NMF model was carried out using $K = 600$ (the number of factorized dimensions in the model), and applying 50 iterations.

For our second approach, the tensor factorization approach, we extracted our input tensor $\mathcal{X}$ of 5K nouns by 2K adjectives by 80K dependency relations from the corpus. The tensor $\mathcal{X}$ was weighted using a three-way extension of pointwise mutual information (Van de Cruys 2011). We set $K = 300$ as our number of latent factors. The value was chosen as a trade-off between a model that is both rich enough, and does not require an excessive amount of memory (for the modeling of the core tensor). The three matrices of our factorization model were initialized using the latent matrices for nouns, adjectives, and dependency relations from our LVW approach, using 300 dimensions. For the adjective matrix, the appropriate adjectives were extracted from the dependency matrix.

In order not to overfit our tensor factorization (i.e., to strike a balance between the semantics coming from the NMF factorization and the interaction information provided by our three-way tensor), we stopped the factorization algorithm early. We created a development set of 200 adjective–noun combinations, and we monitored the cosine similarity between the adjective–noun vector constructed by our model, and the adjective–noun vector that was attested in the corpus. We stopped iterating the factorization when the mean reciprocal rank of the attested combination (computed over a full set of about 100K adjective–noun combinations) was the highest.

All similarity computations for both approaches were performed using cosine as a similarity measure.

4. A Case Study on Adjective–Noun Compositions

4.1 Methodology

In this section, we provide results for a pilot study as to whether the two distributional approaches described earlier reflect a semantic shift in co-composition for adjectives and nouns and can offer something like a justification, in terms of related words, of an expression's use in context. Our evaluation used a list of English adjective–noun combinations drawn from *Wiktionary*, extracted by the method discussed in Bride, Van de Cruys, and Asher (2015). We added to this list adjective–noun combinations that we thought would exhibit more interesting co-compositional interaction, to achieve a list of 246 adjective–noun pairs in total (see Appendix A).

We created vectors for each of the adjective–noun combinations—using both the LVW and TENSOR approach—and computed the top 10 most similar nouns and top 10 most similar adjectives for each of the vectors using cosine similarity. For comparison, we also computed the results for the original, non-composed noun vector

6 We used a fairly large, paragraph-like window of four sentences.

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

(UNMODIFIED), as well as for composed adjective–noun vectors created using the lexical function (LEXFUNC) model of Baroni and Zamparelli (2010).[7]

Two of the authors, both experts in formal semantics, evaluated the resulting sets, guided by the following criteria:

1. **Meaning shift** — Do the distributional approaches predict a meaning shift in the composition of an adjective with a noun?
2. **Subsectivity and intersectivity** — Given an adjective A and noun N and their composition AN, do the methods predict:
   (a) **Subsectivity** — Does the composed adjective–noun meaning predict the individual noun meaning, i.e., $AN(x) \rightarrow N(x)$? For example, *former* is not subsective, but *small* is.
   (b) **Intersectivity** — Does the composed adjective–noun meaning predict the individual adjective meaning, i.e., $AN(x) \rightarrow A(x)$? For example, *small* is not intersective, but *round* is.

   We evaluated subsectivity and intersectivity as follows: if the original noun (adjective) was among the ten most similar nouns (adjectives) to the modified expression, we concluded that subsectivity (intersectivity) holds. Though somewhat limited, it is a straightforward way to examine the tendencies with regard to this criterion.
3. **Entailment** — Evaluators examined whether each of the 10 most similar words Y to the modified adjective or noun was such that $AN(x)$ defeasibly entails $Y(x)$. Our guideline was that X was a defeasible entailment of Y iff an X was normally or usually a Y or the presence of X normally or usually implied the presence of Y. For instance, is heavy bleeding (normally) uncontrolled?
4. **Semantic coherence** — Evaluators examined whether each of the 10 most similar words Y to the modified adjective or noun semantically related to the expression in ways that could help constitute a justification of its use. Semantic relations included: part–whole (e.g., is *wind* a part of a *heavy storm*?), subtype (e.g., is *hurricane* a subtype of *heavy storm*?), typical localization (e.g., does *heavy traffic* typically occur at peak periods?), causal (e.g., can *heavy bleeding* be eventually fatal or cause a fatal condition?), semantic alternative (e.g., is diarrhea a semantic alternative to *heavy bleeding*?), and antonym relations (e.g., is *light* an antonym of *heavy* in *heavy traffic*?) (Girju et al. 2009).

