The Routledge Handbook of Linguistics

Keith Allan

Lexical semantics today

Publication details
https://www.routledgehandbooks.com/doi/10.4324/9781315718453.ch11
Nicholas Asher, Tim Van de Cruys, Márta Abrusán
Published online on: 20 Jul 2015

How to cite: Nicholas Asher, Tim Van de Cruys, Márta Abrusán. 20 Jul 2015, Lexical semantics today from: The Routledge Handbook of Linguistics Routledge
https://www.routledgehandbooks.com/doi/10.4324/9781315718453.ch11
11

Lexical semantics today

Nicholas Asher, Tim Van de Cruys and Márta Abrusán

11.1 Introduction

Lexical semantics is the study of what words mean. In particular, lexical semantics as opposed to formal semantics involves the study of open class words like adjectives, common nouns and verbs, as opposed to functional class words, which contribute to the logical structure of the language – quantifiers, connectives and many adverbs. The latter have been studied in compositional semantics, which uses formal languages like higher-order logic and the lambda calculus to formalize the composition of meaning and the meanings of functional words.

For a long while, lexical semantics developed independently from formal semantics. Apart from a few, daring forays into the formal world (e.g. Dowty 1979), lexical semanticists worked largely in isolation from formal semanticists, who were focused on sentential or discourse meaning and the problem of compositionality. To make the point in a somewhat caricatural fashion, lexical semanticists investigated argument structure, verbal diathesis (shifts in meaning due to shifts in argument structure), polysemy, and meaning decomposition all within various ‘cognitive’ systems lacking rigor and a tie to model theoretic semantics; meanwhile, formal semanticists paid little attention to matters of lexical meaning for open class terms like ordinary nouns and verbs – the meaning of a word $x$ was typically rendered as $x'$. Valuable work was done in both areas but there was something of a missed opportunity in which neither camp profited from the insights of the other.

Recently, however, formal semanticists have begun to take a closer look at the interesting observations by descriptive linguists. An important part of what lexical semantics is today, we think, is to come to grips with the problems that such systems have tackled and the problems that remain to be solved. We will concentrate on some particular phenomena to illustrate the difficulties for formal systems. In a second section, we will also draw attention to a third, and much more recent, approach to lexical semantics, distributional semantics or DS. DS has its origins in various philosophical and older linguistic traditions like those of Wittgenstein and Bloomfield but which has, thanks to the advent of very large corpora of text, opened the door to the automatic acquisition of lexical content. The core idea that is common to Wittgenstein, Bloomfield and DS is that you will know a word by the company
it keeps; the meaning of a word is given by its use, which in turn is given by the contexts of its usage (or surrounding words). A simple implementation of this idea is then to associate with each word a set of contexts in which it occurs along with a relative frequency of its occurrence in such a context. Using computational techniques and very large corpora containing billions of words, researchers can formulate such vectors for words and capture something of word meaning. Like modern formal methods, however, this approach faces deep conceptual and technical problems and interestingly the problems for DS are largely complementary to those for formal lexical semantics. In the final part of this chapter, we will discuss some options for putting formal semantics and DS together, a topic which we feel is currently one of the most exciting in lexical semantics.

11.2 Formal semantics and the lexicon

In this section, we give an introduction to formal lexical semantics and brief survey of some challenges for the view. Formal semantics for natural languages grew out of efforts by logicians, like Tarski and Frege, and mathematicians to specify a representation for the languages of mathematics and logic that would give an abstract characterization of the valid sentences of the language and of the valid inferences that one could make. The manner in which this is done is to specify an abstract set theoretic structure called a model, relative to which one specifies the denotations or semantic values for terms and then rules for combining these values to recursively specify semantic values for complex expressions based on the values for their constituents. Montague (1974) carried out the program of specifying a model for fragments of a natural language by showing how to translate natural language expressions into terms of higher-order logic with the $\lambda$ calculus. He then showed how to derive semantic values or truth conditions for sentences by using the combinatory rules of the $\lambda$ calculus. This provided the first compositional treatment of meaning for natural language in which semantic rules of meaning combination mirrored the rules of syntactic construction for the natural language fragment. Montague’s work, known under the heading of Montague Grammar continues to be the backbone of much formal semantic work almost fifty years after its publication.

Montague’s seminal work, however, gave short shrift to lexical semantics. Apart from the functional or closed class words of the fragments, so-called logical connectives like and, if ... then or quantifiers like some, most and all, the contents of words remained a black box for the compositional system. To give a caricature, the meaning of the word cat is simply $\lambda x.\text{cat}'(x)$. Its meaning or denotation is specified by the type assigned to the $\lambda$ expression; all that the formal system says is that it is a function from individuals to truth values. Such a meaning will return a truth value given any object provided as a value for the variable $x$. The $\lambda$ calculus specifies that such a term when applied to an individual term whose meaning is a simple individual will have the meaning of the type associated with sentences (in the extensional fragment of the language a truth value). Which function cat specifies is not part of the formal system, however.

So what is it to give the meaning of a word? How would we specify such a function? And what kind of data can we use to verify that our specification is on the right track? There are a number of answers in the literature on lexical semantics or theories of word meaning. Cognitive semanticists like Len Talmy and Tom Givón, among others, think that meanings are to be given via a set of cognitively primitive features – which might be pictorial rather than symbolic. According to these semanticists, a lexical theory should provide appropriate cognitive features and lexical entries defined in terms of them. Others in a more logical and
formal framework like Dowty (1979), but also Ray Jackendoff, Roger Shank and other researchers in AI, take a specification of lexical meaning to be given in terms of a set of primitives whose meanings can be axiomatized or computationally implemented. Still others take a 'direct' or denotational view; the function of lexical semantics is to specify the denotation of the various terms, typically to be modeled within some model theoretic framework.

