The Lost Combinator*

Mark Steedman

18 July/10 December 2018

*It linked all perplexed meanings
into one perfect peace.

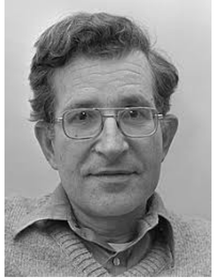
—Procter and Sullivan, 1877, *The Lost Chord*

Outline

- I: How Combinatory Categorial Grammar Grew.
- II: Why is Language Combinatory?
- III: The Future for CCG in Computational Lingustics and NLP
- IV: The Lost Combinator

In the Beginning

- In the '60s, Theoretical Linguists, Psychologists, and Computational Linguists all saw ourselves as working on the same problem, using:
  - The "transformational" theory of grammar proposed by Chomsky (1957, 1965),
  - Theories of psycholinguistics and language acquisition such as those proposed by Miller *et al.* (1960) and Miller (1967). and
  - Computational models of parsing such as those proposed by Thorne *et al.* (1968); Woods (1968).



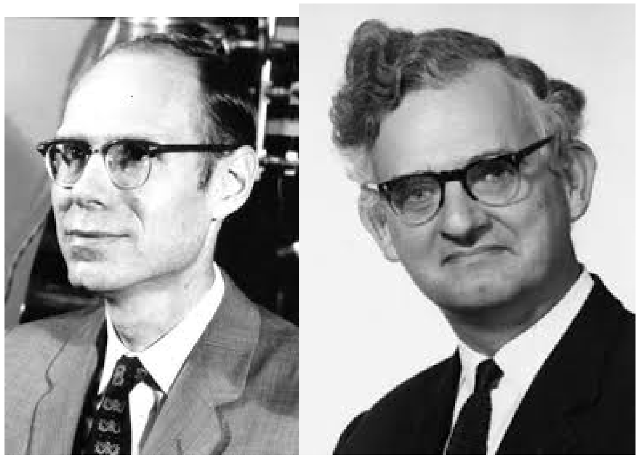
The Fall

- Soon after, this consensus fell apart.
  - Chomsky himself (1965) recognized that transformational rules, though descriptively revealing, were so expressive as to have little explanatory force, and required many apparently arbitrary constraints (Ross, 1967).
  - Psychologists realized that psycholinguistic measures of processing difficulty bore almost no relation to their transformational derivational complexity (Fodor *et al.*, 1974; Marslen-Wilson, 1973)
  - Computational linguists realized that they were spending all their time implementing exceptions to limit search (Friedman, 1971; Gross, 1978).

The Expulsion from the Garden

- And this wasn't just internal squabbling.

- There were also a couple of influential reports commissioned by the US and UK government that ended funding for Machine Translation and Artificial Intelligence (Pierce *et al.*, 1966; Lighthill, 1973).



In the Wilderness

- As a result, the field fragmented.
- Linguists swiftly abdicated any responsibility for their grammars ("Competence") to bear any relation to processing ("Performance").
- Psychologists became agnostic about grammar, retreating to context-free surface grammar or the wild romance of emergence from neural models.
- Computational linguists (whose machines were growing exponentially in size and speed to levels that would soon permit parsing the entire contents of the web) similarly found that very little of what the linguists and psycholinguists talked about was usable at scale, and that (because of Zipf's law) none of it significantly improved performance over very much simpler context-free or even finite-state methods that the linguists had shown to be incomplete..

In the Wilderness

- It also became apparent to computationalists working on speech, MT, and IR that the real problem was not grammar but ambiguity and world-knowledge, and that the solution lay in probabilistic models (Bar-Hillel, 1960/1964; Spärck Jones, 1964/1986; Jelinek and Lafferty, 1991).