The first question—meaning shift—was evaluated quantitatively (taking cosine as a proxy for shift) and qualitatively (by manually inspecting the overlap of the lists of closest words for the unmodified and modified word meanings), and the others were treated as binary classification problems, evaluated in terms of accuracy. We investigated these phenomena for both nouns (weighting the noun vector with regard to the adjective context) and adjectives (weighting the adjective vector with regard to the noun

7 We constructed the models over the same corpus data, making use of the DISSECT toolkit (Dinu, Pham, and Baroni 2013).

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

context).[8] The annotators evaluated criteria 3 and 4 both with regard to the most similar and the top 10 most similar words. Twenty percent of the data were doubly annotated yielding Cohen κ scores of between 0.67 and 0.74 for the various criteria.[9] The rest of it was singly annotated, with the second annotator then reviewing and discussing the decisions of the first when there was disagreement; the annotators then produced the final data sets by consensus. Using their judgments, we then compiled accuracy figures as to whether, according to a given method, the most similar word to the target expression was a defeasible entailment or semantically related and how many of the 10 most similar words stood in one of these relations.

4.2 Results

Meaning shifts were observed for almost all adjective–noun combinations. As an illustration, consider the shift of the adjective *heavy* when it modifies the noun *traffic* (computed using the LVW method). The first listing gives the 10 most similar adjectives to the unmodified vector for *heavy*. The second listing shows the 10 most similar adjectives to the vector for *heavy* in the context of the noun *traffic*.

1. **heavy**$_A$: *heavy*$_A$ (1.000), *torrential*$_A$ (.149), *light*$_A$ (.140), *thick*$_A$ (.127), *massive*$_A$ (.118), *excessive*$_A$ (.115), *soft*$_A$ (.107), *large*$_A$ (.107), *huge*$_A$ (.104), *big*$_A$ (.103)
2. **heavy**$_A$, traffic$_N$: *heavy*$_A$ (.293), *motorized*$_A$ (.231), *vehicular*$_A$ (.229), *peak*$_A$ (.181), *one-way*$_A$ (.181), *horse-drawn*$_A$ (.175), *fast-moving*$_A$ (.164), *articulated*$_A$ (.158), *calming*$_A$ (.156), *horrendous*$_A$ (.146)

There is an evident shift in the composed meaning of *heavy* relative to its original meaning; there is no overlap in the lists 1 and 2 except for *heavy*. Using LVW, $sim_{cos}(\vec{v}_{orig}, \vec{v}_{mod})$ for all the adjectives varied between .25 and .77, with the vast majority of adjectives exhibiting $sim_{cos}(\vec{v}_{orig}, \vec{v}_{mod})$ lower than .50. The mean overlap between the shifted set of similar words and the original set of 20 most similar words is 6, whereas for nouns it is 10. $sim_{cos}(\vec{v}_{orig}, \vec{v}_{mod})$ for all the nouns varied between .3 and .8, and the mean was .5. Using tensor factorization, the quantitative shift was even larger; for nouns, $sim_{cos}(\vec{v}_{orig}, \vec{v}_{mod}) \leq .3$ on average and was never above .5. The raw similarity scores on the LEXFUNC approach were significantly higher on average than those for the other approaches. These quantitative measures show that a shift in co-composition was the norm for adjectives in our test set, both for the LVW and the TENSOR method. Nouns shifted less, something that we expected from TCL and the principle of subsectivity.

The results for **subsectivity and intersectivity** are presented in Table 1. As the results indicate, subsectivity clearly holds for LVW and TENSOR, whereas this is less the case for the LEXFUNC model. The results for intersectivity are mixed: Although the LVW method clearly favors intersectivity, the results of the TENSOR method are quite a bit lower.

Accuracy results for **entailment** are reported in Table 2. Using LVW for nouns, 59% yielded entailments for the most similar noun, and 32% for the top ten most similar nouns. Adjectives score quite a bit lower: 32% were judged to be good defeasible entailments for the most similar adjective, and 25% for the top 10 most similar adjectives.