All of these approaches agree that a specification of lexical meaning consists in the specification of some element, whether representational or not, formal or not, that, when combined with elements associated with other words in a wellformed sentence, yields a meaning for a sentence in a particular discourse context. Whatever theoretical reconstruction of meaning one chooses, however, it should be capable of modeling inferences based on lexical information in a precise manner so that the theory of lexical meaning proposed can be judged on its predictions. In addition, the theoretical reconstruction should provide predictions about when sentences that are capable of having a truth value are true and when they are not. This drastically reduces the options for specifying lexical meaning: such a specification must conform with one of the several ways of elaborating meaning within the domain of formal semantics; it must specify truth conditions, dynamic update conditions of the sort familiar from dynamic semantics (Kamp and Reyle 1993; Groenendijk and Stokhof 1991; Veltman 1996), or perhaps provability conditions of the sort advocated by, e.g. Ranta (2004), among others. In §11.3 we will look at two recent formalizations of lexical meaning that use model theoretic and proof theoretic methods.

11.2.1 Four problems for formal lexical semantics

We look now at four problems for formal lexical semantics with an eye to getting clearer about the promise of and obstacles of a formal account.

11.2.1.1 Different types of lexical meaning

Is there only one kind of lexical meaning that feeds into semantic composition or are there several? Lexical semanticists have for the most part described lexical semantics as involving one sort of content, though they might disagree on how such content is to be represented. Compositional semanticists and computational linguists, however, have proposed distinctions between content. The first is the difference between presupposed content and asserted or at-issue content (Strawson 1950; Karttunen 1973). Presupposed content is content that does not interact with logical operators and at-issue content. An example of presupposed content is the existential implication of definite descriptions: both (1a) and (1b) imply that there is a present King of France; the negation does not take the presupposition (that there is a King of France) in its scope.

(1) a. The present King of France is bald.
    b. The present King of France is not bald.

Presuppositions can often be traced back to the presence of some lexical element in the sentence: definite determiners, factive verbs, certain adverbs and many other lexical elements are thought to give rise to presuppositional material. Accordingly, the presuppositional status of a part of the meaning of these lexical items has to be marked in some way or another. An influential theoretical move was to assume that presuppositions must be satisfied in order for the sentence in which it originates to have a truth value. This
condition is typically represented as a definedness condition on the lexical meaning of the presupposition trigger.\(^1\)

There are other types of content that behave similarly to presupposed content vis-à-vis asserted or at-issue content. The content of emotive expressions or epithets, for instance, seems also not to interact with logical operators or attitude verbs (cf. Potts 2005).

\[(2) \text{I hope that bastard gives me back my money.}\]

In (2) the person referred to pejoratively is implied to be bastard (it is not hoped that the person is a bastard); the material in the epithet escapes the scope of the attitude verb.

Many linguists and philosophers have distinguished between implicatures and asserted content as well (Grice 1975). Implicatures are in many ways similar to asserted content but they can typically be denied in subsequent discourse. So, for instance, a numerical adjective can give rise to a so-called scalar implicature in the following sense: to say \(\text{I have four children}\) implicates that I have exactly four children, though I can cancel this implicature by saying \(\text{I have four children – indeed, I have five}\). Implicatures are most commonly thought to arise from pragmatic reasoning about salient alternative utterances in the discourse. But in order for such an account to be predictive, the range of possible alternatives has to be restricted. Such restrictions (aka \textit{Horn-scales}) are usually associated with lexical items.

11.2.1.2 Selectional restrictions and types

Selectional restrictions of an expression \(\varepsilon\) pertain to the type of object denoted by the expression with which \(\varepsilon\) must combine. The verb \(\text{try}\), for instance, imposes on the compositional context ‘to come’ that its subject must be an intentional agent; a verb like \(\text{hit}\) imposes the restriction that its object or internal argument must be a physical object. Thus,

\[(3) \text{Mary hit John’s idea.}\]

is predicted to be difficult to interpret unless the context allows us to interpret \textit{John’s idea} as some sort of physical object (perhaps it is some artifact that he created). Thus, selectional restrictions impose a necessary condition on semantic evaluability; failing to satisfy a selectional restriction is a \textit{category mistake}, a sort of presupposition failure.

One way to handle selectional restrictions is to take them as requiring an argument of a certain type. Montague Grammar (MG) according to which types are identified with the set of entities in the model of that type, countenances two basic types: the type of all entities \(e\), which is identified in a model with the domain of the model, and the type of truth values, \(t\). Further types are defined recursively: for instance, all nouns have the type \(e \rightarrow t\), which is the set of all functions from the domain of entities into the set of truth values. Types have enough semantic content to check the wellformedness of certain predications, but to handle selectional restrictions, the type system must incorporate many more distinctions. For instance it must distinguish various distinct basic types that are subtypes of \(e\); it must distinguish the type for eventualities from the type for physical objects, as well as distinguish the type of informational or abstract objects from these two. Other distinctions will probably be relevant too: the distinction between states and events.

11.2.1.3 Degrees of ambiguity and copredication

A basic issue concerning lexical representation is lexical ambiguity. The most orthodox model of lexical meaning is the monomorphic, sense enumeration model, according to
which all the different possible meanings of a single lexical item are listed in the lexicon as part of the lexical entry for the item. Each sense in the lexical entry for a word is fully specified. On such a view, most words are ambiguous. This account is the simplest conceptually, and it is the standard way dictionaries are put together, and this is also the view found in Montague Grammar or HPSG.