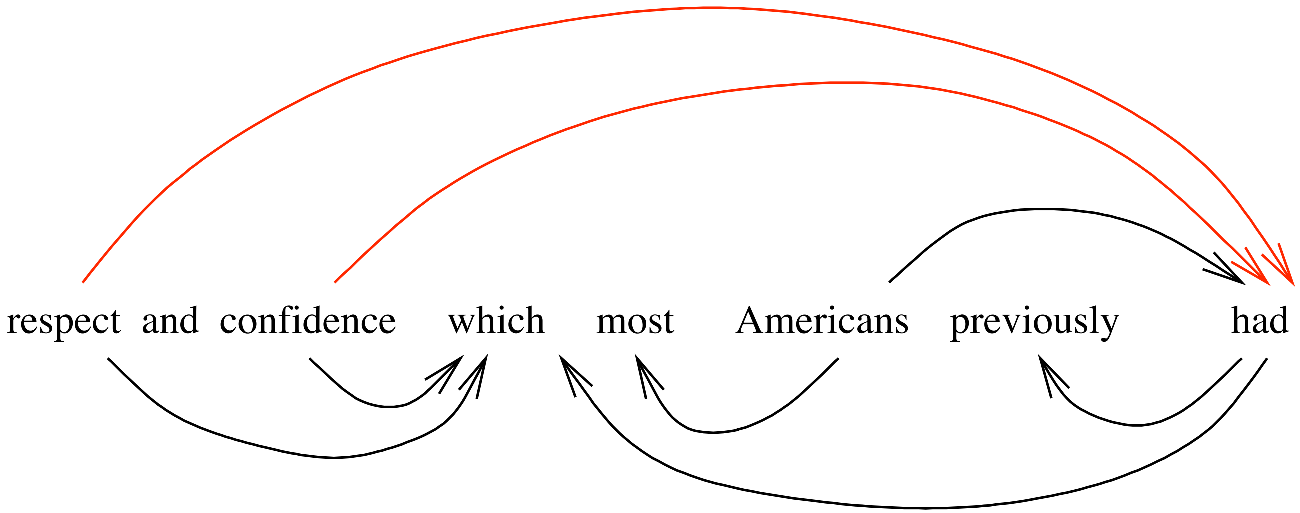
- It was not immediately apparent how to combine this insight with grammar based systems without making obviously false independence assumptions,

On the Burning Lake

- Nevertheless, as the psychologists insisted, a complete divorce between competence and performance didn't make any sense.
- They had to have evolved in lock-step, as a package deal.
- Surface syntax and the conceptual representation must also be closely related for child language acquisition to be possible (Miller, 1967; Bowerman, 1973).
- New theories of grammar were called for (Gazdar, 1981; Joshi and Levy, 1982; Ades and Steedman, 1982).

I: The Problem of Grammar

- Theoretical linguists agree that the central problem for the theory of grammar is discontinuity or non-adjacent dependency:



- Chomsky described discontinuity in terms of movement transformations, which seemed computationally to be far too unconstrained.

The Problem of Unbounded Dependency

- By contrast, the ATN parser for the LUNAR project reduced all discontinuity to local operations on registers (Thorne *et al.*, 1968; Woods, 1968)
- In particular, unbounded *wh*-dependencies like the above were handled by:
  - putting a pointer into a * or HOLD register as soon as the "Who" was encountered without regard to where it would end up, and;
  - retrieving the pointer from HOLD when the verb needing an object "disliked" was encountered without regard to where it had started out.

Generalizing Context-free Grammar

- 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 is 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).
- Ades and Steedman (1982) and Steedman (1985) suggested that the same stack could be used to characterize long-range dependency and recursion in a Combinatory Categorial Grammar (CCG).

Why Does Natural Language Allow Discontinuity?

- Natural language grammar exhibits discontinuity because language is semantically an applicative system.
- Applicative systems (such as programming languages) support the twin notions of
  - Application of a function/concept to an argument/entity.
  - Abstraction, or the definition of a new function/concept in terms of existing functions/concepts.
- Language is in that sense inherently computational.
- It seems to follow that linguistics is inherently computational.
- It does not follow that we have nothing to learn from linguistics.

Two Models for Abstraction

- There are two ways of modeling abstraction in applicative systems::

1. Taking abstraction itself as a primitive operation (λ-calculus, LISP);

   a. $\mathit{father}\,\mathit{Esau} \Rightarrow \mathit{Isaac}$

   b. $\mathit{grandfather} = \lambda x.\mathit{father}\,(\mathit{father}\,x)$

   c. $\mathit{grandfather}\,\mathit{Esau} \Rightarrow \mathit{Abraham}$

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

   b′. $\mathit{grandfather} = \mathbf{B}\,\mathit{father}\,\mathit{father}$

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

Which Model applies to Natural Language?