8 Baroni and Zamparelli's LEXFUNC method does not provide results for adjectives, hence they are not included. Our full results are available upon request.

9 For semantic coherence, κ was calculated on the union of semantic relations.

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

Table 1
Accuracy results for subsectivity and intersectivity.

method	subsectivity	intersectivity
UNMODIFIED	**1.00**	.95
LEXFUNC	.58	–
LVW	.99	**1.00**
TENSOR	.97	.47

Table 2
Accuracy results for entailment. $\mathbf{ent}_1$ looks at the top most similar word, and $\mathbf{ent}_{10}$ looks at the top 10 most similar words.

method	nouns		adjectives	
	$\mathbf{ent}_1$	$\mathbf{ent}_{10}$	$\mathbf{ent}_1$	$\mathbf{ent}_{10}$
UNMODIFIED	.22	.13	.18	.12
LEXFUNC	.42	.23	–	–
LVW	**.59**	**.32**	**.32**	**.25**
TENSOR	.42	.30	.16	.13

The results for LEXFUNC and TENSOR are even lower. We might conclude that cosine similarity between vectors is, as we suspected, not a particularly good way to capture entailments, and it might be better to use other methods (Roller, Erk, and Boleda 2014; Kruszewski, Paperno, and Baroni 2015). Although the presence of similar antonyms contributed to the putative entailments that were judged bad, there were also many cases where the composition method yielded related words but not entailments. We evaluated the methods with respect to these semantically related words in Table 3.

The accuracy results for **semantic coherence** are reported in Table 3. For nouns, our TENSOR method did significantly better than either the LEXFUNC or the LVW methods. 52% of the most similar nouns were semantically related in some relevant way other than entailment; and among the top 10 closest meanings to the original noun, 43% bore one of our semantic relations to the targeted, shifted noun meaning. The noun meaning closest to the target noun meaning modified by co-composition stood either in an entailment relation or other semantic relation 94% of the time; and 73% of the top 10 closest nouns were either entailments or stood in one of the other semantic relations we tested for, an improvement of over 20% compared with the second best method,

Table 3
Accuracy results for semantic relations (**sr**) and entailment. Results are presented for both the top most similar word, and the top 10 most similar words. **ent+sr** combines both entailments and semantically related words.

method	nouns				adjectives			
	$\mathbf{sr}_1$	$\mathbf{sr}_{10}$	$\mathbf{ent+sr}_1$	$\mathbf{ent+sr}_{10}$	$\mathbf{sr}_1$	$\mathbf{sr}_{10}$	$\mathbf{ent+sr}_1$	$\mathbf{ent+sr}_{10}$
UNMODIFIED	.37	.20	.59	.33	.14	.09	.32	.21
LEXFUNC	.19	.15	.61	.38	–	–	–	–
LVW	.27	.20	.86	.52	.35	.24	**.67**	**.49**
TENSOR	**.52**	**.43**	**.94**	**.73**	**.38**	**.30**	.53	.43

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

LVW. The tensor factorization method reproduced the finding in the literature that co-hyponyms are often closely related to the target vector (Weeds, Weir, and McCarthy 2004). On adjectives, the TENSOR method performed significantly better on other semantic relations than with entailments, but still well below its performance with the shifted noun meanings.

5. Integrating Semantic Information from DS Methods into TCL Functors

Following our plan laid out in Section 2.3 for integrating DS content into TCL, we now provide a preliminary and admittedly somewhat speculative account of how to translate our results of Section 4 into a "spell-out" of the functors $\mathcal{O}_A$ and $\mathcal{M}_N$. These operators approximate symbolically the shift in meaning under co-composition.

The spell-out uses the list of similar words for the modified adjective and noun. Because we are translating into a purely symbolic environment, we need a separate process that clusters the predicates into different coherent internal meanings. The operator *Normally* is a modal operator that tells us what is normally the case in context. Let $N_1, \ldots, N_k$ and $A_1, \ldots, A_l$ be the closest nouns and adjectives related to the modified noun/adjective. The relations SR_i stand for (a subset of) the semantic relations discussed in criterion 4 of Section 4.1.

