

---

# 3. Wide Coverage Parsing with Combinatory Grammars

Mark Steedman, University of Edinburgh

March 15th, 2017



# Prospectus

- I: Prologue: Why use CCG for NLP?
- II: Wide Coverage Parsing with CCG
- III: Building interpretations with CCG Parsers
- IV: Moral

# I: Prologue: Why Use CCG for NLP?

## Long-range dependency

- TREC 2005:

Q77.6 Name opponents who Foreman defeated.

Q77.7 Name opponents who defeated Foreman.

- A QA Program (Kor 2005):

<b>Opponents who Foreman defeated:</b>
George Foreman
Joe Frazier
Ken Norton
Sonny
Archie Moore

## Long-range Dependency

- TREC 2005 (and Google 29th Nov 2016!):

Q77.6 Name opponents who Foreman defeated.

Q77.7 Name opponents who defeated Foreman.

- A QA Program (Kor 2005):

<b>Opponents who Foreman defeated:</b>
George Foreman
Joe Frazier
Ken Norton
Sonny
Archie Moore

<b>Opponents who defeated Foreman:</b>
George Foreman
Joe Frazier
Ken Norton
Sonny
Archie Moore

## The Problem

- The **contribution of object questions** and other **long-range dependencies (LRDs)** to determining system acceptability is **disproportionate to their low frequency**.
- This is bad news.
- ❖ **Machine learning is very bad at acquiring systems for which the important information is in rare events.**

## Anatomy of a Natural Language Processor

- Every parser can be characterized by three elements:
  - A **Grammar** determined by the semantics (Regular, Context Free, Linear Indexed, etc.) and an associated automaton (Finite state, Push-Down, Extended Push-Down, etc.), together with the necessary working memories (stacks, registers, etc.);
  - A **Search Algorithm** (left-to-right etc., bottom-up etc.), etc.;
  - An **Oracle**, to resolve ambiguity and nondeterminism (lexical, structural, etc.) on some criterion (statistical, semantic, etc.).
- ◊ For some parsers (e.g. MST, McDonald *et al.*, 2005), it is sometimes hard to separate the grammar from the oracle, but it can always be done.
- The oracle can be used in two ways: either to actively limit the search space; or in the case of an all paths parser, to rank the results.

## How to Build a Parser

- Get underpaid linguistics students to **annotate 1M words of text** with Chomskian (GB) S-structures, at a cost of around \$1 per word (or just use the **Penn WSJ Treebank** Marcus *et al.*, 1993.)
- Induce a polynomial CF (or near-CF) covering **grammar** from the trees;
- Induce a probabilistic **Oracle** or parsing model from the **same** trees by counting frequencies of all the events (rules etc.) in the corpus, including **head-word dependencies** (Hindle and Rooth, 1993; Charniak, 1997; Collins, 1997; Hockenmaier and Steedman, 2002b).

(1) I saw<sub>*i*?</sub> the squirrel<sub>*j*?</sub> with a telescope<sub>*i/j*?</sub>.

- Parse with your favorite **Algorithm**, such as CKY, using the model to limit search.



## II: Wide-Coverage Parsing with CCG

## CCG is Nearly Context-Free

- It has polynomial parsing complexity (Vijay-Shanker and Weir 1990)
- Hence it has nice “Divide and Conquer” algorithms, like CKY, and Dynamic Programming.

## CKY Algorithm (adapted from Harrison 1978)

```
(2) 1. for  $j := 1$  to  $n$  do
      begin
         $t(j, j) := \{A \mid A \text{ is a lexical category for } a_j\}$ 
      2. for  $i := j - 1$  down to  $0$  do
          begin
            3. for  $k := i$  down to  $0$  do
                begin
                   $t(k, j) := \text{pack}\{A \mid \text{for all } B \in t(k, i), C \in t(i + 1, j) \text{ end}$ 
                    such that  $B C \Rightarrow A$  for some
                    combinatory rule in  $R$ 
                    and  $\text{admissible}(B C \Rightarrow A)\}$ 
                end
            end
          end
        end
      end
```

## Nearly Context-Free Grammar

- Such Grammars capture the deep dependencies associated with coordination and long range dependency.
- Both phenomena are frequent in corpora, and are **explicitly annotated in the Penn WSJ corpus**.
- Standard treebank grammars ignore this information and fail to capture these phenomena entirely.
- ◇ Zipf's law says capturing them won't give us much better overall numbers. (around 3% of sentences in WSJ include long-range object dependencies, and LRODs are only a small proportion of the dependencies in those sentences.)
- **But** there is a big difference between getting a perfect eval-b score on a sentence including an object relative clause and interpreting it!