- Natural Language deals with all sorts of fragments that linguists don't normally think of as semantically typable constituents, without the use of variables such as pronouns:

  a. Give [Anna books]$_?$ and [Manny records]$_?$
  b. MUM (to child): There's a DOGGIE! [You LIKE]$_?$ # the doggie.
  c. Food$_i$ that you [wash$_i$ [before eating]$_i$]$_?$.
  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 (Szabolcsi, 1989; Steedman, 1987).

Combinatory Categorial Grammar (CCG)

- CCG lexicalizes all bounded dependencies, such as passive, raising, control, "exceptional case-marking", etc., via lexical logical form.
- All language-specific information such as word-order is specified in the lexicon: the syntactic rules are free and universal to all languages.
- All syntactic rules are Combinatory—that is, binarized operators over contiguous phonologically realized categories and their logical forms.
- These rules are restricted by a Combinatory Projection Principle (CPP) which essentially says they cannot override the language-specific lexicon.
- 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.

Combinatory Categorial Grammar (CCG)

- Long range dependencies are established by application of a dislocated argument such as *who(m)* $(N\backslash N)/(S/NP)$ to an adjacent non-traditional constituent S/NP, formed by rules of function composition.

- $$\begin{array}{cccc}
\text{man} & \text{who(m)} & \text{she} & \text{likes} \\
\overline{N} & \overline{(N\backslash N)/(S/NP)} & \overline{S/(S\backslash NP_{3s})}^{>\mathbf{T}} & \overline{(S\backslash NP_{3s})/NP} \\
 & & \multicolumn{2}{c}{\overline{\qquad S/NP \qquad}^{>\mathbf{B}}} \\
 & \multicolumn{3}{c}{\overline{\qquad\qquad N\backslash N \qquad\qquad}^{>}}
\end{array}$$

- $$\begin{array}{cccccc}
\text{man} & \text{who(m)} & \text{she} & \text{thinks} & \text{she} & \text{likes} \\
\overline{N} & \overline{(N\backslash N)/(S/NP)} & \overline{S/(S\backslash NP_{3s})}^{>\mathbf{T}} & \overline{(S\backslash NP_{3s})/S} & \overline{S/(S\backslash NP_{3s})}^{>\mathbf{T}} & \overline{(S\backslash NP_{3s})/NP} \\
 & & \multicolumn{2}{c}{\overline{\qquad S/S \qquad}^{>\mathbf{B}}} & \multicolumn{2}{c}{\overline{\qquad S/NP \qquad}^{>\mathbf{B}}} \\
 & & \multicolumn{4}{c}{\overline{\qquad\qquad S/NP \qquad\qquad}^{>\mathbf{B}}} \\
 & \multicolumn{5}{c}{\overline{\qquad\qquad N\backslash N \qquad\qquad}^{>}}
\end{array}$$

Second Order Combinatory Ru;es

- We also need second-order composition $\mathbf{B}^2$ to capture Germanic languages like Dutch, in which we saw serial verbs are linearized to require crossing discontinuous dependencies

  a. "Z helps Y teach X to sing."

  b. $\lambda z \lambda y \lambda x.help\,(teach\,(sing\,x)\,x\,y)\,y\,z \equiv \mathbf{B^2}\,help\,(\mathbf{B}\,teach\,sing)$

- $\mathbf{B^2}$ rules allow us to build categories of arbitrarily high valency for Dutch (and Mandarin?), such as $((S\backslash NP_z)\backslash NP_y)\backslash NP_x$.

- Thus, $\mathbf{B^2}$ rules give CCG slightly greater than context-free power.

A Combinatory Universal

- However, CCG is still not as expressive as movement tranformations.
- In particular, we can only capture separable permutations (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. *$C|D \quad A|B \quad D \quad B|C$

- This 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

The Separable Permutation Generalization

- 21 of the 22 separable permutations of “These five young boys” are attested (Cinque, 2005; Nchare, 2012).
- The two forbidden orders are among the unattested 3
- The probability of this happening by chance is

$$P = \frac{\binom{3}{2}}{\binom{24}{2}} = \frac{(3*2)/2}{(24*23)/2} = \frac{6}{552} = 0.01$$

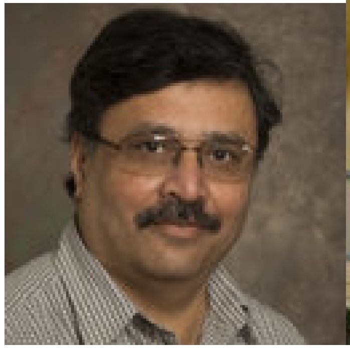
Implications of Separable Permutation for NLP

- 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.