(7) $\mathcal{O}_A(N)(x) := N(x) \wedge \text{normally}(N_1(x) \wedge \ldots \wedge N_j(x) \wedge SR_1(N_{j+1}, N)(x)) \wedge \ldots \wedge SR_m(N_k, N)(x))$

For adjectives, we have:

(8) $\mathcal{T}_N(A)(x) := \text{normally}(A_1(x) \wedge \ldots \wedge A_j(x) \wedge SR_1(A_{j+1}, A)(x) \wedge \ldots \wedge SR_n(A_l, A)(x))$

Consider the example *heavy bleeding*. Looking at the the top five closest noun meanings for the adjective–noun composition using the TENSOR method, we obtain the following functor modifying the noun.

(9) $\mathcal{O}_{\text{HEAVY}}(\text{bleeding})(x) := \text{bleeding}(x) \wedge \text{normally}(\text{complication}(x) \wedge \text{irritation}(x) \wedge \text{alternative}(\text{diarrhea}, \text{bleeding})(x), \wedge \text{result}(\text{bleeding}, \text{discomfort})(x))$

To automatically specify our noun functors we would need good detectors for semantic relations between nouns or between NPs and between adjectives. This, as far as we know, is not yet feasible, but there are signs that we are not so far away.[10]

The functor for the adjectival meaning is based on the closest adjectives to *heavy* in its predicational context *heavy bleeding*, calculated with the TENSOR method:

(10) $\mathcal{O}_{\text{BLEEDING}}(\text{heavy})(x) := \text{normally}(\text{sudden}(x) \wedge \text{prolonged}(x) \wedge \text{uncontrolled}(x) \wedge \text{avoidable}(x)) \wedge \text{possible-result}(\text{heavy}, \text{fatal})(x)$

The TCL meaning for *heavy bleeding* is derived from conjoining the output of the two functors in Equations (9) and (10) and then lambda-abstracting over x. This yields a term that accords with the co-composition schema in Equation (2). We have thus taken some first steps to provide appropriate justification rules and internal contents for terms and a much more robust approach to lexical semantics combining both TCL and DS. We have opted for the discretized output given here to ensure that our composition has a

10 For causal relations, see the work of Do, Chan, and Roth (2011).

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

model-theoretic interpretation as well as a type-theoretic one, and yields semantically appropriate inferences. An alternative might have been to take the similarity values assigned to the words in our list to correspond to probabilities that such predicates hold in this context. But we see little theoretical or empirical grounds for taking similarity values to be probability values that the predicates hold; and doing so makes no semantic sense in many cases using our method.[11]

6. Related Work

In recent years, a number of methods have been developed that try to capture compositional phenomena within a distributional framework. One of the first approaches to tackle compositional phenomena in a systematic way is Mitchell and Lapata (2010). They explore a number of different models for vector composition, of which vector addition (the sum of each feature) and vector multiplication (the element-wise multiplication of each feature) are the most important. They evaluate their models on a noun–verb phrase similarity task, and find that the multiplicative model yields the best results, along with a weighted combination of the additive and multiplicative model. We have argued here that simple additive and multiplicative models will not do the job we want because they fail on decompositionality.

Baroni and Zamparelli (2010) present the lexical function model (LEXFUNC) for the composition of adjectives and nouns. In their model, an adjective is a linear function of one vector (the noun vector) to another vector (the vector for the adjective–noun combination). The linear transformation for a particular adjective is represented by a matrix, and is learned automatically from a corpus, using partial least-squares regression. We have evaluated their method to the extent we were able on our test set, and saw that it yielded results that did not suit our purposes as well as other methods.

Dinu and Lapata (2010) advocate a method that resembles LVW, in that it uses a distribution over latent dimensions in order to measure semantic shifts in context. However, whereas their approach computes the contextualized meaning directly within the latent space, the LVW approach we adopt in this article exploits the latent space to determine the features that are important for a particular context, and adapt the original (out-of-context) dependency-based feature vector of the target word accordingly. This allows for a more precise and more distinct computation of word meaning in context. Secondly, Dinu and Lapata use window-based context features to build their latent model, whereas our approach combines both window-based and dependency-based features.