## LRDs Really Are Out There in the Treebank

- Full Object Relatives ( 570 in WSJ treebank)
- Reduced Object Relatives ( 1070 in WSJ treebank)
- Argument Cluster Coordination ( 230 in WSJ treebank):

```
(S (NP-SBJ It)
  (VP (MD could)
    (VP (VP (VB cost)
      (NP-1 taxpayers)
      (NP-2 $ 15 million))
      (CC and)
      (VP (NP=1 BPC residents)
        (NP=2 $ 1 million))))))
```

- It could cost **taxpayers 15 million and \_\_ BPC residents 1 million**

## LRDs Really Are Out There in the Treebank

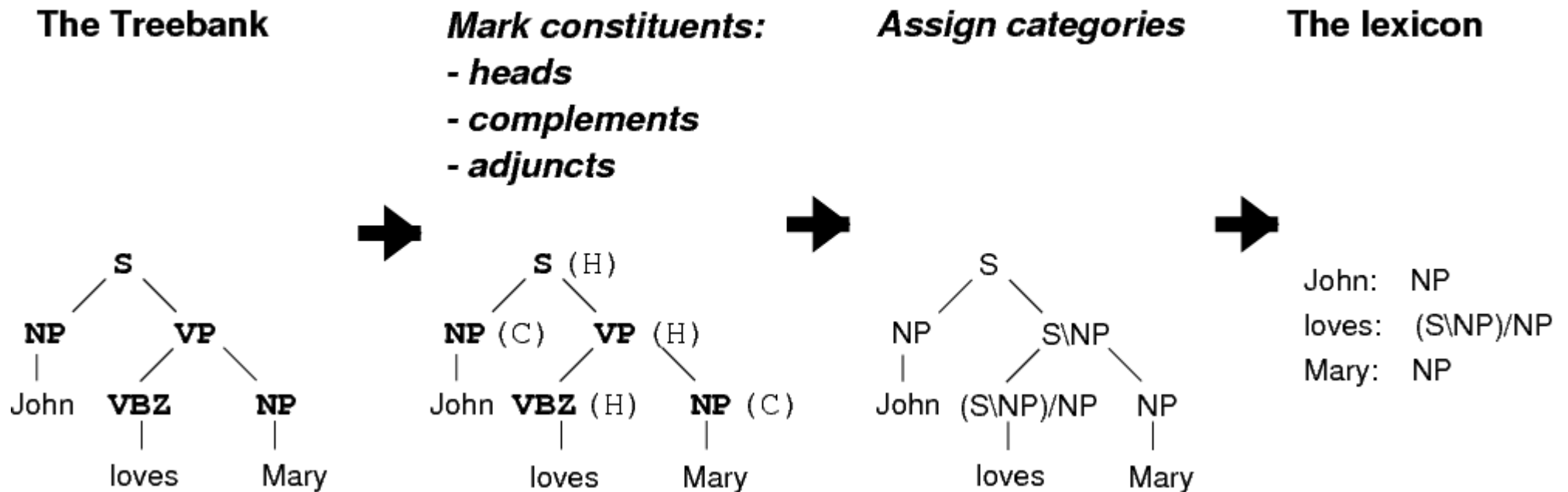
- Parasitic Gaps (at least 6 in WSJ treebank):

```
(S (NP-SBJ Hong Kong's uneasy relationship with China)
  (VP (MD will)
    (VP (VP (VB constrain)
      (NP (-NONE- *RNR*-1)))
      (PRN (: --)
        (IN though)
        (VP (RB not)
          (VB inhibit)
          (NP (-NONE- *RNR*-1)))
          (: --))
        (NP-1 long-term economic growth))))))
```

- HK's relation with C will **constrain** \_, **though not inhibit** \_, **long-term growth**.

# Supervised CCG Induction by Machine

- Extract a CCG lexicon from the Penn Treebank: Hockenmaier and Steedman (2002a), Hockenmaier (2003) (cf. Buszkowski and Penn 1990; Xia 1999).



