Lifetime Achievement Award

The Lost Combinator

It linked all perplexed meanings
Into one perfect peace.
—Procter and Sullivan, 1877, "The Lost Chord"

Mark Steedman
University of Edinburgh
Informatics Forum
steedman@inf.ed.ac.uk

Let me begin by thanking the Association for Computational Linguistics and its Executive Committee for conferring on me the great honor of their Lifetime Achievement Award for 2018, which of course I share with all the wonderful students and colleagues that have made many essential contributions to this work over many years.

At the heart of the work that I have been pursuing over my research lifetime so far, whether in parsing and sentence processing, spoken language understanding, semantics, or even in musical understanding by machine, there lies a theory of natural language grammar that brings parsing, compositional semantics, statistical modeling, and logical inference into the closest possible relation. This theory of grammar is *combinatory*, in the sense that its operations are type-dependent and restricted to strictly string-adjacent phonologically or graphologically-realized inputs, and *categorial*, in the sense that those operands pair a syntactic type with a type-transparent semantic representation or logical form.

I'd like to use this opportunity to briefly address three questions that revolve around the theory of grammar, both combinatory and otherwise. The first question concerns the way that Combinatory Categorial Grammar (CCG) was developed with a number of colleagues, over a number of stages and in slightly different forms. The second is an essentially evolutionary question of why natural language grammar should take a combinatory form. The third question is that of what the future holds for CCG and other structural theories of grammar in computational linguistics and NLP in the age of deep learning.

I have called this talk "The Lost Combinator" in homage to the Victorian era poem "The Lost Chord," in the hope of suggesting that the theoretical development of CCG has always been empirical, rather than axiomatic, in search of the simplest explanation of the facts of language, rather than for confirmation of linguistic received opinion, however intuitively salient.

doi:10.1162/coli_a_00328

1. In the Beginning

In the late 1960s (when I was a psychology undergraduate at the University of Sussex under Stuart Sutherland, and then started as a graduate student in artificial intelligence at Edinburgh under Christopher Longuet-Higgins), a broad community of theoretical linguists, psychologists, and computational linguists saw themselves as all working on the same problem, under the definition provided by the "transformational" theory of grammar proposed by Chomsky (1957, 1965), using theories of psycholinguistic processing, language acquisition, and language evolution proposed by Lashley (1951), Miller, Galanter, and Pribram (1960), Miller (1967), and Lenneberg (1967), theories of natural language semantics proposed by Carnap (1956), Montague (1970), and Lewis (1970), and computational models of parsing such as those proposed by Thorne, Bratley, and Dewar (1968) and Woods (1970). (I myself was so convinced that this program would succeed that I believed it was time to apply the same methods to other cognitive faculties, taking as my research project for Ph.D. their application to the interpretation of music by machine, following the lead of Max Clowes [1971] in machine vision.)

Almost immediately, this consensus fell apart. First, Chomsky himself was among the first (1965) to recognize that transformational rules, though descriptively revealing, were so expressive as to have little explanatory force, and required many apparently arbitrary constraints (Ross, 1967). Second, psychologists realized that psycholinguistic measures of processing difficulty of sentences bore almost no relation to their transformational derivational complexity (Marslen-Wilson 1973; Fodor, Bever, and Garrett, 1974). Finally, computational linguists attempting to implement transformational grammars as parsers realized that they were spending all their time implementing even more constraints on rules, in order to limit search arising from overgeneration (Friedman 1971; Gross 1978). (Meanwhile, I realized that the problem had not in fact been solved, and returned to natural language processing, thanks to a postdoc at Sussex with Philip Johnson-Laird.)

This disillusion wasn't just a case of internal academic squabbling. There were also a couple of influential reports commissioned by the U.S. and UK governments that ended funding for machine translation (MT) and artificial intelligence (AI) (Pierce et al. 1966; Lighthill 1973). As a result of the second of these reports, which determined that AI was never going to work, PhDs in artificial intelligence like my classmate Geoff Hinton and myself spent ten years or so after graduation in psychology departments (in my case, at the Universities of Sussex and Warwick), until yet another report said AI was working after all and that Britain and the U.S. were falling behind Japan in this vital area. As a result, I could get hired again in computer science, first briefly back at Edinburgh, and then at the University of Pennsylvania (I learned a lesson from this odyssey that I have tried to remember whenever I have been appointed to a committee to report on anything, which is that while reports very rarely do any good, they can very easily do a great deal of harm.)

Meanwhile, as a result of these conflicts, the scientific study of language fragmented. The linguists swiftly abjured any responsibility for their grammars ("Competence") bearing any relation to processing ("Performance"). Because the psychologists could hardly abandon Performance, they in turn became agnostic about grammar, retreating to context-free surface grammar (which they tended to refer to as "parsing strategies"), or a touchingly optimistic belief in its emergence from neural models. Meanwhile, the computational linguists (whose machines were growing exponentially in size and speed from the 16K byte core of the machine that supported the whole group when I started my graduate studies, on to levels that would soon permit parsing

the entire contents of the then embrionic Web) similarly found that very little of what the linguists and psychologists cared about was usable at scale, and that none of it significantly improved *overall* performance over very much simpler context-free or even finite-state methods that the linguists had shown to be incomplete. The reason of course was Zipf's law, which means that the events with respect to which the low-level methods are incomplete are off in the long tail.

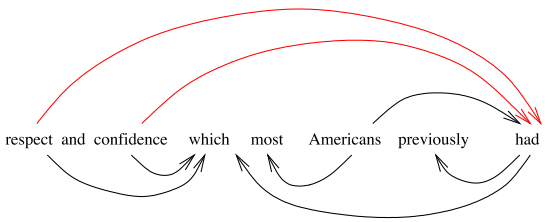
It also became apparent to a few computationalists working on speech, MT, and information retrieval that the real problem was not grammar but ambiguity and its resolution by world-knowledge, and that the solution lay in probabilistic models (Bar-Hillel 1960/1964; Spärck Jones 1964/1986; Wilks 1975; Jelinek and Lafferty 1991) (although it was not immediately apparent how to combine statistical models with grammar-based systems without making obviously false independence assumptions).

Nevertheless, as any red-blooded psychologist had always insisted, the divorce between competence and performance that everyone else had accepted did not make any sense. The grammar and the processor had to have evolved in lock-step, as a package deal, for what could be the evolutionary selective advantage of a grammar that you cannot process, or a parser without a grammar?