While conceptually simple, this approach fails to explain how some senses are intuitively related to each other and some are not. Words or, perhaps more accurately, word occurrences that have closely related senses are called logically polysemous, while those that do not receive the label accidentally polysemous or simply homonymous. Cruse (1986) suggests copredication as a test to distinguish logical polysemy from accidental polysemy: if two different predicates, each requiring a different sense, predicate properties of different senses of a given word felicitously, then the word is logically polysemous with respect at least to those two senses. Another test is pronominalization or ellipsis: if you can pronominalize an occurrence of a possibly ambiguous word felicitously in a context where the pronoun is an argument of a predicate requiring one sense while its antecedent is an argument of a predicate requiring a different sense, then the word is logically polysemous with respect to those senses. Contrast (4a–b) and (4c–e).

(4) a. #The bank specializes in IPOs. It is steep and muddy and thus slippery.
   b. #The bank specializes in IPOs and is steep and muddy and thus slippery.
   c. Lunch was delicious but took forever.
   d. He paid the bill and threw it away.
   e. The city has 500,000 inhabitants and outlawed smoking in bars last year.

*Bank* is a classic example of an accidentally polysemous word. As (4a–b), show, both the pronominalization and copredication tests produce anomalous sentences, which confirm its status as accidentally polysemous. On the other hand, *lunch, bill* and *city* are classified as logically polysemous, as (4c–e) witness that they pass the tests of copredication and pronominalization.

The distinction between accidental and logical polysemy is not absolute, and there are degrees of relatedness that the felicity of copredications and pronominal tests reflect. But sense enumeration models have no way of explaining the differing degrees of success that copredications appear to have. It is for this reason that many approaches to lexical semantics have adopted some rather more complex way of representing meanings, e.g. one in which each word may have multiple types or underspecified types that may be further specified during the composition process. These are sometimes called polymorphic languages (or lexicons for a language).\(^2\)

### 11.2.1.4 Context sensitivity of meaning

For formal semanticists and philosophers of language, context sensitivity of meaning traditionally stops with anaphoric pronouns and indexical expressions. While it is clear that the sentence *I am hungry* will vary in meaning according to who asserts it, we believe, however, that context dependence pervades the lexicon. More recently, semanticists have argued that other linguistic elements have at least some sort of context sensitivity. These elements include modals, attitude verbs like *believe, want* and *know* and tense. (See Roberts 1989; Veltman 1996 and references therein.) However, philosophers and linguists who call themselves *relevance theorists* have argued that context sensitivity of meaning is pretty much ubiquitous (Sperber and Wilson 1986; Recanati 2004).
One of the intriguing and not well understood observations about the composition of meaning is that when word meanings are combined, the meaning of the result can vary from what standard compositional semantics has led us to expect. While the choice of words obviously affects the content of a predication, prior discourse can also sometimes affect how lexical meanings interact.

(5) All the children were drawing fish. Suzie’s salmon was blue.

In (5) we understand the relationship between Suzie and *salmon* in a complex way: Suzie is drawing a picture of a salmon that is blue. This interpretation is due not only to the genitive construction but also to its discourse environment.

The example above is an example of a phenomenon of coercion. Consider also the following example, discussed at length in the literature, in particular by Pustejovsky (1995).

(6) Julie enjoyed the book.

The intuition of many who work on lexical semantics is that (6) has a meaning like (7) with the *doing something* filled in by an appropriate activity.

(7) Julie enjoyed doing something to (e.g. reading, writing, …) the book.

The intuition is this: *enjoy* requires an event as its direct object as in *enjoy the spectacle, enjoy the view*. This also happens when *enjoy* takes a question as its complement, as in *enjoy (hearing) what he said*. When the direct object of a transitive use of *enjoy* does not denote an event, it is ‘coerced’ to denote some sort of eventuality.

The apparent meaning shifts discussed above should receive as uniform a treatment as possible within a semantic/pragmatic framework of lexical meanings and semantic composition – that is, how lexical meanings compose together to form meanings for larger semantic constituents like propositions or discourses. If the intuitions behind coercion are easy to grasp, modeling coercion by formal means is rather difficult, as we will see in the next section.

11.3 Prior work in formal lexical semantics

In this section we discuss previous work on the themes described above, concentrating on the problem of selectional restrictions, ambiguity and context sensitivity. We will not address the problem of presuppositions, conventional and conversational implicatures any further here: the vast amount of work that has been done on these topics in the last fifty years warrants at least a separate chapter for them. For a recent overview, see Potts (2014) and references therein.

11.3.1 Semantic approaches

We have discussed the sense enumeration model briefly but enough to indicate that it has troubles explaining selectional restrictions, copredication and coercions. In contrast to the sense enumeration model of word meaning is the view that lexical semantics consists of a set of lexical entries, supplemented with a set of maps from one lexical meaning to another. These maps specify, inter alia, coercions. Frege’s theory of meaning includes coercions: he
stipulated that the meaning of terms when they occur within the scope of an intensional operator shifted from their reference to their sense.

Generalizing Frege’s strategy, Nunberg (1979, 1995) proposes that lexical meanings are subject to shifts in predication. A lexical entry specifies a denotation for a referring expression or for a predicate. Let’s take a look at Nunberg’s transfer functions (Nunberg 1995) on some of his examples.

(8) I’m parked out back.

(9) The ham sandwich is getting impatient.