There exists a large body of work on lexical substitution that aims to compute the meaning of words in context. Erk and Padó (2008, 2009) make use of selectional preferences to express the meaning of a word in context; to compute the meaning of a word in the presence of an argument, they multiply the word's vector with a vector that captures the inverse selectional preferences of the argument. Thater, Fürstenau, and Pinkal (2010) extend the approach based on selectional preferences by incorporating second-order co-occurrences in their model; their model allows first-order co-occurrences to act as a filter upon the second-order vector space, which computes meaning in context. And Erk and Padó (2010) propose an exemplar-based approach, in which the meaning of a word in context is represented by the activated exemplars that are most similar to it.

11 For a different take on this issue, see, e.g., Beltagy, Erk, and Mooney (2014).

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

Coecke, Sadrzadeh, and Clark (2011) introduce an abstract categorial framework that unifies both certain theories of syntactic structure and certain general approaches to DS. A number of instantiations of the framework are tested experimentally in Grefenstette and Sadrzadeh (2011a, 2011b). The key idea is that relational words (e.g., adjectives or verbs) have a rich (multi-dimensional) structure that acts as a filter on their arguments. Like Mitchell and Lapata (2010) and Baroni and Zamparelli (2010), they explore the idea that lexical semantics and composition can be done in a fully algebraic setting, which is quite different from our hybrid view. Coecke, Sadrzadeh, and Clark and TCL both use a categorial semantics. The categorial structure of the former is a compact closed category, which means decomposition does not hold. From their categorial model of an adjective (A) noun (N) combination, $A \otimes N$, there is no projection to A and N. TCL internal semantics exploits a different categorial structure, that of a topos, in which projection is valid but the categorial structure of an adjective–noun combination is more complex than a simple (tensor) product of the adjective and noun meaning due to the presence of the functors introduced in Equation (2).

Lewis and Steedman (2013) also seek to combine DS methods within a formal framework. And they, like we, are interested in inference as a testing ground for composition methods. The principal difference between our approach and theirs is that we are interested in testing the predictions of DS at a local predicational level, and we are interested in importing the information from DS into the functors that guide co-composition in TCL. Lewis and Steedman do not translate the effects of DS composition into logical form except to single out most probable senses for arguments and predicates over a limited set of possible senses. They concentrate on pure ambiguities; for example, they offer a representation of two ambiguous words *file* (*remove an outer coating* vs. *depose*) and *suit* (*clothing* vs. *legal document*) and show that in *file a suit*, the ambiguity disappears. It is unclear to us how they actually exploit this disambiguation that they informally describe in inference. Our approach performs disambiguations of homonymous ambiguities, but we argued that this is only a special case of co-composition.

McNally and Boleda (2016) offer empirical and conceptual arguments in favor of the TCL dual approach to meaning and, like us, see DS as an ally in specifying the internal content aspect of composition. However, we offer a much more detailed and specific investigation of the interactions between TCL and particular methods of DS composition. More crucially, we do not see how to semantically interpret vectorial predicates that McNally and Boleda introduce as components of an FS, intensional interpretation. We think that such an interpretation is important to exploit the strengths of FS. It is for this reason that we have investigated how we can go back to TCL functors from our DS composition methods and have pursued a DS approach that is largely isomorphic to the TCL one. McNally and Boleda, however, cite a very important open area of research: Given that the internal content shifts in composition, how is that reflected at the referential or intensional level? To some extent, we answer this in our translation of our DS composition back into TCL functors. However, our method needs further work to reach its full potential.

Boleda et al. (2013) also compared several methods of adjective–noun composition, and we have used their method to determine which iteration of the TENSOR method should produce the best results without being overfitted to the corpus. However, we have compared various composition methods with respect to the predictions on several semantic dimensions; they compare methods with respect to variance from predicted distributions. Thus, our evaluation is one that is external to DS methods; theirs is not.

Other interesting approaches to integrating FS and DS include Melamud et al. (2013) and Beltagy, Erk, and Mooney (2014). Like them, we are interested in rules that

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

relate to lexical inference. However, we integrate these directly into the compositional process using the TCL functor approach.