It seemed equally obvious that surface syntax and the underlying semantic or conceptual representation must also be closely related, since the only reasonable basis for child language acquisition that has ever been on offer is that the child attaches language-specific grammar to a universal conceptual relation or "language of mind" (Miller 1967; Bowerman 1973; Wexler and Culicover 1980). It seemed to follow that radically new theories of grammar were needed.

2. The Problem of Discontinuity

Theoretical linguists agree that the central problem for the theory of grammar is discontinuity or non-adjacent dependency between predicates and their arguments:



(1)

Chomsky described discontinuity in terms of movement, which was known to be formally very unconstrained. By contrast, the ATN parser used in the LUNAR project (Woods, Kaplan, and Nash-Webber 1972) reduced all discontinuity to local operations on registers (Thorne, Bratley, and Dewar 1968; Bobrow and Fraser 1969; Woods 1970).

In particular, unbounded *wh*-dependencies like the above were handled by: (a) putting a pointer into a * or HOLD register as soon as the "which" was encountered without regard to where it would end up; and (b) retrieving the pointer from HOLD when the verb needing an object "had" was encountered without regard to where it had started out. (It also included an ingenious mechanism for coordination called SYSCONJ, which one finds even now being reinvented on an almost yearly basis—cf. Woods [2010].) A * register was also used for *wh*-constructions within a systemic grammar framework by Winograd (1972, pages 52–53) in his inspiring conversational program SHRDLU.

However, it was unclear how to generalize the HOLD register to handle the multiple long-range dependencies, including crossing dependencies, that are found in many other languages. In particular, if the HOLD register were assumed to be a stack, then the ATN becomes a two-stack machine (since we are already implicitly using one stack as a PDA to parse the context-free core grammar).

On the computational side at least, the reaction to this impass took two distinct forms. Both reactions took the form of trying to reduce the two major operators of the transformation theory, substitution of immediate constituents, or what is nowadays called "Merge," and "Move," or displacement of non-immediate constituents, to one. On the one hand, Lexical Functional Grammar (Bresnan and Kaplan 1982) and Head-driven Phrase Structure Grammar (Pollard and Sag 1994) followed Kay (1979) in making *unification* the basis of movement and merger. Because unification can pass information across unbounded structures, this can be thought of as reducing Merge to Move.

On the other hand, Generalized Phrase Structure Grammar (Gazdar 1981), Tree Adjoining Grammar (TAG; Joshi and Levy 1982), and Combinatory Categorial Grammar (CCG, Ades and Steedman, 1982) sought to reduce Move to various forms of local merger. In particular, the latter authors suggested that the same stack could be used to capture both long-range dependency and recursion in CCG.[1]

3. Why Does Natural Language Allow Discontinuity?

Natural language grammar exhibits discontinuity because semantically language is an applicative system. Applicative systems (such as programming languages) support the twin notions of: (a) Application of a function/concept to an argument/entity; and (b) Abstraction, or the definition of a new function/concept in terms of existing ones.

Language is in that sense inherently computational. It seems to follow that linguistics is (or should be) inherently computational as well. (Of course, it does not follow that computationalists have nothing to learn from linguistics.)

There are two ways of modeling abstraction in applicative systems: Taking abstraction itself as a primitive operation (λ-calculus, LISP):

$$\begin{aligned} &\text{a. } \mathit{father}\ \mathit{Esau} \Rightarrow \mathit{Isaac} \\ &\text{b. } \mathit{grandfather} = \lambda x.\mathit{father}\,(\mathit{father}\,x) \\ &\text{c. } \mathit{grandfather}\ \mathit{Esau} \Rightarrow \mathit{Abraham} \end{aligned} \tag{2}$$

or Defining abstraction in terms of a collection of operators on strictly adjacent terms aka Combinators, such as function composition (Combinatory Calculus, MIRANDA).

$$\text{b}'.\ \mathit{grandfather} = \mathbf{B}\,\mathit{father}\,\mathit{father} \tag{3}$$

The latter does the work of the λ-calculus without using any variables.

1 As well as at Warwick and Edinburgh, much of the early work on CCG was done during 1980–1981 as a visiting fellow at the University of Texas at Austin, under funding from the Sloan Foundation to Stanley Peters and Phil Gough, my first introduction to the United States.

Despite the resemblance of the "traces" (or copies) and "operators" (or complementizer positions) of the transformational theory to the λ-operators and variables of applicative systems of the first kind, natural language actually seems to be a system of the second, combinatory kind. The evidence stems from the fact that natural language deals with all sorts of fragments that linguists do not normally think of as semantically typable constituents, *without the use of any phonologically realized equivalent of variables*, such as pronouns:

(4)
a. Give [Anna books]$_?$ and [Manny records]$_?$
b. (Mother to child): There's a DOGGIE! [You LIKE]$_?$ # the doggie.
c. Food that you must [wash$_{VP/NP}$ [before eating]$_{(VP\backslash VP)/NP}$]$_?$.
d. ik denk dat ik$_1$ Henk$_2$ Cecilia$_3$ [zag$_1$ leren$_2$ zingen$_3$]$_?$

These fragments are diagnostic of a Combinatory Calculus based on $\mathbf{B}^n$, $\mathbf{T}$, and the "duplicator" $\mathbf{S}^n$, plus application (Steedman 1987; Szabolcsi 1989; Steedman and Baldridge 2011).[2]

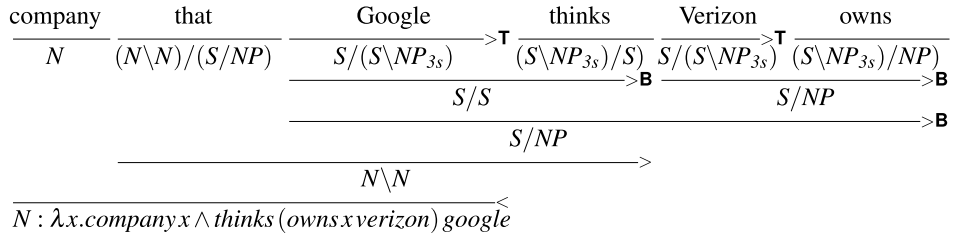
4. Combinatory Categorial Grammar (CCG)

CCG lexicalizes all bounded dependencies, such as passive, raising, control, exceptional case-marking, and so forth, via lexical logical form. All syntactic rules are Combinatory—that is, binary operators over contiguous phonologically realized categories and their logical forms. These rules are restricted by a Combinatory Projection Principle, which in essence says they cannot override the decisions already taken in the language-specific lexicon, but must be consistent with and project unchanged the directionality specified there. All such language-specific information is specified in the lexicon: The combinatory rules like composition are free and universal. All arguments, such as subjects and objects, are lexically type-raised to be functions over the predicate, as if they were morphologically cased as in Latin, exchanging the roles of predicate and argument.