## Supervised CCG Induction by Machine

- CCGbank trades lexical types (>500 against 48) for rules (around 3000 instantiated binary combinatory rule types against around 12000 PS rule types) with standard Treebank grammars.
- ◊ The trees in the CCG-bank are CCG derivations, and in cases like Argument Cluster Coordination and Relativisation they depart radically from Penn Treebank structures.
- Otherwise, CCGbank trees are “Right Normal-form”, where the derivations modeled are those in which type-raising and composition are only used when there is no alternative.



## Supervised CCG Induction: Full Algorithm

- foreach tree T:  
  preprocessTree(T);  
  preprocessArgumentCluster(T);  
  determineConstituentType(T);  
  makeBinary(T);  
  percolateTraces(T);  
  assignCategories(T);  
  treatArgumentClusters(T);  
  cutTracesAndUnaryRules(T);
- The resulting treebank is somewhat cleaner and more consistent, and is offered for use in inducing grammars in other expressive formalisms. It was **released in June 2005 by the Linguistic Data Consortium** with documentation and can be searched using t-grep.

## Translating Dependency Treebanks to CCG

- Penn-style treebanks are much rarer than dependency treebanks, because of **widespread crossing dependencies** in languages other than English.
- Dependency banks can be similarly translated by walking the dependency graph from the root, using dependency labels to distinguish arguments (SUBJ, OBJ, etc.) from adjuncts (TEMP, LOC, etc.), and using alignment to determine directionality.
- This has been done successfully for **Turkish** (Çakıcı, 2005, 2009) and **Hindi** (Ambati *et al.*, 2013, 2016a).
- This is greatly facilitated by the growing use of **Universal Dependencies** (McDonald *et al.*, 2013).

# Statistical Models for Wide-Coverage Parsers

- There are two kinds of statistical models:
  - **Generative** models directly represent the **probabilities of the rules of the grammar**, such as the probability of the word **eat** being transitive, or of it taking a nounphrase headed by the word **integer** as object.
  - **Discriminative** models compute values for whole parses as a function of the product of a number of **weighted features**, like a Perceptron. These features typically include those of generative models, but can be anything.
- Both are estimated from **counts of corresponding events in the treebank**
- Both have been applied to CCG parsing

## Generative Head-Word Dependency Model

- Hockenmaier and Steedman (2002b):

Model	LexCat	Parseval				Surface dependencies	
		LP	LR	BP	BR	$\langle PHS \rangle$	$\langle \rangle$
Baseline	87.7	72.8	72.4	78.3	77.9	81.1	84.3
HWDep	92.0	81.6	81.9	85.5	85.9	84.0	90.1

- Collins (1999) reports 90.9% for unlabeled  $\langle \rangle$  “surface” dependencies.
- **CCG benefits greatly from word-word dependencies.**  
(in contrast to Gildea (2001)’s observations for Collins’ Model 1)
- This parser is available on the project webpage.

## Long Range Dependencies (Hockenmaier, 2003)

- Extraction:

- Dependencies involving **subject relative pronoun**  
( $\text{NP} \setminus \text{NP}$ ) / ( $\text{S}[\text{dcl}] \setminus \text{NP}$ ): 98.5%LP, 95.4%LR (99.6%UP, 98.2%UR)
- Lexical cat. for **embedded subject extraction** (Steedman, 1996)  
(( $\text{S}[\text{dcl}] \setminus \text{NP}$ ) /  $\text{NP}$ ) / ( $\text{S}[\text{dcl}] \setminus \text{NP}$ ): 100.0%P, 83.3%R
- Dependencies involving **object relative pronoun (including ES)**  
( $\text{NP} \setminus \text{NP}$ ) / ( $\text{S}[\text{dcl}] / \text{NP}$ ): 66.7%LP, 58.3%LR (76.2%UP, 58.3%UR)

- Coordination:

- VP coordination (coordination of  $S[.] \setminus \text{NP}$ ): 67.3%P, 67.0%R
- Right-node-raising (coordination of  $(S[.] \setminus \text{NP}) / \text{NP}$ ): 73.1%P, 79.2%R

## Log-Linear Conditional CCG Parsing Models

- Features  $f_i$  encode evidence indicating good/bad parses
- (3)  $p(d|S) = \frac{1}{Z(S)} e^{\sum_i \lambda_i f_i(d,S)}$
- Use standard Maximum Entropy techniques to train a FSM “supertagger” Clark (2002) to assign CCG categories, **multitagging** ( $n \approx 3$ ) **at over 98% accuracy** (Clark and Curran 2003, 2004).
- Clark and Curran use a conditional log-linear model such as Maximum Entropy of **either**:
  - The derived structure or parse yield;
  - All derivations;
  - All derivations with Eisner Normal Form constraints.