Mild Context Sensitivity of CCG

- Vijay-Shanker and Weir (1990) proved the "shared stack" claim of Ades and Steedman (1982).

- They showed that both CCG and Aravind Joshi's Tree Adjoining Grammar (TAG) were weakly equivalent to Linear Indexed Grammar (LIG), a new level of the Language Hierarchy characterized by the (Linear) Embedded Push-down Automaton (EPDA) (cf. Kuhlmann *et al.*, 2015).



CCG for Computational Linguistics

- CCG was originally assumed to be completely hopeless for parsing, because of the extra derivational ambiguity introduced by type raising and 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 exactly the same degree of nondeterminism in the parser.
- It exists, right there in the Competence grammar
- In fact, this is just another drop in the ocean of ambiguity that faces all natural language processors, and can be handled by exactly the same statistical models as other varieties.

Statistical Models for CCG Parsers

- In particular, the Head-word dependency models pioneered by Don Hindle and Mats Rooth, Mike Collins, and Eugene Charniak are straightforwardly compatible (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 LSTM (Lewis and Steedman, 2014; Lewis *et al.*, 2016)

CCG for Natural Language Processing

- CCG is now quite widely used in applications, especially those that call for transparency between semantic and syntactic processing, and/or unbounded dependencies, such as machine translation (Birch and Osborne, 2011; Mehay and Brew, 2012), machine reading (Krishnamurthy and Mitchell, 2014), incremental parsing (Xu *et al.*, 2014; Ambati *et al.*, 2015), and semantic parser induction (Kwiatkowski *et al.*, 2010; Abend *et al.*, 2017).

Two Questions

- This modest success prompts two further questions:

1. Why should natural language be combinatory in the first place?

2. Will CCG continue to be relevant to NLP in the face of Deep Learning and Recursive Neural Networks.?

- I'll take these questions in order in parts II and III.

II: Why is Language Combinatory?

- Language is combinatory because **B**, **T**, and **S** evolved independently, before there was any language, for planning action sequences (Steedman, 2002).
- Even pure reactive planning animals like pigeons need application (Peck it!).
- Composition **B** and Substitution **S** are needed for seriation of actions. (Even rats need to compose actions to make plans. Macacques can learn 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 can plan with tools.)
- Second-order combinators like $\mathbf{B}^2$ are needed to make plans with variables, including non-present tools, such as other agents whose cooperation is yet to be obtained.
- Only humans seem to be able to do the latter.

Example: Chimpanzee and Banana

- Chimpanzees solve the misnamed monkey and bananas problem, using tools like old crates to gain altitude in order to obtain objects that are out of reach.

Figure 1: Köhler 1925

Figure 2: Köhler 1925

Example: Chimpanzee and Banana

- This amounts to composing (**B**) the affordances (**T**) of crates etc..
- But the crates etc. have to be there already.
- Apes are unable to achieve plans which involve fetching boxes from the next room, even if they have previously seen boxes there.

Example: Macacque Food-Washing

- Macacque applying $\mathbf{S}\,(\mathbf{B}\,\mathit{before}\,\mathit{eat})\,\mathit{wash}$ to a sweet potato.

Planning and Affordance

- The problem of planning can be viewed as the problem of Search for a sequence of actions in a labyrinth of possible states:

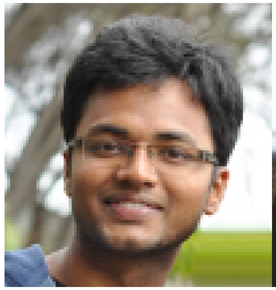
- Such search has the same recursive character as parser search
- Planning therefore also provides the infrastructure for linguistic Performance.