All long-range dependencies are established by contiguous reduction of a *wh*-element, such as $(N\backslash N)/(S/NP)$, with an adjacent non-standard constituent with category S/NP, formed by rules of function composition.

$$\dfrac{\dfrac{\text{company}}{N} \quad \dfrac{\dfrac{\text{that}}{(N\backslash N)/(S/NP)} \quad \dfrac{\dfrac{\text{Verizon}}{S/(S\backslash NP_{3s})}\,{>}\mathbf{T} \quad \dfrac{\text{owns}}{(S\backslash NP_{3s})/NP}}{S/NP}\,{>}\mathbf{B}}{N\backslash N}\,{>}}{N : \lambda x.company\,x \wedge owns\,x\,verizon}\,{<} \tag{5}$$

2 The combinators are identified as $\mathbf{B}^n$, $\mathbf{T}$, and $\mathbf{S}^n$ for historical reasons. This combinatory calculus is distinct from the categorial type-logic stemming from the work of Lambek (1958) that similarly underpins Type-Logical Grammar (Oehrle 1988; Hepple 1990; Morrill 1994; Moortgat 1997), in which the grammar is a logic, rather than a calculus, and some but not all of the CCG combinatory rules are theorems.



(6)

The combinatory rules synchronize composition of the syntactic types shown here with corresponding composition of logical forms (suppressed in the derivations above), to yield the logical forms shown as λ-terms for the resulting nouns N.

To capture the construction in Example (4c), whose syntactic derivation we pass over here, we also need rules based on the duplicator **S**:

a. "wash X before eating X"
b. $VP/NP : \lambda x.before\ (eat x)(wash x) \equiv \mathbf{S}\ (\mathbf{B}\ before\ eat)\ wash$ (7)

To capture constructions like Example (4d) (whose syntactic derivation is similarly suppressed), we also need rules based on second-order composition $\mathbf{B}^2$:

a. "Y saw X teach W to sing."
b. $((S\backslash NP)\backslash NP)\backslash NP : \lambda w \lambda x \lambda y.help\ (teach\ (sing\ w)\ wx)\ xy \equiv \mathbf{B}^2\ sees\ (\mathbf{B}\ teach\ sing)$ (8)

CCG thus reduces the operator MOVE of transformational theory to applications of purely adjacent operators—that is, to recursive combinatory MERGE.

Interestingly, the latest "minimalist" form of the transformational theory has also proposed that Move should be relabeled as an "internal" form of standard or "external" Merge (Chomsky 2001/2004, page 110), though without providing any formal basis for the reduction other than identifying internal Merge as "a grammatical transformation." (If anything deserved the soubriquet "the lost combinator," it would be this notional unitary combination of application and abstraction in a single perfect operator, linking or merging all types, as in the epigraph to this article.)

4.1 Expressivity of CCG

$\mathbf{B}^2$ rules allow us to "grow" categories of arbitrarily high valency, such as $((S\backslash NP_y)\backslash NP_x)\backslash NP_w$. As we saw earlier, in some Germanic languages like Dutch, Swiss German, and West-Flemish, serial verbs are linearized using such rules to require crossing discontinuous dependencies. Thus, $\mathbf{B}^2$ rules give CCG slightly greater than context-free power.

Nevertheless, CCG is still not as expressive as movement. In particular, we can only capture permutations that are what is called "separable," where separability is related to the idea of obtaining the permutations by rebracketing and rotating sister nodes (Steedman 2018).

For example, for the categories of the form $A|B$, $B|C$, $C|D$, and D, it is obvious by inspection that we cannot recognize the following permutations:

i. $*B|C \quad D \quad A|B \quad C|D$
ii. $*C|D \quad A|B \quad D \quad B|C$ (9)

This generalizatiion appears likely to be true cross-linguistically for the components of this form for the NP "These five young boys":

i. *Five boys these young
ii. *Young these boys five (10)

Twenty-one of the 22 separable permutations of "These five young boys" are attested (Cinque 2005; Nchare 2012). The two forbidden orders are among the unattested three.[3]

The probability of this happening by chance is the prior probability of the hypothesis itself—that is the reciprocal of 24 choose 2—times 3 choose 2, the number of ways of predicting two impossible orderings out of three unattested ones, given by the following:

$$p = \frac{\binom{3}{2}}{\binom{24}{2}} = \frac{(3 * 2)/2}{(24 * 23)/2} = \frac{6}{552} \approx 0.01 \tag{11}$$

—that is, about one in a hundred. (If the sole predicted order that remains unattested so far were to be attested, this chance would fall to about one in 250.)

The number of separable permutations grows much more slowly in n than $n!$, the number of all permutations. For example, for $n = 8$, around 80% of the permutations are non-separable. There are obvious implications for the problem of alignment in machine translation and neural semantic parsing, to which we will return.

In 1986, I moved to Computer and Information Science at Penn, first as a visitor (incredibly, partly still under Sloan funding—at least, that was what Aravind Joshi told me), and then as a member of faculty, where much of the further development of CCG was worked out. Crucially, Aravind's students proved in a series of papers (Vijay-Shanker, Weir, and Joshi 1987; Joshi, Vijay-Shanker, and Weir 1991; passim) that the "shared stack" claim of Ades and Steedman (1982) was correct, by showing that both CCG and Aravind Joshi's TAG were weakly equivalent to Linear Indexed Grammar (Gazdar 1988), a new level of the Language Hierarchy characterized by the (Linear) Embedded Push-down Automaton (cf. Kuhlmann, Koller, and Satta 2015).

The class of languages characterizable by these formalisms fell within the requirements of what Joshi (1988) called "mild context sensitivity" (MCS), which proposed as a criterion for what could count as a computationally "reasonable" theory of natural languages—informally speaking, that they are polynomially recognizable, and exhibit constant growth and some limit on crossing dependencies. However, the MCS class is much much larger than the CCG/TAG languages, including the multiple context free languages and even (under certain further assumptions) the languages of Chomskian minimalism, so it seems appropriate to distinguish TAG and CCG as "slightly non-context-free" (SNCF, with apologies to the French railroad company).

3 Dryer (2018) notes four languages that have been claimed to have the first of these orders as basic. However, examination of the source materials makes it clear that the examples in question involve extraposed APs rather than A, which Cinque correctly excludes (cf. Cinque 2010).