7. Conclusions

Our article has provided a case study of one way to integrate formal and distributional methods in lexical semantics. We have examined how one formal theory, TCL, corresponds in its treatments of adjective–noun composition to some distributional models of composition that can automatically provide information needed to construct the functors within the TCL construction process. We tested a number of distributional models with regard to the entailments and other semantic relations they predict for adjective–noun compositions; in general, the TENSOR approach was superior to the other methods tested, at least for nouns. We also have at least some indirect evidence that a localist approach like ours, where we use DS to calculate the modifications of each word meaning in context in TCL fashion, is *preferable* to a model in which DS methods are used to calculate the meaning of larger constituents.

As a next step, we plan to extend upon the tensor model, so that both nouns and adjectives are modified by weighting syntactic dependency vectors. This will improve the accuracy of the TENSOR model's predictions for the semantics of the shifted adjectives. We then want to apply the TENSOR model to verbal predications. Decomposition is not just a feature of adjective–noun compositions, but also of verb–argument composition and adverbial modification (Davidson 1967a). These predications are, at least in the vast majority of cases, decomposable into a conjunction of formulas where possibly a functor applies and shifts the meaning of each argument and of the verb itself. We expect to see more shifting with verbs, as they combine with many different types of arguments. The TENSOR approach generalizes to such predications without much modification.

TCL's functor approach to composition with its decomposition property offers a natural place within which to exploit DS composition methods like TENSOR or LVW to inform internal, type-theoretic content. The parallelism between TCL and distributional methods of composition allows us to integrate them in principle throughout the construction of logical form. We can insert modifications due to co-composition at a local level so that this information interacts appropriately with scoping operators and refine the co-composition functors as more contextual information becomes available, something we hope to investigate in future research.

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

Appendix A: List of Adjective–Noun Combinations Studied

adjective	noun	adjective	noun	adjective	noun
physical	activity	great	fear	grand	piano
close	affinity	natural	feeling	succulent	plant
moral	agent	natural	food	intense	pleasure
dead	air	close	friendship	narrow	portion
light	aircraft	complex	function	low	price
close	alley	mutual	fund	close	prisoner
wild	animal	physical	game	great	promise
written	application	large	gathering	solemn	promise
late	application	social	gathering	loud	promise
grand	army	emotional	greeting	physical	property
critical	assessment	social	grouping	personal	question
mutual	attraction	regular	guy	smart	question
investment	bank	natural	habitat	loud	question
muddy	bank	long	hair	economic	reason
spacious	bank	narrow	hall	short	report
popular	bank	rough	handling	great	respect
wooden	bed	central	heating	flexible	sac
flat	beer	large	house	deep	sea
light	beer	publishing	house	middle	section
persisting	belief	nearby	house	moral	sense
false	belief	public	house	formal	series
young	bird	guest	house	large	settlement
heavy	bleeding	public	image	low	shelf
light	blow	bright	image	deep	shelves
dead	body	shocking	image	heavy	shoe
stupid	book	central	importance	dead	silence
heavy	book	basic	ingredient	deep	sleep
interesting	book	basic	instinct	heavy	smoker
small	bread	artificial	intelligence	lyric	soprano
deep	breath	public	interest	young	soprano
large	building	personal	interview	open	space
small	cafeteria	stupid	joke	low	spirit
rough	calculation	moral	judgment	formal	stage
early	cancer	deep	layer	emotional	state
immense	canvas	long	lecture	bronze	statue
modernist	canvas	interesting	lecture	famous	statue
complex	carbohydrate	small	letter	succulent	steak
great	caution	low	limit	heavy	stick
personal	charm	formal	linguistcs	heavy	storm
small	child	early	lunch	great	storm
short	circuit	delicious	lunch	moral	strength
easy	circumstance	long	lunch	basic	substance
polluted	city	purplish	mark	moral	support
socialist	city	light	meal	flat	surface
modernist	city	delicious	meal	rough	surface
middle	class	basic	measure	short	symbol
large	collection	social	media	natural	talent
yellow	color	regular	meeting	difficult	task
small	community	narrow	mind	darjeeling	tea
public	company	open	mind	five-o'clock	tea
formal	complaint	stupid	mistake	stupid	telephone
material	concern	grand	mistake	explosive	temperament
formal	conclusion	critical	moment	formal	test
critical	condition	grand	mountain	legal	testimony
close	contest	forward	movement	secret	testimony
long	corridor	short	news	shocking	testimony
large	country	printed	newspaper	economic	theory
narrow	crack	owned	newspaper	deep	thought
large	crane	small	number	sudden	thought
mechanical	crane	close	observation	ridiculous	thought
feathery	crane	head	officer	dead	time
early	death	public	official	long	time