The basic idea is intuitive. In these examples, applying a particular predicate whose argument is required to have type $\alpha$ to an argument whose type is $\beta$, where $\alpha \sqcap \beta = \bot$ (the combining of $\alpha$ with $\beta$ gives a false/uninterpretable proposition) forces either the argument term or the predicate term or both to change their meanings so that the predication can succeed. For instance, ham sandwiches cannot literally get impatient; and if I am standing in front of you, I cannot literally be out back. So what happens is that we shift the meaning of the terms so that the predications succeed: it is my car that is parked out back, and it is the guy who is eating the ham sandwich who is getting impatient. One problem, however, is when exactly is the sense transfer function introduced. If the transfer function were always optionally available, that would lead to vast over-generation problems (10).

(10) The ham sandwich that hasn’t been eaten is on the counter.

(10) would be predicted to have a reading on which the eater of the ham sandwich that has not been eaten is on the counter. It must be that the incompatible selectional restrictions of the verb trigger the introduction of the sense transfer functioning some way.

Suppose that Nunberg’s sense transfer functions work on lexical entries for common noun phrases or N’ (Sag 1981). The result we want for (9) is clear. The ham sandwich should have the following logical form:

\[ (9') \lambda P \text{ the}(x) \left( f(\text{ham sandwich})(x) \land P(x) \right) \]

where $f$ is the transfer function mapping ham sandwiches to people who are eating or have just eaten them. The problem is that we only become aware of the need for a type adjustment mechanism in building the logical form of the sentence when we try to put the subject noun phrase together with the verb phrase; to add the transfer function to work over the contribution of the common noun, we need a rather specific and ad hoc processing of the syntactic structure into logical form (Egg 2003).

Another suggestion is to have the full DP (determiner phrase) or the verb have to undergo sense transfer. But if it is the entire noun phrase or DP that is shifted, we get incorrect truth conditions. Consider (11).

(11) George enjoyed many books last weekend.

A straightforward application of the Nunberg strategy yields the following logical form:

\[ (11') f(\text{many books}(x)) \exists e(\text{enjoy}(g, x, e) \land \text{last weekend}(e)) \]
The transfer function shifts the meaning of the entire quantifier, and so presumably the quantifier ranges over the elements in the image of the transfer function, namely some sort of eventualities. For the sake of concreteness, we could assume that they are events of reading books. But if (11′) says that there were many book reading events that George enjoyed over the weekend, this is compatible with there being just one book that George read over and over again that weekend. Clearly, this is not a possible reading of (11); this proposal gives incorrect predictions.

In fact, this observation about (11) constitutes a problem for the best-known treatment of coercion, Pustejovsky’s Generative Lexicon or GL approach (Pustejovsky 1995). Key to its treatment of these phenomena is the notion of a qualia structure, which involves contents associated with the core meaning of a noun. Following Moravscik (1975), GL postulates that many nouns contain information about other objects and eventualities that are associated as roles with the denotation of the noun. GL hypothesizes that in predications like (6), verbs like begin and enjoy, which require an eventuality for their internal arguments, select as their arguments the eventualities encoded in the feature structures associated with nouns like cigarette. GL’s qualia structures pack Nunberg’s sense transfer function into the lexical entry for nouns. The data show that the qualia model is insufficiently general and too inflexible to succeed, as well as having technical problems (Asher 2011).

In GL only the verb is involved in the coercion of its arguments; for instance, the aspectual verbs and enjoy coerce their object argument to be of event type, and the event type is specified by one of the qualia. But this is empirically incorrect. For instance, the subject of an aspectual verb may determine what the type of its theme or event argument is.

\[(12)\]
\[\begin{align*}
\text{a. } & \text{The janitor has begun (with) the kitchen.} \\
\text{b. } & \text{The cleaners have started the suits.} \\
\text{c. } & \text{The exterminator has begun (with) the bedroom.}
\end{align*}\]

In each of these there is a coercion to a particular type of event for the object argument of the aspectual verb. Yet it is obvious that the noun phrases in object position do not by themselves supply the requisite events in (12a–c).

A further problem is this. Let us suppose that the denotation of the noun phrase the book in (6) is transformed into some sort of eventuality denoting expression. If that is the case, then how can we access in subsequent discourse the referent of the book?

\[(13)\] Julie enjoyed the book. It was a mystery.

These observations are familiar but show that we cannot shift the meaning of the book to some sort of eventuality. Or at least if we do, whatever process is responsible for the shift must also allow the book to retain its original, lexical meaning and its original contribution to logical form.

To avoid these problems, we could resort to an alternative transfer strategy: shift the meaning of the verbs rather than the meaning of common nouns or DPs. This can in principle avoid the problem noted with (11). But this shift leads to another set of difficulties. Consider the following ellipsis facts.3

\[(14)\]
\[\begin{align*}
\text{a. } & \text{I’m parked out back, and Mary’s car is too.} \\
\text{b. } & \text{?I own a car that is parked out back and Mary’s car does too.} \\
\text{c. } & \text{Mary enjoys her garden and John his liquor.} \\
\text{d. } & \text{?Mary enjoys digging in her garden, and John his liquor.}
\end{align*}\]
If we transfer the sense of *parked out back* to the property of owning a car that is parked out back to account for the parking examples, then the ellipsis in (14a) should be odd or it should give rise to the following reading: Mary’s car owns a car that is parked out back as well. But the ellipsis is fine, and it lacks this reading. Similarly if we transfer the sense of *enjoy* to something like *enjoy digging* to make sense of the predication that Mary enjoys her garden, then we should predict that the ellipsis in (14c) should be bad much the way (14d) is. But the ellipsis in (14c) is fine. The transfer function approach thus faces hurdles on all the options we have surveyed.4

11.3.2 Pragmatic approaches

In contrast to GL’s attempt to locate the mechanisms for coercion and aspect selection within the lexical semantic entry for particular words, there is a pragmatic approach to such phenomena. One might suppose that in fact coercion and type adjustment in general is not part of semantics at all, but rather a pragmatic mechanism. The ellipsis facts seem to point to a pragmatic account according to which the appropriate enrichments to predicates and/or arguments occur after semantics has done its job, and hence phenomena like ellipsis have been resolved.