## Conditional CCG Parsing Models (Contd.)

- Discriminative estimation via the limited-memory BFGS algorithm is used to set feature weights
- Estimation is computationally expensive, particularly for “all derivations”:
  - A cluster was used to allow complete Penn Treebank to be used for estimation.
  - The fact that the supertagger is very accurate makes this possible.

## Overall Dependency Recovery

	LP	LR	UP	UR	cat
Clark et al. 2002	81.9	81.8	90.1	89.9	90.3
Hockenmaier 2003	84.3	84.6	91.8	92.2	92.2
<b>Clark and Curran 2004</b>	<b>86.6</b>	<b>86.3</b>	<b>92.5</b>	<b>92.1</b>	<b>93.6</b>
Hockenmaier (POS)	83.1	83.5	91.1	91.5	91.5
<b>C&amp;C (pos)</b>	<b>84.8</b>	<b>84.5</b>	<b>91.4</b>	<b>91.0</b>	<b>92.5</b>

Table 1: Dependency evaluation on Section 00 of the Penn Treebank

- To maintain comparability to Collins, Hockenmaier (2003) did not use a Supertagger, and was forced to use beam-search.

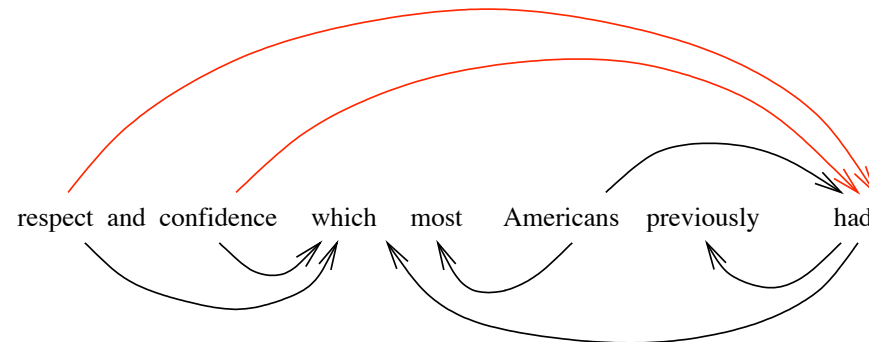


## Log-Linear Overall Dependency Recovery

- The C&C parser delivered **state-of-the-art dependency recovery**.
- The C&C parser was **very fast** ( $\approx$  30 sentences per second)
- **The speed comes from highly accurate "supertagging"**, used in an aggressive **"Best-First increasing adaptive"** mode (Clark and Curran 2004), and behaved as an "almost parser" (Bangalore and Joshi 1999)
- ◊ Clark and Curran 2006 show that CCG all-paths almost-parsing with supertagger-assigned categories loses only 1.3% dependency-recovery F-score against parsing with a full dependency model (!)

# Recovering Long-Range Dependencies

Clark *et al.* (2004)



lexical_item	category	slot	head_of_arg
<i>which</i>	$(NP_X \setminus NP_{X,1}) / (S[dcl]_2 / NP_X)$	2	<i>had</i>
<i>which</i>	$(NP_X \setminus NP_{X,1}) / (S[dcl]_2 / NP_X)$	1	<i>confidence</i>
<i>which</i>	$(NP_X \setminus NP_{X,1}) / (S[dcl]_2 / NP_X)$	1	<i>respect</i>
<i>had</i>	$(S[dcl]_{had} \setminus NP_1) / NP_2$	2	<i>confidence</i>
<i>had</i>	$(S[dcl]_{had} \setminus NP_1) / NP_2$	2	<i>respect</i>