4.2 CCG for Natural Language Processing

Despite its SNCF complexity, CCG was originally assumed to be totally unpromising as a grammar formalism for parsing, because of the extra derivational ambiguity introduced by type-raising and the combinatory rules, particularly composition. However, these supposedly "spurious" constituents also show up under coordination, and as intonational phrases. So any grammar with the same coverage as CCG will engender the same degree of nondeterminism in the parser (because it is there in the grammar). In fact, this is just another drop in the ocean of derivational ambiguity that faces all natural language processors, and can be handled by exactly the same statistical models as other varieties.

In particular, the head-word dependency models pioneered by Don Hindle, Mats Rooth, Mike Collins, and Eugene Charniak are straightforwardly applicable (Hockenmaier and Steedman 2002b; Clark and Curran 2004). CCG is also particularly well-adapted to parsing with "supertagger" front ends, which can be optimized using embeddings and long short-term memory (LSTM) (Lewis and Steedman 2014; Lewis, Lee, and Zettlemoyer 2016).

CCG is now quite widely used in applications, especially those that call for transparency between semantic and syntactic processing, and/or dislocation, such as machine translation (Birch and Osborne 2011; Mehay and Brew 2012), machine reading (Krishnamurthy and Mitchell 2014), incremental parsing (Pareschi and Steedman 1987; Niv 1994; Xu, Clark, and Zhang 2014; Ambati et al. 2015), human–robot interaction (Chai et al. 2014; Matuszek et al. 2013), and semantic parser induction (Zettlemoyer and Collins 2005; Kwiatkowski et al. 2010; Abend et al. 2017). Some of my own work with the same techniques has returned to their application in musical analysis (Granroth-Wilding and Clark 2014; McLeod and Steedman 2016), and shown that CCG grammars of the same SNCF class and parsing models of the same statistical kind are required there as well: It is only in the details of their compositional semantics that music and language differ very greatly.

Rather than reflecting on this substantial body of work in detail, I'd like to conclude by examining two further more speculative questions. The first is an evolutionary question: Why should natural language be a combinatory calculus in the first place? The second is a question about the future development of our subject: Will CCG and other grammar-based theories continue to be relevant to NLP in the age of deep learning and recursive neural networks? I'll take these questions in order in the next two sections.

5. Why is Language Combinatory?

Natural language looks like a distinctively combinatory applicative system because **B**, **T**, and **S** evolved independently, in our animal ancestors, to support planning of action sequences before there was any language (Steedman 2002). Even pure reactive planning animals like pigeons need application. Composition **B** and Substitution **S** are needed for seriation of actions. (Even rats need to compose sensory-motor actions to make plans.) Macaques can form the concept "food that you need to wash before eating"). Type-raising **T** is needed to map tools onto actions that they allow (affordances). (Chimpanzees and some other animals can plan with tools.) Second-order combinators like $\mathbf{B}^2$ are needed to make plans with arbitrary numbers of entities, including non-present tools, and crucially including other agents whose cooperation is yet to be obtained. Only humans seem to be able to do the latter.

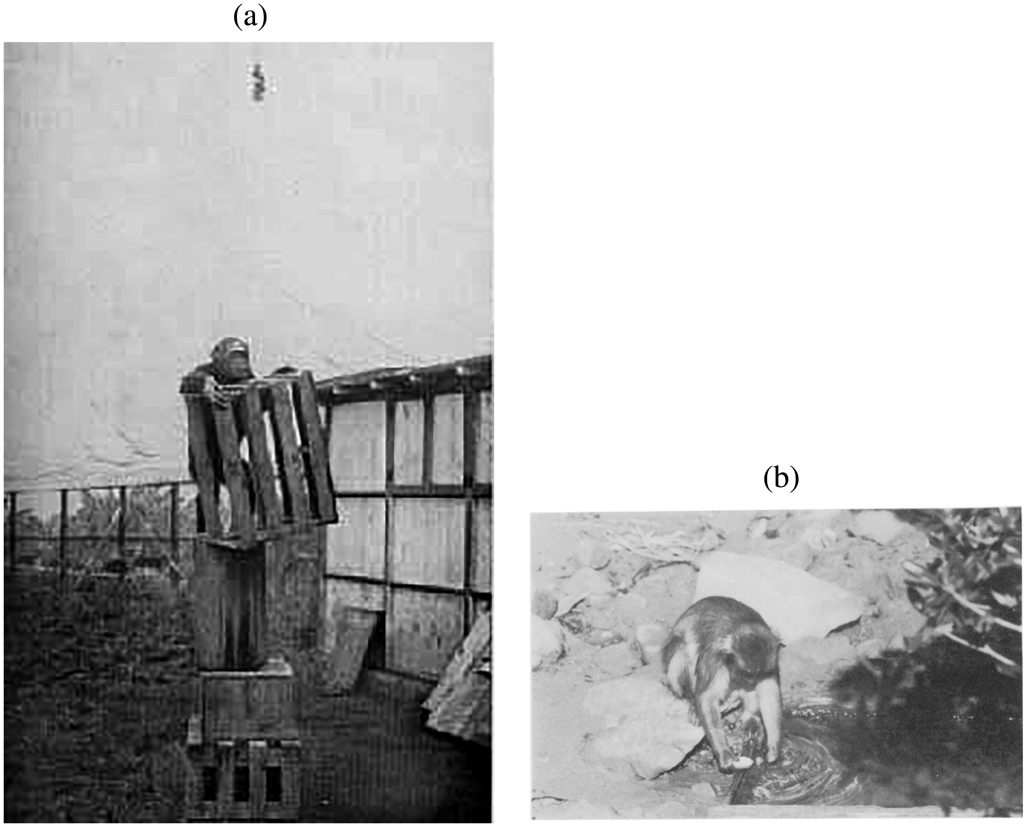


Figure 1
(a) Köhler 1925; (b) Kawai 1965.

Thus, chimpanzees solve the (mistitled) monkey and bananas problem, using tools like old crates to gain altitude in order to obtain objects that are otherwise out of reach (Köhler 1925) (Figure 1(a)).

Köhler's chimpanzee Grande's planning amounts to composing (**B**) affordances (**T**) of crates and so on, such as actions of moving, stacking, and climbing-on them. Similarly, macacques can apply **S** (**B** *before eat*) *wash* to a sweet potato (Kawai 1965) (Figure 1(b)). Interestingly, Köhler and much subsequent work showed that the crates and other tools have to be there in the situation already for the animal to be able to plan with them: Grande was unable to achieve plans that involve fetching crates from the next room, even if she had recently seen them there.