Relevance theorists (Sperber and Wilson 1986; Recanati 2004; Carston 2002; Recanati 1993) attempt to capture these inferences by appealing to a pragmatic process of free enrichment. However, the clearest, most formally explicit pragmatic account is Stanley and Szabo’s hidden variables approach (Stanley and Szabo 2000). Such approaches attempt to analyze phenomena such as coercion or aspect selection as involving pragmatic reasoning. For Stanley and Szabo the pragmatic reasoning involves finding contextually salient values for hidden variables.

Egg (2003) develops something like Stanley and Szabo’s approach for coercion. One postulates for each coercing predicate \( \lambda x \varphi \) two hidden variables, one higher-order relational variable and another that will end up being related to \( x \). *Enjoy*, which is understood to take an event as a direct object, would now look something like this:

\[
\lambda y \lambda e \lambda z \text{enjoy}(z, e) \land R(e, y)
\]

When no coercion is present \( R \) could be resolved to identity, but when the direct object of the verb involves denotations to non-event-like objects, then \( R \) would be resolved to some contextually salient property. So, for instance, *enjoying playing a soccer game* intuitively involves no coercion, and the variable introduced by the DP that ranges over playing soccer game events would be identified with the event variable that is the direct object argument of *enjoy*. However, *Laura enjoyed a cigarette* would involve a coercion; in this case the variable introduced by the DP ranging over cigarettes would be related via some contextually salient relation \( R \) like smoking to get the logical form:

\[
\exists y \exists e (\text{enjoy}(l, e) \land \text{smoke}(e, y) \land \text{cigarette}(y))
\]

This approach has an advantage over GL and other sense transfer views in that it gets the quantification cases right: *enjoying many books* does not, on this view, mean that one enjoys many events of reading books. On the other hand, this approach has nothing to say about when coercion takes place. It would need to be supplemented by some theory of types, type checking and repair to say when coercion would be triggered. In fact none
of the approaches we have looked at so far deals with this problem in a formal and systematic way.

The pragmatic approach also seems to analyze coercions at the wrong linguistic level. Pragmatic principles are supposed to follow from general principles of rational interaction between agents, and so they are expected to be universal, but we have already seen that coercions are typically language specific. In addition, we should expect pragmatic enrichment to allow coercions, say from objects to eventualities, whenever the grammar allows for an eventuality reading in the argument. But this is not true. Consider (17).

(17)  
a. The reading of the book started at 10.  
b. #The book started at 10.  
c. John started the book.

It is perfectly straightforward to get the eventuality reading for the object of \textit{start} (17c) in an out of the blue context. But it is impossible to do this when the intransitive form of \textit{start} is used with \textit{the book} in subject position, even though an eventuality reading for the subject of intransitive \textit{start} is available, as seen by (17a).

Such shifts in the predication relation are lexically governed. It is the verb \textit{enjoy} that requires an event but also which licenses a change in the predicational glue between it and its object when it is not of the right sort. As argued in Asher (2011), these licensings are proper to \textit{certain arguments} of the verb or rather to the syntactic realization of other arguments of the predicate. We are not dealing with a phenomenon of general pragmatic strengthening or repair but rather with a problem at the syntax/semantics interface, which is how to account for the differences in (17).

Finally, coercions are language dependent. Direct translation of (18), for example, is not good in Chinese.

(18) Julie started a book.

In Chinese, one has to say (19) instead:

(19) Julie started reading/writing a book.

The coercion that inserts ‘reading/writing’ actions is not licensed in Chinese when ‘start’ is used.

Coercion is a semantic phenomenon, not a pragmatic one. Pragmatic accounts use principles that are supposed to follow from general principles of rational interaction between agents; they are expected to be universal. So we should expect pragmatic enrichment to allow coercions, say from objects to eventualities, whenever the grammar allows for an eventuality reading in the argument. But this is not true.

11.3.3 Recent type driven theories

Recent formal lexical semantic theories developed to deal with systematic ambiguity and coercion try to remedy the difficulties of previous semantic accounts. One approach is to change the relation of predication that holds between the predicate and the argument. A type clash between the type of the argument and the type presupposition of the predicate induces not a type shift in either the argument or the predicate but rather a type shift on the predication
relation itself, which is implemented by introducing a functor that is inserted around the argument. The meaning of the argument does not shift – it remains what it always was; the predicate also retains its original meaning, but what changes is how they combine.