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

Appendix A
(*continued*)

adjective	noun	adjective	noun	adjective	noun
emotional	decision	underground	organization	light	touch
explosive	device	heavy	paper	heavy	traffic
wooden	dialogue	central	part	close	translation
large	difference	deep	part	low	trick
deep	discussion	great	party	small	tube
functional	disorder	late	payment	basic	unit
wild	dream	large-value	payment	strong	urge
young	dream	cash	payment	underground	vault
light	duty	purplish	pen	wild	vegetation
small	dwelling	yellow	pen	early	version
forward	earnings	flexible	person	easy	victim
physical	effect	short	person	grand	view
functional	element	strong	person	deep	voice
young	elephant	difficult	person	rough	voice
small	enclosure	social	person	low	voice
grand	end	stupid	person	artificial	waterway
material	entity	intense	person	formal	wear
easy	exam	rough	person	head	wind
heavy	expense	emotional	person	natural	world
dead	face	complex	personality	regular	writer

References

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

Baroni, Marco, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The Wacky Wide Web: A collection of very large linguistically processed Web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Baroni, Marco and Roberto Zamparelli. 2010. Nouns are vectors, adjectives are matrices: Representing adjective–noun constructions in semantic space. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1183–1193, Cambridge, MA.

Beltagy, Islam, Katrin Erk, and Raymond J. Mooney. 2014. Probabilistic soft logic for semantic textual similarity. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1210–1219, Baltimore, MD.

Boleda, Gemma, Marco Baroni, Louise McNally, and Nghia Pham. 2013. Intensionality was only alleged: On adjective–noun composition in distributional semantics. In *Proceedings of the 10th International Conference on Computational Linguistics*, pages 35–46, Potsdam.

Bride, Antoine, Tim Van de Cruys, and Nicholas Asher. 2015. A generalisation of lexical functions for composition in distributional semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 281–291, Beijing.

Chi, Yun and Shenghuo Zhu. 2013. Facetcube: A general framework for non-negative tensor factorization. *Knowledge and Information Systems*, 37(1):155–179.

Church, Kenneth W. and Patrick Hanks. 1990. Word association norms, mutual information & lexicography. *Computational Linguistics*, 16(1):22–29.

Coecke, B., M. Sadrzadeh, and S. Clark. 2011. Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis: A Festschrift for Joachim Lambek*, 36(1-4):345–384.

Davidson, Donald. 1967a. The logical form of action sentences. In N. Rescher, editor, *The Logic of Decision and Action*. University of Pittsburg Press.

Davidson, Donald. 1967b. Truth and meaning. *Synthese*, 17(1):304–323.

Ding, Chris, Tao Li, and Wei Peng. 2008. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing.