More generally, the problem of planning can be viewed as the problem of *search* for a sequence of actions in a lattice or labyrinth of possible states. Crucially, such search has the same recursive character as parser search. For example, there are both chart-based dynamic-programming and stack-based shift-reduce-style algorithms for the purpose. The latter seem a more plausible alternative in evolutionary terms, as affording a mechanism that is common to both semantic interpretation and parsing, rather than requiring the independent evolution of something like the CKY algorithm.[4]

4 This point seems to have escaped Berwick and Chomsky (2016, pages 175–176, n9).

Planning therefore provides the infrastructure for linguistic performance, as well as the operators for competence grammar, allowing the two to emerge together, as what was referred to above as an evolutionary "package deal."

6. CCG in the Age of Deep Learning

I hope to have convinced you that CCG grammars are both linguistically and evolutionarily explanatory, as well as supporting practical semantic parsers that are fast enough to parse the Web.

Nevertheless, like other supervised parsers, CCG parsers are limited by the weakness of parsing models based on only a million words of WSJ training data, no matter how cleverly we smooth them using synthetic data and embeddings. Moreover, the constructions they are specifically needed for—unbounded dependencies and so forth—are, as we have noted, rare, and off in the long tail. As a consequence, supervised parsers are easy to equal on performance overall by using end-to-end models (Vinyals et al., 2015).

This outcome says more about the weakness of treebank grammars and parsers than about the strength of deep learning. Nevertheless, in the area of semantic parser induction for arbitrary knowledge graphs such as FreeBase (Reddy, Lapata, and Steedman 2014), I would say that CCG and other grammar-based parsers have already been superseded by semisupervised end-to-end training of deep neural networks (DNNs) (Dong and Lapata 2016, 2018; Jia and Liang 2016), raising the question of whether DNNs will replace structured models for practical applications in general.

DNNs are effective because we don't have access to the universal semantic representations that allow the child to induce full CCG for natural languages and that we really ought to be using both in semantic parser induction, and in building knowledge graphs. Because the FreeBase Query language is quite unlike any kind of linguistic logical form, let alone like the universal language of mind, it may well be more practically effective with small and idiosyncratic data sets to induce semantic parsers for them by end-to-end deep neural brute force, rather than by CCG semantic parser induction.

But is it really possible that the problem of parsing could be completely solved by recurrent neural network (RNN)/LSTM, perhaps augmented by attention/a stack (He et al., 2017; Kuncoro et al., 2018)? Do semantic-parsing-as-end-to-end-translation learners actually *learn syntax*, as has been claimed?

It seems likely that end-to-end semantic role labeler parsers and neural machine translation will continue to have difficulty with long-range *wh*-dependencies, because the evidence for their detailed idiosyncrasies is so sparse.

For example, both English and French treat embedded subject extraction as a special case, either involving special bare complement verb categories (English) or a special complementizer "qui" (French).

a. A woman who I believe (*that) won
b. Une femme que je crois qui/*que à gagné (12)

(This is actually predicted by CCG, which says that if you are an SVO language with relatively rigid word order, like English and French, then you will not in general be able to extract embedded subjects, whereas in verb-initial languages like Welsh or verb-final languages like Japanese and German, you will either be able to extract both embedded subjects and objects, or neither—[Steedman 1987, 2000].)

Not surprisingly, at the time of writing, a well-known end-to-end DNN translation system Near You shows no sign of having learned these syntactic niceties. Starting from a legal English, we get an ambiguous French sentence whose back-translation to English translation means something different:

$$\begin{array}{l} \text{This is the company that the agency told us owned the title.} \\ \neq \text{C'est la compagnie que l'agence nous a dit détenir le titre.} \\ = \text{This is the company that the agency told us to hold the title.} \end{array} \quad (13)$$

Similarly, if we start with French subject extraction, we obtain ungrammatical English (which happily translates back into the original correct French):

$$\begin{array}{l} \text{C'est la compagnie que l'agence nous a dit qui détient le titre.} \\ \neq \text{*This is the company that the agency told us *that holds the title.} \\ = \text{C'est la compagnie que l'agence nous a dit qui détient le titre.} \end{array} \quad (14)$$

If instead we start with legal English subject extraction, using a bare complement, the (correct) translation again includes an infinitival that is incorrectly back-translated:

$$\begin{array}{l} \text{This is the company that they said had owned the bank.} \\ \neq \text{C'est la compagnie qu'ils ont dit avoir possédé la banque.} \\ = \text{*This is the company they said they owned the bank.} \end{array} \quad (15)$$

By contrast, supervised CCG parsers do rather well on embedded subject extraction (Hockenmaier and Steedman 2002a; Clark, Steedman, and Curran 2004), which is a fairly frequent construction that happens to be rather unambiguously determined by the parsing model.

End-to-end methods are similarly unreliable when faced with long-range agreement, even when the source sentence is completely unambiguous in this respect:

$$\begin{array}{l} \text{The banks think that the chairman owns the stock, and know that it is stolen.} \\ \neq \text{Les banques pensent que le président est propriétaire du stock et sait qu'il est volé.} \\ = \text{Banks think that the president owns the stock and knows it is stolen.} \end{array} \quad (16)$$

Moreover, in principle at least, these constructions could be learned by grammar-based semantic parser induction from examples, using the methods of Kwiatkowski et al. (2010, 2011) and Abend et al. (2017).

These observations make it seem likely that there will be a continued need for structured representations in tasks like QA where long-range dependencies matter. Nevertheless, like rock n'roll, deep learning and distributional representations are clearly here to stay. The future in parsing for such tasks probably lies with hybrid systems using neural front ends for disambiguation, and grammars for assembling meaning representations.

7. The Shape of Things to Come

The most important open problem in NLP remains the fact that natural language understanding involves inference as well as semantics, and we have no idea of the

meaning representation involved. If your question is *Has Verizon bought Yahoo?*, the text will almost certainly answer it many times over. But it is almost equally certain to present the information in a form that is not immediately compatible with the form of the question. For example, sentences like the following use: a different verb; a noun rather than a verb; an implicative verb; an entailing quantification; a negated entailment; a modal verb; and disjunction:

a. Verizon purchased Yahoo.	("Yes")	
b. Verizon's purchase of Yahoo	("Yes")	
c. Verizon managed to buy Yahoo.	("Yes")	
d. Verizon acquired every company.	("Yes")	(17)
e. Verizon doesn't own Yahoo.	("No")	
f. Yahoo may be sold to Verizon.	("Maybe")	
g. Verizon will buy Yahoo or Yazoo.	("Maybe not")	

To arrive at a meaning representation language that is form-independent (and ultimately language-independent), we are using CCG parsers to machine-read the Web for relations between typed named entities. in order to detect consistent patterns of entailment between relations over named entities of the same types, using directional similarity over entity vectors representing relations. We then build an entailment graph (cleaning it up and closing it under relations such as transitivity [cf. Berant et al. 2015]). Cliques of mutually entailing relations in the entailment graph then constitute paraphrases that can be collapsed to a single relation identifier (Lewis and Steedman 2013a). (This can be done across text from multiple languages [Lewis and Steedman 2013b].)