## Full Object Relatives in Section 00

- 431 sentences in WSJ 2-21, 20 sentences (24 object dependencies) in Section 00.
  1. Commonwealth Edison now faces an additional court-ordered **refund** on its summerwinter rate differential collections **that** the Illinois Appellate Court has **estimated** at DOLLARS.
  2. Mrs. Hills said many of the 25 **countries that** she **placed** under varying degrees of scrutiny have made genuine progress on this touchy issue.
  - ✓ 3. It's the petulant complaint of an impudent **American whom** Sony **hosted** for a year while he was on a Luce Fellowship in Tokyo – to the regret of both parties.
  - ✓ 4. It said the **man, whom** it did not **name**, had been found to have the disease after hospital tests.
  5. Democratic Lt. Gov. Douglas Wilder opened his gubernatorial battle with Republican Marshall Coleman with an abortion **commercial** produced by Frank Greer **that** analysts of every political persuasion **agree** was a tour de force.
  6. Against a shot of Monticello superimposed on an American flag, an announcer talks about the strong **tradition** of freedom and individual liberty **that** Virginians have **nurtured** for generations.
  - ✓ 7. Interviews with analysts and business people in the U.S. suggest that Japanese capital may produce the economic **cooperation that** Southeast Asian politicians have **pursued** in fits and starts for decades.
  8. Another was Nancy Yeargin, who came to Greenville in 1985, full of the **energy** and **ambitions that** reformers wanted to **reward**.
  9. Mostly, she says, she wanted to prevent the **damage** to self-esteem **that** her low-ability students would **suffer** from doing badly on the test.
  - ✓ 10. Mrs. Ward says that when the cheating was discovered, she wanted to avoid the morale-damaging public **disclosure that** a trial would **bring**.

- ✓ 11. In CAT sections where students' knowledge of two-letter consonant sounds is tested, the authors noted that Scoring High concentrated on the same **sounds that** the test **does** – to the exclusion of other **sounds that** fifth graders should **know**.
  - ✓ 12. Interpublic Group said its television programming **operations** – **which** it **expanded** earlier this year – agreed to supply more than 4,000 hours of original programming across Europe in 1990.
  - 13. Interpublic is providing the programming in return for advertising **time**, **which** it **said** will be valued at more than DOLLARS in 1990 and DOLLARS in 1991.
  - ✓ 14. Mr. Sherwood speculated that the **leeway that** Sea Containers **has** means that Temple would have to substantially increase their bid if they're going to top us.
  - ✓ 15. The Japanese companies bankroll many small U.S. companies with promising products or ideas, frequently putting their money behind **projects that** commercial banks won't **touch**.
  - ✓ 16. In investing on the basis of future transactions, a role often performed by merchant banks, trading companies can cut through the **logjam that** small-company owners often **face** with their local commercial banks.
  - 17. A high-balance **customer that** banks **pine for**, she didn't give much thought to the rates she was receiving, nor to the fees she was paying.
  - ✓ 18. The events of April through June damaged the **respect** and **confidence which** most Americans previously **had** for the leaders of China.
  - ✓ 19. He described the situation as an escrow **problem**, a timing **issue**, **which** he **said** was rapidly rectified, with no losses to customers.
  - ✓ 20. But Rep. Marge Roukema (R., N.J.) instead praised the House's acceptance of a new youth training wage, a **subminimum that** GOP administrations have **sought** for many years.
- **Cases of object extraction from a relative clause in 00 associated with the object relative pronoun category  $(NP_X \setminus NP_X) / (S[dcl] / NP_X)$ ;**
  - **The extracted object, relative pronoun and verb are in italics; sentences marked with a ✓ are cases where the parser correctly recovers all object dependencies**

## Clark *et al.* (2004): Full Object Relatives

- 24 cases of extracted object in Section 00 associated with object relative pronoun category  $(NP_X \setminus NP_X) / (S[dcl] / NP_X)$
- 15/24 (62.5%) recovered with all dependencies correct (15/20 (75%) precision)
  - That is, with both noun verb dependency correct—cf. 58.3%/67% labelled recall/precision by Hockenmaier.
  - 1 sentence (1) failed to parse at all (necessary category for seen verb **estimated** unseen in 2-21).
  - 5 were incorrect because wrong category assigned to relative pronoun, of which: in two (5, 9) this was only because again the necessary category for a seen verb was unseen in 2-21, and one (17) was incorrect because the POS tagger used for back-off labeled the entirely unseen verb incorrectly
  - 3 incorrect only because relative clause attached to the wrong noun

## Clark *et al.* (2004): Free Relatives

- 14/17 (82%) recall 14/15 (93%) precision for the single dependency.
  - Better performance on long-range dependencies can be expected with more features such as regular expressions for Max Ent to work on.
  - Other varieties of deep dependency (Control, subject relatives, reduced relatives) discussed in Hockenmaier (2003); Clark *et al.* (2002, 2004).
  - It looks as though about half the errors arise because the lexicon is too small, and about half because the head-dependency model is too weak.
- ◇ 1M words of treebank is nothing like enough data