The Type Composition Logic (TCL) developed in Asher (2011) uses a rich system of types to guide the construction of logical form, which then gets a standard intensional semantics. Type information is treated as a kind of presupposed information. Selectional restrictions of predicate are like presuppositions, because they must be satisfied somehow or else the predication will be anomalous and difficult to assign truth conditions to. (3) is an example where the type presuppositions of the predicate cannot be satisfied. They are also not part of ordinary ‘at-issue’ content because they escape the scope of operators like negation and question operators, as Asher (2011) details. TCL’s basic subtypes of e are types of individuals and verify intuitive subtyping relations like the fact that MAN is a subtype of ANIMAL. A common noun has in TCL like in Montague Grammar a functional type; the type of the lambda bound argument of the predicate \( \lambda x \text{man}(x) \) is, however, not the type MAN in TCL but rather PHYSICAL-OBJECT, in order to allow for a distinction between semantic anomaly and semantic falsity.\(^5\) In TCL uttering that’s a man while pointing to a statue yields a coherent proposition that is false; the type assigned to the individual variable in the subject DP (PHYSICAL OBJECT) matches the type presupposition of the predicate is a man. On the other hand, the sentence the set \( \{\emptyset, \{\emptyset\}\} \) is a man is semantically anomalous and does not yield a proposition at all, because the type on the individual level variable of the DP (INFORMATIONAL OBJECT) is incompatible with the type presupposition of the predicate.

The second point of comparison between selectional restrictions and presuppositions is that like presuppositions, selectional restrictions may be satisfied in various ways. When a predicate requires that an argument has a type \( \tau \), then as long as its argument type \( \sigma \) is compatible with \( \tau \) (i.e. \( \sigma \sqcap \tau \neq \bot \)), type checking succeeds and the predicate and the argument combine together in TCL with the argument taking as type the meet \( \sqcap \) of \( \sigma \) and \( \tau \). But other forms of type presupposition satisfaction are possible.

In coercions there is no common subtype for the type presupposition of the predicate and the argument; the contents of a bottle is of a type incompatible with that of the bottle that contains it, and similarly, the eventuality required by start or enjoy is a distinct and incompatible type with that of a physical or abstract object like a book – these types have different identity and individuation criteria. However, predicates like start license a justification of their type requirements by allowing one to introduce material that links the actual argument to an object of the desired type in a manner akin to the sort of presupposition justification employed in bridging. This changes the relation between the predicate and its arguments: the predicate allows a natural transformation from a given type to the desired type. This transformation is translated into the language of logical forms and has a truth conditional content. So in particular, for example (6), the transformation licenses the introduction of a functor that applies to the verb’s entry, allowing the result to combine with the original argument of the verb. Here is what the functor looks like (20).

\[
(20) \quad \lambda P \lambda u \lambda e \lambda e' (\exists e(P(v, e, e') \land \varphi(e, u)))
\]

Note that the functor introduces into logic a description of an eventuality as the second argument of enjoy, which will go in for the \( \lambda \) bound variable \( P \); note also that this description is itself underspecified, because without a particular context, we do not know exactly what Julie did to the book that she enjoyed. All this is reflected in the logical form that results
from applying the functor to \textit{enjoy} and then combing the result with the object and subject arguments of \textit{enjoy} in (6). See (21).

\begin{equation}
\exists x \exists e' \exists e (\text{book}(x) \land \text{enjoy}(j, e, e') \land \phi(j, x, e))
\end{equation}

\(\phi(j, x, e)\) is the underspecified description of the eventuality \(e\). Note that \(e\) is now the argument of \textit{enjoy}, as is required, but that the discourse entity introduced by \textit{a book} is also available for anaphoric reference. The meaning of \textit{book} remains what it always was; it is just that it combines in a different way with the verb from the usual function application method. Note also that the verb has not changed its meaning either. Unlike the predicate shifting approach of Nunberg, the verb \textit{enjoy} means what it usually means even in predications involving coercion. As a consequence, examples involving ellipsis such as (14c) discussed above are no problem.

With regard to degrees of ambiguity, TCL follows a radical program of encoding ambiguities in the type system rather than as separate lexical entries. So, accidentally polysemous expressions are given a disjoint union type that predications can disambiguate when the predicate selects for one of the disjoined types. On the other hand logically polysemous expressions like \textit{book} require a more complex type construction that makes essential use of the categorial semantics with Asher (2011). In some sense books are both physical and informational objects and different predicates will select either the physical or the informational aspect, an idea already expressed in Cruse (1986). Which aspect is selected makes a difference to how we count books and so different predications on ‘dual aspect’ nouns can actually alter quantificational domains. For instance, we are not counting the same objects in (22a) and (22b); in the first she need not have opened every (or even any) copy of \textit{The Generative Lexicon} in the library, whereas in the second, she has to have burned every copy of that book in the library.

\begin{enumerate}
\item Mary mastered every book on lexical semantics in the library.
\item Mary burned every book on lexical semantics in the library.
\end{enumerate}

Another modern formal lexical system comes from Modern Type Theories (MTTs) (Ranta 2004; Luo 2010). MTT has two novel features with regard to lexical semantics; first it does not treat common nouns as predicates but rather as simple atomic types. In MTT each common noun is interpreted as the most specific fine-grained type associated with the noun in TCL; this is the noun’s sole semantic contribution. The second is the introduction of a non-subsumptive notion of subtyping Luo (2010). This notion of subtyping is called \textit{coercive subtyping}.

Coercive subtyping provides an interesting way to interpret various linguistic coercions. The basic coercive subtyping mechanism coerces \(f(a)\) into \(f(c(a))\) by inserting the coercion \(c\) into a gap between \(f\) and \(a\). The coercion like the functor in TCL in principle need affect neither the meaning of the predicate nor its argument. So for (6), \(\text{Julie enjoyed a book}\), an MTT would assume the coercion

\begin{equation}
\text{Book} <_{\text{reading}} \text{Event}
\end{equation}

so that (6) is coerced into (24):

\begin{equation}
\text{Julie enjoyed reading a book.}
\end{equation}
In the MTT treatment, we have considered only one possible coercion (23): from ‘enjoy a book’ to ‘enjoy reading a book.’ As we noted in the previous section, however, there are in fact context-dependent ‘multiple coercions’: e.g. (6) could have meant ‘Julie enjoyed writing a book’; there could also be several reading events of that book. Thus, coercive subtyping is a context-sensitive notion of subtyping, and it is necessary to limit the scope of a coercion, but it is not clear that all ambiguous coercions can be so handled.