We can then replace the original naive semantics for relation expressions with the relevant paraphrase cluster identifiers, and reparse the entire corpus using this now both form-independent and language-independent semantic representation, building an enormous knowledge graph, with the entities as nodes, and the paraphrase identifiers as relations.

To answer questions concerning the knowledge in this graph, we parse questions Q into the same form-independent semantics representation, which is now the language of the knowledge graph itself. To answer the question, we use the knowledge graph and the entailment graph, and the following rule:

a. if Q or anything that entails Q is in the knowledge graph, then answer in the positive.
b. If $\neg Q$ or the negation of anything that Q entails is in the knowledge graph, then answer in the negative. (18)

For this to work, we need complex expressions (including negation, auxiliaries, modals, implicative verbs, etc.) to be nodes in their own right in both knowledge and entailment graphs. We shall then be able to discard hand-built knowledge graphs like Freebase in favor of a truly organic semantic net built in the language of mind, obviating the need to learn end-to-end transduction between semantic representations and the language of the knowledge graph.

If this project is successful, the language-independent paraphrase cluster identifiers will perform the function of a "hidden" version of the decompositional semantic

features in semantic representations like those of Katz and Postal (1964), Jackendoff (1990), Moens and Steedman (1988), White (1994), and Pustejovsky (1998), while the entailment graph will form a similarly hidden version of the "meaning postulates" of Carnap (1952) and Fodor, Fodor, and Garrett (1975). Such semantic representations are essentially distributional, but with the advantage that they can be combined with traditional logical operators such as quantifiers and negation.

This proposal for the discovery of hidden semantic primitives underlying natural language semantics stands in contrast to another quite different contemporary approach to distributional semantics that seeks to use dimensionally reduced vector-based representations of collocations to represent word meanings, using linear-algebraic operations such as vector and tensor addition and multiplication in place of traditional compositional semantics. It is an interesting open question whether vector-based distributional word embeddings can be used, together with directional similarity measures, to build entailment graphs of a similar kind (Henderson and Popa 2016; Chang et al. 2018). It is quite likely that some kind of hybrid approach will be needed here too, to combat the eternal silence of the infinite spaces of the long tail.

8. Conclusion

Algorithms like LSTM and RNN may work in practice. But do they work in theory? In particular, can they learn all the syntactic stuff in the long tail, like non-constituent coordination, subject extractions, and crossing dependency, in a way that will support semantic interpretation? If they are not actually learning syntax, but are instead learning a huge finite-state transducer or a soft augmented transition network, then by concentrating on them as mechanisms for natural language processing, we are in danger of losing sight of the computational linguistic project of also providing computational explanations of language and mind.

Even if we convince ourselves that something like SEQ2TREE is a psychologically real learning mechanism, and that children learn their first language by end-to-end mapping of the sentences of their language onto the situationally afforded structures of the universal language of mind, we still face the supreme challenge of finding out what that universal semantic target language looks like. We will not get an answer to that question unless we can rise above using SQL, SPARQL, and hand-built ontologies and logics such as OWL as proxies for the hidden language of mind, to use machine learning for what it is really good at, namely, finding out such hidden variables and their values, for use in a truly *natural* language processing.

Acknowledgments

The work was supported in part by ERC Advanced Fellowship GA 742137 SEMANTAX; ARC Discovery grant DP160102156; a Google Faculty Award and a Bloomberg L.P. Gift Award; a University of Edinburgh and Huawei Technologies award; my co-investigators Nate Chamber and Mark Johnson; the combinator found, Bonnie Webber; and all my teachers and students over many years.

References

Abend, Omri, Tom Kwiatkowski, Nathaniel Smith, Sharon Goldwater, and Mark Steedman. 2017. Bootstrapping language acquisition. *Cognition*, 164:116–143.

Ades, Anthony and Mark Steedman. 1982. On the order of words. *Linguistics and Philosophy*, 4:517–558.

Ambati, Bharat Ram, Tejaswini Deoskar, Mark Johnson, and Mark Steedman. 2015. An incremental algorithm