III: CCG in the Age of Deep Learning

- CCG parsers are therefore both linguistically and evolutionarily explanatory, and fast enough to parse the web.
- However, 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 tweaked by synthetic data and embeddings, and easy to equal on F1 using end-to-end models (Vinyals *et al.*, 2015).
- In the area of semantic parser induction for arbitrary Knowledge Graphs such as Freebase (Reddy *et al.*, 2014), CCG has already been superseded by semisupervised end-to-end training of Deep Neural Networks (DNN) (Dong and Lapata, 2016, 2018; Jia and Liang, 2016; Chen *et al.*, 2018).

Will DNNs Replace Structured Models?

- 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 use to build knowledge graphs.
- The FreeBase Query language is quite unlike linguistic logical form.
- It is probably more effective with small and idiosyncratic datasets to induce semantic parsers for them by end-to-end deep neural brute force than by CCG parser induction.




Will Parsing Go the Way of Chess and Go?

- Is it possible that the problem of parsing could be completely solved by RNN/LSTM 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?

- End-to-end SRL parsers and Neural Machine Translation (NMT) still have difficulty with long range wh-dependencies.

NMT and Subject Extraction

- 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 man who I believe (*that) won

  b. Un homme que je crois qui/*que à gagné

- This is the company that the agency told us owned the title.

  $\neq$ C'est la compagnie que l'agence nous a dit détenir le titre.

  $=$ This is the company that the agency told us to hold the title.

- C'est la compagnie que l'agence nous a dit qui détient le titre.

  $\neq$ *This is the company that the agency told us *that holds the title.

  $=$ C'est la compagnie que l'agence nous a dit qui détient le titre.

The Way Forward

- Supervised CCG parsers do rather well on embedded subject extraction (Hockenmaier and Steedman, 2002a).
- In principle, these constructions can be learned by grammar-based semantic parser induction from examples (Kwiatkowski *et al.*, 2010; Abend *et al.*, 2017).
- There is likely to be a continued need for structured representations in tasks like QA where precision on long-range dependencies matters.
- Deep Learning and distributional representations are 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.

Semantics Matters More!

- The central problem of QA is thatit involves inference as well as semantics, and we have no idea of the representation involved.
- Your Question: *Has Verizon bought Yahoo?*
- The Text:

1. Verizon purchased Yahoo. ("Yes")
2. Verizon's purchase of Yahoo ("Yes")
3. Verizon managed to buy Yahoo. ("Yes")
4. Verizon acquired every company. ("Yes")
5. Verizon doesn't own Yahoo ("No")
6. Yahoo may be sold to Verizon. ("Maybe")
7. Verizon will buy Yahoo or Yazoo. ("Maybe not")

IV: Work in Progress

- Use CCG parsers to machine-read the Web for relations between typed named entities.
- Detect consistent patterns of entailment between relations over named entities of the same types, using directional similarity over entity vectors.
- Build an entailment graph (and clean it up and close it) (Berant *et al.*, 2015).
- Cliques in the graph are 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)

Work in Progress

- Replace the original naive semantics for relation expressions with the relevant paraphrase cluster identifiers.
- Reparse the entire corpus using this form -independent semantic representation.
- Build an enormous knowledge graph, with the entities as nodes, and the paraphrase identifiers as relations.
- Throw away your other knowledge graphs!

Work in Progress

- Parse questions Q into the form-independent semantics, which is now the language of the knowledge graph itself.

- Answer questions using the knowledge graph and the entailment graph, and the following rule:
  - if Q or anything that entails Q is in the knowledge graph then answer in the positive.
  - If $\neg Q$ or the negation of anything that Q entails is in the knowledge graph, then answer in the negative.

There is Another Approach . . .

- We could do something similar with vector embeddings (Peters *et al.*, 2018).
- The machine reading problem is very similar, to the extent that we need complex expressions (including negation, auxiliaries, modals, implicative verbs etc.) to be primitive relations in theentailment graph.
- Question answering still calls for structured representation.
- We are currently pursuing a hybrid approach.