11.3.4 Interim conclusion

We have surveyed some problems for formal lexical semantics and sketched some proposals for solving them. We have seen how types and type theories, as well as tools from compositional semantics like presupposition, are crucial tools for this area. However, there is one big problem we have not mentioned yet. To be useful to computer scientists and computational linguists, lexical theories have to have good coverage; that is, they have to codify meanings of large numbers of expressions so as to support lexical inferences. None of the theories we have surveyed so far have done this or are in any position to do so in the near future. In the next sections we turn to DS which has the great advantage of automatically acquiring meanings for large numbers of words.

11.4 Distributional semantics

11.4.1 Introduction

Since the last decade of the twentieth century, a new paradigm for the modeling of lexical semantics has emerged, somewhat orthogonal to and independent of developments in formal semantics. The goal of this new paradigm – called distributional semantics – is to automatically extract the meaning of words by looking at their distribution in text, and comparing these distributions within a mathematical model. The unsupervised nature of distributional semantics, and consequently its ability to process large volumes of text in a fully automatic way, make it an attractive method for a data-driven and broad-coverage model of lexical semantics.

Inspired by Harris (1954), work within distributional semantics relies on the distributional hypothesis of meaning, according to which words that appear within the same contexts tend to be semantically similar. In the spirit of this well-known adage, numerous algorithms have been developed that try to capture the semantics of single words by looking at the contexts of these words, and comparing those contexts within a mathematical vector space model. The vector space model was developed as a method for information retrieval in the beginning of the 1970s. The idea is that each document of a text collection is represented as a vector in vector space, such that the features of the vector are constituted by the document’s words. The distance between vectors is then taken as a measure of semantic similarity. A user’s query may then be presented within the same vector space, and the most similar documents may be retrieved.

At the beginning of the 1990s, researchers have started to apply the idea of a vector space model to other semantic tasks, giving rise to the distributional semantic approach. Methods within distributional semantics have been able to achieve rather impressive results. Famously, one of the best-known methods within distributional semantics – latent semantic analysis (LSA) – in its initial paper (Landauer and Dumais 1997) reached a score of 64.38 percent on multiple-choice synonym questions of the Test of English as a Foreign Language (TOEFL) which, as Landauer and Dumais note, is adequate for admission to many US
universities. Since then, distributional models of semantics have been shown to reach a perfect score on the test (Bullinaria and Levy 2012).

11.4.2 Overview

The vector space model applies straightforwardly to similarity calculations between words. The two words for which the semantic similarity is to be calculated, are represented as vectors of the words’ various contexts. Table 11.1 shows an example matrix $M$ containing vectors for four different target words (using context words as features), automatically extracted from a large corpus. In this example the set of target words is

$$T = \{\text{raspberry, strawberry, car, truck}\}$$

and the set of basic elements is

$$B = \{\text{red, yellow, tasty, fast, eat, drive}\}$$

<table>
<thead>
<tr>
<th></th>
<th>red</th>
<th>yellow</th>
<th>tasty</th>
<th>fast</th>
<th>eat</th>
<th>drive</th>
</tr>
</thead>
<tbody>
<tr>
<td>raspberry</td>
<td>728</td>
<td>6</td>
<td>592</td>
<td>1</td>
<td>568</td>
<td>0</td>
</tr>
<tr>
<td>strawberry</td>
<td>1035</td>
<td>4</td>
<td>638</td>
<td>0</td>
<td>682</td>
<td>0</td>
</tr>
<tr>
<td>car</td>
<td>392</td>
<td>145</td>
<td>1</td>
<td>487</td>
<td>0</td>
<td>370</td>
</tr>
<tr>
<td>truck</td>
<td>104</td>
<td>46</td>
<td>0</td>
<td>44</td>
<td>0</td>
<td>293</td>
</tr>
</tbody>
</table>

The value in matrix cell $(i, j)$ is the co-occurrence frequency of word $i$ with value $j$. In the example above, the word *red* appears 728 times within the context of the word *raspberry*, and the word *drive* appears 370 times in the context of the word *car*.

Note that context is a determining factor in the nature of the semantic similarity that is induced. A broad context window (e.g. a paragraph or document) yields broad, topical similarity, whereas a small context window yields tight, synonym-like similarity. This has led a number of researchers to use the dependency relations that a particular word takes part in as contextual features.

Once contextual feature vectors for each word are created, the vectors may be compared within a vector space. A graphical representation of a vector space using just two dimensions (*red* and *fast*), is given in Figure 11.1. A typical semantic vector space will have somewhere between 2,000 and 100,000 dimensions.

![Figure 11.1](image-url)

*Figure 11.1* A graphical representation of a word vector space. The angle between the vectors for strawberry and raspberry is much smaller than the angle between the vectors for strawberry and car
In order to formally compute the textual overlap between two word vectors $\vec{v}$ and $\vec{w}$ we need a proper vector similarity measure. One of the best known and most widely used measures for vector similarity is the cosine, which computes the angle between two vectors. Cosine is easy to compute, and often achieves the best results. The measure is calculated as shown in (25).

\[
\cos(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{||\vec{v}|| \cdot ||\vec{w}||}
\]

where $\vec{v} \cdot \vec{w}$ is the dot product between vector $\vec{v}$ and $\vec{w}$, both of length $k$ and $||\bullet||$ represents the norm of a vector.

\[
\vec{v} \cdot \vec{w} = \sum_{i=1}^{k} v_i w_i
\]

When word vectors are normalized to a vector length of 1, a similarity computation between two vectors boils down to a simple dot product.