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

Computational Statistics & Data Analysis, 52(8):3913–3927.
Dinu, Georgiana and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1162–1172, Cambridge, MA.
Dinu, Georgiana, Nghia The Pham, and Marco Baroni. 2013. Dissect—distributional semantics composition toolkit. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 31–36, Sofia.
Do, Quang, Yee Seng Chan, and Dan Roth. 2011. Minimally supervised event causality identification. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 294–303, Edinburgh.
Erk, Katrin and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 897–906, Waikiki.
Erk, Katrin and Sebastian Padó. 2009. Paraphrase assessment in structured vector space: Exploring parameters and datasets. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, pages 57–65, Athens.
Erk, Katrin and Sebastian Padó. 2010. Exemplar-based models for word meaning in context. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference Short Papers*, pages 92–97, Uppsala.
Frege, Gottlob. 1985. On sense and meaning. In A. P. Martinich, editor, *The Philosophy of Language*. Oxford University Press.
Garrette, Dan, Katrin Erk, and Raymond Mooney. 2011. Integrating logical representations with probabilistic information using Markov logic. In *Proceedings of the International Conference on Computational Semantics*, pages 1162–1172, Oxford.
Gaussier, Eric and Cyril Goutte. 2005. Relation between PLSA and NMF and implications. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 601–602, Salvador.
Girju, Roxana, Preslav Nakov, Vivi Nastase, Stan Szpakowicz, Peter Turney, and Deniz Yuret. 2009. Classification of semantic relations between nominals. *Language Resources and Evaluation*, 43:105–121.
Grefenstette, Edward and Mehrnoosh Sadrzadeh. 2011a. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1394–1404, Edinburgh.
Grefenstette, Edward and Mehrnoosh Sadrzadeh. 2011b. Experimenting with transitive verbs in a discocat. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 62–66, Edinburgh.
Howard, William A. 1980. The formulas-as-types notion of construction. In P. Seldin and J. R. Hindley, editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*. Academic Press, pages 479–490.
Kamp, Hans and Barbara Partee. 1995. Prototype theory and compositionality. *Cognition*, 57(2):129–191.
Kripke, Saul. A. 1980. *Naming and Necessity*. Harvard University Press, Cambridge, MA.
Kruszewski, German, Denis Paperno, and Marco Baroni. 2015. Deriving Boolean structures from distributional vectors. *Transactions of the Association for Computational Linguistics*, 3:375–388.
Lee, Daniel D. and H. Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.
Lee, Daniel D. and H. Sebastian Seung. 2000. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, pages 556–562.
Lewis, David. 1970. General semantics. *Synthese*, 22(1):18–67.
Lewis, Mike and Mark Steedman. 2013. Combining distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.
Magidor, Ofra. 2013. *Category Mistakes*. Oxford University Press.
McNally, Louise and Gemma Boleda. 2016. Conceptual vs. referential affordance in concept composition. In Yoad Winter and James Hampton, editors, *Concept Composition and Experimental Semantics/ Pragmatics*. Springer.
Melamud, Oren, Jonathan Berant, Ido Dagan, Jacob Goldberger, and Idan Szpektor. 2013. A two level model for context sensitive inference rules. In *Proceedings of the*

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024

51st Annual Meeting of the Association for Computational Linguistics, pages 1331–1340, Sofia.

Mel'cuk, Igor. 2006. Explanatory combinatorial dictionary. *Open Problems in Linguistics and Lexicography*, pages 225–355.

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

Montague, Richard. 1974. *Formal Philosophy*. Yale University Press, New Haven, CT.

Nivre, Joakim, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, pages 2216–2219, Genoa.

Partee, Barbara. 1995. Lexical semantics and compositionality. In L. Gleitman and M. Liberman, editors, *An Invitation to Cognitive Science: Language, vol. 1*, pages 311–360, Cambridge, MIT Press.

Partee, Barbara H. 2010. Privative adjectives: Subsective plus coercion. In R. Bauerle, U. Reyle, and T. E. Zimmermann, editors, *Presuppositions and Discourse: Essays Offered to Hans Kamp*. Elsevier, pages 273–285.

Pustejovsky, James. 1995. *The Generative Lexicon*. MIT Press.

Putnam, Hilary. 1975. The meaning of 'meaning'. In Keith Gunderson, editor, *Language, Mind and Knowledge. Minnesota Studies in the Philosophy of Science, vol. 7*, pages 131–193.

Roller, Stephen, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of the Twenty Fifth International Conference on Computational Linguistics*, pages 1025–1036, Dublin.

Thater, Stefan, Hagen Fürstenau, and Manfred Pinkal. 2010. Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 948–957, Uppsala.

Toutanova, Kristina, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 252–259, Edmonton.

Toutanova, Kristina and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*, pages 63–70, Hong Kong.

Van de Cruys, Tim. 2011. Two multivariate generalizations of pointwise mutual information. In *Proceedings of the Workshop on Distributional Semantics and Compositionality*, pages 16–20, Portland, OR.

Van de Cruys, Tim, Thierry Poibeau, and Anna Korhonen. 2011. Latent vector weighting for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1012–1022, Edinburgh.

Weeds, Julie, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 1015–1021, Geneva.

Downloaded from http://direct.mit.edu/coli/article-pdf/42/4/703/1807631/coli_a_00264.pdf by guest on 16 July 2024