Conclusion: The Lost Combinator

- Algorithms like LSTM and Transformers may work in practice. But do they work in theory?
- Can they learn with the precision that is needed in the long tail of coordination and extraction that will be needed for Open Domain QA?
- If they aren't actually learning syntax, but are instead learning a huge FST or a Soft ATN, then by concentrating on them, we are in danger of giving up on the project of providing computational explanations of language and mind.
- Even if we believe that something like SEQ2TREE is psychological reality, and that children learn their first language that way, we still face the Kantian challenge of finding out what the target language of mind looks like.
- We can't go on using SQL and SPARQL as proxies for the language of mind.

Thanks!

- To: 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 Chambers, and Mark Johnson;
- The combinator found, Bonnie Webber;
- And all my teachers and students over many years.

References

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

Ades, Anthony and Steedman, Mark, 1982. "On the Order of Words." *Linguistics and Philosophy* 4:517–558.

Ambati, Bharat Ram, Deoskar, Tejaswini, Johnson, Mark, and Steedman, Mark, 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)*. Denver, 53–63.

Bar-Hillel, Yehoshua, 1960/1964. "A Demonstration of the NonFeasability of Fully Automatic Machine Translation." In Yehoshua Bar-Hillel (ed.), *Language and Information*, Reading, MA: Addison-Wesley. 174–179.

Berant, Jonathan, Alon, Noga, Dagan, Ido, and Goldberger, Jacob, 2015. "Efficient Global Learning of Entailment Graphs." *Computational Linguistics* 42:221–263.

Birch, Alexandra and Osborne, Miles, 2011. "Reordering Metrics for MT." In *Proceedings of the Association for Computational Linguistics*. Portland, OR: ACL, 1027–1035.

Bowerman, Melissa, 1973. "Structural Relationships in Children's Utterances: Syntactic or Semantic?" In *Cognitive Development and the Acquisition of Language*, Academic Press.

Chen, Bo, Sun, Le, and Han, Xianpei, 2018. “Sequence-to-Action: End-to-End Semantic Graph Generation for Semantic Parsing.” In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. ACL, 766–777.

Chomsky, Noam, 1957. *Syntactic Structures*. The Hague: Mouton.

Chomsky, Noam, 1965. *Aspects of the Theory of Syntax*. Cambridge, MA: MIT Press.

Cinque, Guglielmo, 2005. “Deriving Greenberg’s Universal 20 and its Exceptions.” *Linguistic Inquiry* 36:315–332.

Clark, Stephen and Curran, James R., 2004. “Parsing the WSJ using CCG and Log-Linear Models.” In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*. Barcelona: ACL, 104–111.

Dong, Li and Lapata, Mirella, 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)*. Berlin: ACL, 33–43.

Dong, Li and Lapata, Mirella, 2018. "Coarse-to-Fine Decoding for Neural Semantic Parsing." In *Proceedings of ACL*. to appear.

Fodor, Jerry A., Bever, Thomas, and Garrett, Merrill, 1974. *The Psychology of Language*. New York: McGraw-Hill.

Friedman, Joyce, 1971. *A Computer Model of Transformational Grammar*. New York: Elsevier.

Gazdar, Gerald, 1981. "Unbounded Dependencies and Coordinate Structure." *Linguistic Inquiry* 12:155–184.

Gross, Maurice, 1978. "On the Failure of Generative Grammar." *Language* 55:859–885.

He, Luheng, Lee, Kenton, Lewis, Mike, and Zettlemoyer, Luke, 2017. "Deep Semantic Role Labeling: What Works and What's Next." In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. to appear.

Hockenmaier, Julia and Steedman, Mark, 2002a. "Acquiring Compact Lexicalized Grammars from a Cleaner Treebank." In *Proceedings of the Third International Conference on Language Resources and Evaluation*. Las Palmas, Spain, 1974–1981.

Hockenmaier, Julia and Steedman, Mark, 2002b. "Generative Models for Statistical Parsing with Combinatory Categorial Grammar." In *Proceedings of* *the 40th Meeting of the Association for Computational Linguistics*. Philadelphia, 335–342.

Jelinek, Fred and Lafferty, John, 1991. "Computation of the Probability of Initial Substring Generation by Stochastic Context-Free Grammars." *Computational Linguistics* 17:315–323.

Jia, Robin and Liang, Percy, 2016. "Data Recombination for Neural Semantic Parsing." In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 12–22.

Joshi, Aravind and Levy, Leon, 1982. "Phrase Structure Trees Bear More Fruit than You Would Have Thought." *Computational Linguistics* 8:1–11.