This procedure represents the basic intuition behind word similarity computations. Often, researchers apply a number of extra preprocessing steps in order to improve similarity computations. One of these preprocessing steps consists of feature weighting, in which more informative features are given more weight. Another preprocessing step might consist of the application of dimensionality reduction algorithm (such as singular value decomposition), where the abundance of overlapping features is reduced to a limited number of latent and allegedly more meaningful dimensions. For an elaborated overview of semantic word space models, the interested reader is referred to Turney and Pantel (2010).

### 11.4.3 New developments

To date, work on the automatic acquisition of semantics has mostly dealt with individual words. The modeling of meaning beyond the level of individual words – i.e. the composition of word meanings into larger semantic units – remains a conceptual and practical problem for the distributional approach. A number of models have been explored that try to capture compositional phenomena within a distributional framework (Baroni et al. 2014). Another interesting development, somewhat related to the work on the modeling of compositionality, is represented by research on the computation of word meaning in context. By adapting the global meaning of a word to its particular use within a specific context, these models aim to compute the precise meaning of a particular word instance in context, implicitly performing word sense disambiguation (Erk et al. 2013).

### 11.5 Where do we go from here?

We have reviewed some themes in the two major contemporary currents of lexical semantics, and we have seen that formal lexical semantics and distributional semantics each have attractive features. Currently, a hot research topic is to integrate these two approaches. We see two ways this integration could go.
11.5.1 DS furnishes types to FS

If we follow recent FS proposals that assign a component of lexical content in the form of types, a conservative integration of the two would allow DS to determine the contents of basic types associated with words – in effect providing type content for lexical expressions. DS could also investigate the presence of more general types. One could use various similarity metrics to see whether certain words clustered together in a natural similarity class. FS would then provide a ready-made composition mechanism for these types.

To compose type level information, one would have to be able to recast the content contained within distributional meanings in terms of some idiom that can be interpreted within the symbolic and compositional framework. One idiom that type theories typically avail themselves of is the proof theoretic one: the type meaning of a term \( t \) should provide evidence (or an algorithm) that an object or set of objects is in the denotation of \( t \). So for instance the type meaning of \( \text{boy} \) would be defined as follows: \( \|\text{cat}\|: \) a function that given any individual either provides a justification that the object is a cat or returns \( \perp \). What would such a justification consist of? Presumably, it would be a collection of properties typically associated with cats; any object possessing most of those properties would be justifiably be taken to be a cat. While it seems that the meaning objects of at least some versions of DS promise to contain such information, it remains an open research problem how to carry out the proposal in detail.

11.5.2 A dynamic interaction between DS and FS

A more ambitious integration of DS and FS would be to divide up the task of composing meaning. Undoubtedly FS’s main strength has been in the analysis of close class terms, while DS’s main strength lies in the rich, distributed modeling of open class terms. In addition, new developments in DS have successfully explored models of meaning beyond the level of words, effectively calculating the changes in word meaning that take place when words interact. An ambitious integration of FS and DS would combine the best of both frameworks. DS would supply algebraic style denotations not only for words but for compositions up to and including a complete verbal complex with non-quantificational arguments – i.e. a simple clause without tense or modal information – dynamically updating word meanings as they combine with other words. FS would then contribute the meanings of functional elements (quantifiers, connectives, modals, tense morphemes) to the output of the DS computation. Or we could go still further and replace FS’s construal of functional meanings with distributional functions that operate on distributional vectors and return other vectors as output (Baroni et al. 2014). Still, it remains unclear how the denotations from FS would combine with the representations of DS, or even what the appropriate DS representations are for clauses. Furthermore, we do not have a clear idea of the compositional principles that would guide the integration of the meanings of functional terms with DS denotations. Nonetheless, we believe an integrated approach holds great potential for a proper modeling of compositionality. DS is able to provide the lexical richness and the ability to model meaning interactions that are missing from traditional formal semantic theories.
11.6 Conclusions

Lexical semantics has become at the beginning of the twenty-first century an exciting field. There have been impressive computational developments resulting in distributional lexical semantics that provides wide coverage assignments of automatically acquired lexical meanings. Formal lexical semantics has also made impressive strides in the precision with which it has attacked recalcitrant problems of lexical meaning like contextual sensitivity and ambiguity, enabling it to take account of perceptive descriptions of the data that have long resisted formal analysis. There is also the exciting prospect of marrying the best that formal semantics has to offer with the empirical methods of distributional semantics. We do not know yet how this last part of the story is going to go, but we hope that we are all around to see how it evolves.

Acknowledgement

Thanks to Cedric Dégremont for helpful suggestions, especially on the title.

Notes

1 More recently, approaches that attempt to predict the presupposition from the at-issue content and pragmatic principles have been proposed as well. See Potts (2014) for an overview.
2 Underspecification is a tool or method devised by linguists working on ambiguity in sentential and discourse semantics.
3 Thanks to Alexandra Aramis for suggesting these sorts of examples.
4 See Egg (2003); Asher (2011) for more detailed criticisms.
5 TCL does exploit the fine-grained types in the compositional process however. For man, the finest-grained type associated with its lambda bound variable is man.

References