for transition-based CCG parsing. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 53–63, Denver, CO.
Bar-Hillel, Yehoshua. 1960/1964. A demonstration of the nonfeasability of fully automatic machine translation. In Yehoshua Bar-Hillel, editor, *Language and Information*, Addison-Wesley, Reading, MA, pages 174–179.
Berant, Jonathan, Noga Alon, Ido Dagan, and Jacob Goldberger. 2015. Efficient global learning of entailment graphs. *Computational Linguistics*, 42:221–263.
Berwick, Robert and Noam, Chomsky. 2016. *Why Only Us: Language and Evolution*. MIT Press, Cambridge, MA.
Birch, Alexandra and Miles Osborne. 2011. Reordering metrics for mt. In *Proceedings of the Association for Computational Linguistics*, pages 1027–1035, Portland, OR.
Bobrow, Daniel and Bruce Fraser. 1969. An augmented state transition network analysis procedure. In *Proceedings of the 1st International Joint Conference on Artificial Intelligence*, pages 557–567, Washington, DC.
Bowerman, Melissa. 1973. Structural relationships in children's utterances: Syntactic or semantic? In T. Moore, editor, *Cognitive Development and the Acquisition of Language*. Academic Press, pages 197–213.
Bresnan, Joan and Ronald Kaplan. 1982. Introduction: Grammars as mental representations of language. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, MIT Press, Cambridge, MA, pages xvii–lii.
Carnap, Rudolf. 1952. Meaning postulates. *Philosophical Studies*, 3:65–73. Reprinted as Carnap, 1956:222–229.
Carnap, Rudolf, editor. 1956. *Meaning and Necessity*, 2nd edition, University of Chicago Press, Chicago.
Chai, Joyce, Lanbo She, Rui Fang, Spencer Ottarson, Cody Littley, Changsong Liu, and Kenneth Hanson. 2014. Collaborative effort towards common ground in situated human-robot dialogue. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*, pages 33–40, Bielefeld.
Chang, Haw Shiuan, Ziyun Wang, Luke Vilnis, and Andrew McCallum. 2018. Distributional inclusion vector embedding for unsupervised hypernymy detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 485–495, New Orleans, LA.
Chomsky, Noam. 1957. *Syntactic Structures*. Mouton, The Hague.
Chomsky, Noam. 1965. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.
Chomsky, Noam. 2001/2004. Beyond explanatory adequacy. In Adriana Belletti, editor, *Structures and Beyond: The Cartography of Syntactic Structures*, Oxford University Press, pages 104–131.
Cinque, Guglielmo. 2005. Deriving Greenberg's universal 20 and its exceptions. *Linguistic Inquiry*, 36:315–332.
Cinque, Guglielmo. 2010. *The Syntax of Adjectives*, Linguistic Inquiry Monograph 57. MIT Press, Cambridge MA.
Clark, Stephen and James R. Curran. 2004. Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Barcelona.
Clark, Stephen, Mark Steedman, and James R. Curran. 2004. Object-extraction and question-parsing using CCG. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 111–118, Barcelona.
Clowes, Max. 1971. On seeing things. *Artificial Intelligence*, 2:79–116.
Dong, Li and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin.
Dong, Li and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of ACL*, pages 731–742, Melbourne.
Dryer, Matthew. 2018. On the order of demonstrative, numeral, adjective, and noun. *Language*, 94:(to appear).
Fodor, Janet, Jerry Fodor, and Merrill Garrett. 1975. The psychological unreality of semantic representations. *Linguistic Inquiry*, 6:515–531.
Fodor, Jerry A., Thomas Bever, and Merrill Garrett. 1974. *The Psychology of Language*. McGraw-Hill, New York.
Friedman, Joyce. 1971. *A Computer Model of Transformational Grammar*. Elsevier, New York.
Gazdar, Gerald. 1981. Unbounded dependencies and coordinate structure. *Linguistic Inquiry*, 12:155–184.
Gazdar, Gerald. 1988. Applicability of indexed grammars to natural languages. In Uwe Reyle and Christian Rohrer,

editors, *Natural Language Parsing and Linguistic Theories*. Reidel, Dordrecht, pages 69–94.
Granroth-Wilding, Mark and Mark Steedman. 2014. A Robust Parser-Interpreter for Jazz Chord Sequences. *Journal of New Music Research*, 43:355–374.
Gross, Maurice. 1978. On the failure of generative grammar. *Language*, 55:859–885.
He, Luheng, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 473–483.
Henderson, James and Diana Popa. 2016. A vector space for distributional semantics for entailment. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2052–2062, Berlin.
Hepple, Mark. 1990. *The Grammar and Processing of Order and Dependency: A Categorial Approach*. Ph.D. thesis, University of Edinburgh.
Hockenmaier, Julia and Mark Steedman. 2002a. Acquiring compact lexicalized grammars from a cleaner treebank. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, pages 1974–1981, Las Palmas.
Hockenmaier, Julia and Mark Steedman. 2002b. Generative models for statistical parsing with Combinatory Categorial Grammar. In *Proceedings of the 40th Meeting of the Association for Computational Linguistics*, pages 335–342, Philadelphia, PA.
Jackendoff, Ray. 1990. *Semantic Structures*. MIT Press, Cambridge, MA.
Jelinek, Fred and John Lafferty. 1991. Computation of the probability of initial substring generation by stochastic context-free grammars. *Computational Linguistics*, 17:315–323.
Jia, Robin and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22, Berlin.
Joshi, Aravind. 1988. Tree-adjoining grammars. In David Dowty, Lauri Karttunen, and Arnold Zwicky, editors, *Natural Language Parsing*. Cambridge University Press, Cambridge, pages 206–250.
Joshi, Aravind and Leon Levy. 1982. Phrase structure trees bear more fruit than you would have thought. *Computational Linguistics*, 8:1–11.
Joshi, Aravind, K. Vijay-Shanker, and David Weir. 1991. The convergence of mildly context-sensitive formalisms. In Peter Sells, Stuart Shieber, and Tom Wasow, editors, *Processing of Linguistic Structure*. MIT Press, Cambridge, MA, pages 31–81.
Katz, Jerrold and Paul Postal. 1964. *An Integrated Theory of Linguistic Descriptions*. MIT Press, Cambridge, MA.
Kawai, Masao. 1965. Newly-acquired pre-cultural behavior of the natural troop of Japanese monkeys on Koshima islet. *Primates*, 6:1–30.
Kay, Martin. 1979. Functional grammar. In *Proceedings of the 5th Annual Meeting of the Berkeley Linguistics Society*, pages 142–158, Berkeley, CA.
Köhler, Wolfgang. 1925. *The Mentality of Apes*. Harcourt Brace and World, New York.
Krishnamurthy, Jayant and Tom Mitchell. 2014. Joint syntactic and semantic parsing with combinatory categorial grammar. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1188–1198, Baltimore, MD.
Kuhlmann, Marco, Alexander Koller, and Giorgio Satta. 2015. Lexicalization and generative power in CCG. *Computational Linguistics*, 41:187–219.
Kuncoro, Adhiguna, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1426–1436.
Kwiatkowski, Tom, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1223–1233, Cambridge, MA.
Kwiatkowski, Tom, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical generalization in CCG grammar induction for semantic parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1512–1523, Edinburgh.
Lambek, Joachim. 1958. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–170.

Lashley, Karl. 1951. The problem of serial order in behavior. In L. A. Jeffress, editor, *Cerebral Mechanisms in Behavior*, Wiley, New York, pages 112–136. Reprinted in Saporta (1961).

Lenneberg, Eric. 1967. *The Biological Foundations of Language*. John Wiley, New York.

Lewis, David. 1970. General semantics. *Synthèse*, 22:18–67.

Lewis, Mike, Kenton Lee, and Luke Zettlemoyer. 2016. LSTM CCG parsing. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 221–231, San Diego, CA.

Lewis, Mike and Mark Steedman. 2013a. Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, 1:179–192.

Lewis, Mike and Mark Steedman. 2013b. Unsupervised induction of cross-lingual semantic relations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 681–692, Seattle, WA.

Lewis, Mike and Mark Steedman. 2014. A^* CCG parsing with a supertag-factored model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 990–1000, Doha.