Krishnamurthy, Jayant and Mitchell, Tom, 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). Baltimore, MD, 1188–1198.

Kuhlmann, Marco, Koller, Alexander, and Satta, Giorgio, 2015. "Lexicalization and Generative Power in CCG." *Computational Linguistics* 41:187–219.

Kuncoro, Adhiguna, Dyer, Chris, Hale, John, Yogatama, Dani, Clark, Stephen, and Blunsom, Phil, 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, 1426–1436.

Kwiatkowski, Tom, Zettlemoyer, Luke, Goldwater, Sharon, and Steedman, Mark, 2010. "Inducing Probabilistic CCG Grammars from Logical Form with Higher-Order Unification." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Cambridge, MA: ACL, 1223–1233.

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

Lewis, Mike and Steedman, Mark, 2013a. "Combined Distributional and Logical Semantics." *Transactions of the Association for Computational Linguistics* 1:179–192.

Lewis, Mike and Steedman, Mark, 2013b. "Unsupervised Induction of Cross-Lingual Semantic Relations." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. ACL, 681–692.

Lewis, Mike and Steedman, Mark, 2014. "A^* CCG Parsing with a Supertag-factored Model." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Doha, Qatar: ACL, 990–1000.

Lighthill, Sir James, 1973. "Artificial Intelligence: A General Survey." In James Lighthill, Stuart Sutherland, Roger Needham, and Christopher Longuet-Higgins (eds.), *Artificial Intelligence: A Paper Symposium*, London: Science Research Council.

Marslen-Wilson, William, 1973. "Linguistic Structure and Speech Shadowing at Very Short Latencies." *Nature* 244:522–523.

Mehay, Denis and Brew, Chris, 2012. "CCG Syntactic Reordering Models for Phrase-based Machine Translation." In *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montréal: ACL, XXX–XXX.

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

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

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

Peters, Matthew, Neumann, Mark, Iyyer, Mohit, Gardner, Matt, Clark, Christopher, Lee, Kenton, and Zettlemoyer, Luke, 2018. “Deep Contextualized Word Representations.” In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2227–2237.

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

Reddy, Siva, Lapata, Mirella, and Steedman, Mark, 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!* Norton, NJ: Ablex.

Spärck Jones, Karen, 1964/1986. *Synonymy and Semantic Classification*. Edinburgh University Press. PhD Thesis (Cambridge, 1964).

Steedman, Mark, 1985. “Dependency and Coordination in the Grammar of Dutch and English.” *Language* 61:523–568.

Steedman, Mark, 1987. "Combinatory Grammars and Parasitic Gaps." *Natural Language and Linguistic Theory* 5:403–439.

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

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* :1–35.

Szabolcsi, Anna, 1989. "Bound Variables in Syntax: Are There Any?" In Renate Bartsch, Johan van Benthem, and Peter van Emde Boas (eds.), *Semantics and Contextual Expression*, Dordrecht: Foris. 295–318.

Thorne, James, Bratley, Paul, and Dewar, Hamish, 1968. "The Syntactic Analysis of English by Machine." In Donald Michie (ed.), *Machine Intelligence*, Edinburgh: Edinburgh University Press, volume 3. 281–309.

Vijay-Shanker, K. and Weir, David, 1990. "Polynomial Time Parsing of Combinatory Categorial Grammars." In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*. Pittsburgh: ACL, 1–8.

Vinyals, Oriol, Kaiser, Łukasz, Koo, Terry, Petrov, Slav, Sutskever, Ilya, and Hinton, Geoffrey, 2015. "Grammar as a Foreign Language." In *Advances in Neural Information Processing Systems*. 2755–2763.

Woods, William, 1968. "Procedural Semantics for a Question-Answering Machine." In *Proceedings of the Fall Joint Computer Conference, part I*. ACM, 457–471.

Woods, William, Kaplan, Ron, and Nash-Webber, Bonnie, 1972. "The Lunar Sciences Natural Language Information System: Final Report." Technical Report 2378, Bolt, Beranek, and Newman Inc, Cambridge, MA.

Xu, Wenduan, Clark, Stephen, and Zhang, Yue, 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)*. Baltimore, MD, 218–227.