Lighthill, Sir James. 1973. Artificial intelligence: A general survey. In James Lighthill, Stuart Sutherland, Roger Needham, and Christopher Longuet-Higgins, editors, *Artificial Intelligence: A Paper Symposium*, Science Research Council, London, pages 3–18.

Marslen-Wilson, William. 1973. Linguistic structure and speech shadowing at very short latencies. *Nature*, 244:522–523.

Matuszek, Cynthia, Andrzej Pronobis, Luke Zettlemoyer, and Dieter Fox. 2013. Combining world and interaction models for human–robot collaborations. In *Proceedings of Workshops at the 27th National Conference on Artificial Intelligence (AAAI)*, pages 15–22, Bellevue, WA.

McLeod, Andrew and Mark Steedman. 2016. HMM-based voice separation of MIDI performance. *Journal of New Music Research*, 45:17–26.

Mehay, Denis and Chris Brew. 2012. CCG syntactic reordering models for phrase-based machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 210–221, Montréal.

Miller, George. 1967. *The Psychology of Communication*. Basic Books, New York.

Miller, George, Eugene Galanter, and Karl Pribram. 1960. *Plans and the Structure of Behavior*. Holt, New York.

Moens, Marc and Mark Steedman. 1988. Temporal ontology and temporal reference. *Computational Linguistics*, 14:15–28. Reprinted in Inderjeet Mani, James Pustejovsky, and Robert Gaizauskas (eds.), *The Language of Time: A Reader*. Oxford University Press, pages 93–114.

Montague, Richard. 1970. Universal grammar. *Theoria*, 36:373–398. Reprinted as Thomason 1974:222–246.

Moortgat, Michael. 1997. Categorial type logics. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, North Holland, Amsterdam, pages 93–177.

Morrill, Glyn. 1994. *Type-Logical Grammar*. Kluwer, Dordrecht.

Nchare, Abdoulaye Laziz. 2012. *The Grammar of Shupamem*. Ph.D. thesis, New York University.

Niv, Michael. 1994. A psycholinguistically motivated parser for CCG. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 125–132, Las Cruces, NM.

Oehrle, Richard. 1988. Multidimensional compositional functions as a basis for grammatical analysis. In Richard Oehrle, Emmon Bach, and Deirdre Wheeler, editors, *Categorial Grammars and Natural Language Structures*. Reidel, Dordrecht, pages 349–390.

Pareschi, Remo and Mark Steedman. 1987. A lazy way to chart parse with categorial grammars. In *Proceedings of the 25th Annual Conference of the Association for Computational Linguistics*, pages 81–88, Stanford, CA.

Pierce, John, John Carroll, Eric Hamp, David Hays, Charles Hockett, Anthony Oettinger, and Alan Perlis. 1966. *Language and Machines: Computers in Translation and Linguistics*. National Academy of Sciences, National Research Council, Washington, DC. (ALPAC Report).

Pollard, Carl and Ivan Sag. 1994. *Head Driven Phrase Structure Grammar*. CSLI Publications, Stanford, CA.

Pustejovsky, James. 1998. *The Generative Lexicon*. MIT Press.

Reddy, Siva, Mirella Lapata, and Mark Steedman. 2014. Large-scale semantic parsing without question-answer pairs.

Transactions of the Association for Computational Linguistics, 2:377–392.
Ross, John Robert. 1967. *Constraints on Variables in Syntax*. Ph.D. thesis, MIT. Published as Ross 1986.
Ross, John Robert. 1986. *Infinite Syntax!* Ablex, Norton, NJ.
Saporta Sol, editor. 1961. *Psycholinguistics: A Book of Readings*. Holt Rineharts & Winston, New York.
Spärck Jones, Karen. 1964/1986. *Synonymy and Semantic Classification*. Edinburgh University Press. Ph.D. Thesis, Cambridge, 1964.
Steedman, Mark. 1987. Combinatory grammars and parasitic gaps. *Natural Language and Linguistic Theory*, 5:403–439.
Steedman, Mark. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA.
Steedman, Mark. 2002. Plans, affordances, and combinatory grammar. *Linguistics and Philosophy*, 25:723–753.
Steedman, Mark. 2018. A formal universal of natural language grammar. Submitted.
Steedman, Mark and Jason Baldridge. 2011. Combinatory categorial grammar. In Robert Borsley and Kirsti Börjars, editors, *Non-Transformational Syntax: A Guide to Current Models*. Blackwell, Oxford, pages 181–224.
Szabolcsi, Anna. 1989. Bound variables in syntax: Are there any? In Renate Bartsch, Johan van Benthem, and Peter van Emde Boas, editors, *Semantics and Contextual Expression*. Foris, Dordrecht, pages 295–318.
Thomason, Richmond, editor. 1974. *Formal Philosophy: Papers of Richard Montague*. Yale University Press, New Haven, CT.
Thorne, James, Paul Bratley, and Hamish Dewar. 1968. The syntactic analysis of English by machine. In Donald Michie, editor, *Machine Intelligence*, volume 3. Edinburgh University Press, Edinburgh, pages 281–309.
Vijay-Shanker, K., David Weir, and Aravind Joshi. 1987. Characterizing structural descriptions produced by various grammatical formalisms. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Stanford, CA.
Vinyals, Oriol, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2755–2763, Montreal.
Wexler, Kenneth and Peter Culicover. 1980. *Formal Principles of Language Acquisition*. MIT Press, Cambridge, MA.
White, Michael. 1994. *A Computational Approach to Aspectual Composition*. Ph.D. thesis, University of Pennsylvania.
Wilks, Yorick. 1975. Preference semantics. In Edward Keenan, editor, *Formal Semantics of Natural Language*. Cambridge University Press, Cambridge, pages 329–348.
Winograd, Terry. 1972. *Understanding Natural Language*. Academic Press, New York.
Woods, William. 1970. Transition network grammars for natural language analysis. *Communications of the Association for Computing Machinery*, 18:264–274.
Woods, William. 2010. The right tools: Reflections on computation and language. *Computational Linguistics*, 36:601–630.
Woods, William, Ron Kaplan, and Bonnie Nash-Webber. 1972. The lunar sciences natural language information system: Final report. Technical Report 2378, Bolt, Beranek, and Newman Inc, Cambridge, MA.
Xu, Wenduan, Stephen Clark, and Yue Zhang. 2014. Shift-reduce CCG parsing with a dependency model. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 218–227, Baltimore, MD.
Zettlemoyer, Luke and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with Probabilistic Categorial Grammars. In *Proceedings of the 21st Conference on Uncertainty in AI (UAI)*, pages 658–666, Edinburgh.