## Experiments with Porting the Parser

- As with all treebank grammars, almost any practical application involves porting the parser to a different grammar and model.
- For example, in ongoing experiments with open domain question answering, we would like to use the parser for parsing the questions.
- However, all treebank grammars including this one do appallingly badly on the TREC question database, because WSJ contains almost no direct questions, and none at all of some common patterns.
- Hand-labelling data for retraining is usually not possible.
- However, semi-automatically hand-supertagging a few thousand sentences and retraining the supertagger with those included is quite practical.
- We did the 1,171 **What** questions from TREC in a week

## Porting to Questions: Results

- 171 **What**-question development set. 1000 for training (and testing using tenfold cross-validation), average length 8.6 words.
- Since the gold standard question data is only labelled to the level of lexical category we can only evaluate to that level.
- However, supertagger accuracy and sentence accuracy correlate very highly with dependency and category recall by the parser, and we know we need around 97% per word and 60% per sentence for the original WSJ performance

MODEL	1 CAT ACC	SENT ACC	1.5 cats /word	SENT ACC
• CCGbank	72.0	1.8	84.8	11.1
Qs	92.3	66.7	96.6	80.7
10×Qs+CCGbank	93.6	66.7	97.9	83.0

Table 2: Accuracy of Supertagger on Development set Question Data



## Porting to Questions: Results

- For the **What** object questions, per word/sentence accuracies were 90%/71%, suggesting that they are harder than the average question.
- Object dependency recall by the parser for these questions was 78%.

## On Supertagging

- What the above experiments are telling us is the **centrality of supertagging** to practical CCG parsing.
- Supertagging seems to work **better for CCG than other grammars**, including TAG.
- This seems to be because CCG has **many lexical category types**, but **few rule types** (around 10 combinatory rules, plus 13 unary rules).
- This means that **apart from the lexicon**, the grammar is small enough to be handcoded.
- This in turn will allow us **confine the entire statistical model to the lexicon and the supertagger**.

## Modern CCG Parsing

- The next slides show:
  - That supertagging can be improved using word-embeddings trained on Bns of words of unlabelled text as features in a CRF sequence model, allowing elimination of POS tags and improved domain transfer.
  - That the improved supertagger is so efficient that the residual parsing problem can be solved by exhaustive search over the entire supertag distributions, without an adaptive beam, using the A\* algorithm (Klein and Manning, 2003; Auli and Lopez, 2011), with an order of magnitude increase in speed and improved transfer.
  - That A\* parsing works even better with LSTM.

## Supertagging with Word-embeddings

- Lewis and Steedman (2014b) show that for just about any of the available pretrained embeddings (Collobert et al., Mikolov, Turian et al., Mnih and Hinton, *passim*, and a variety of neural network language models (Windows Approach Network, CRF), using embeddings as features improved supertagging accuracy.
- The best results were with 50 dimensional Turian embeddings and CRF (Turian *et al.*, 2010).
- Extrinsic evaluation was against the C&C parser using various taggers (C&C, Honnibal and Curran (2009)) testing on WSJ CCGbank, Wikipedia and Bioinfer, with around **1% improvement on WSJ, and around 2% on out-of-domain**.
- Xu *et al.* (2015) show that an RNN works even better.

## Supertagging with Word-embeddings

- Error analysis showed the improvement came from **eliminating automatic POS tags**, and for **unseen word-category pairs**, where the embeddings add the most information.
- However, Clark and Curran **lose 20% of the correct parses from the Best-first Increasing Adaptive beam**.<sup>1</sup>

---

<sup>1</sup>Low impact on F-score reflects the fact that the full parsing model is extremely weak.

## A<sup>\*</sup> CCG Parsing

- A<sup>\*</sup> search is a variety of branch-and-bound algorithm in which the search space is dynamically ordered by the use of an **estimator**, which assigns an upper bound to the value of a continuation, so that choices can be ordered and search may be best-first.
- Klein and Manning (2003) showed that such estimators exist for probabilistic parsing models. Auli and Lopez (2011) showed that A<sup>\*</sup> could be applied to CCG parsing and was speeded by supertagging.
- Lewis and Steedman (2014a) showed that **a better estimator** could be obtained by factorization of the parsing model as a unigram category model alone, allowing an upper bound for a partial parse to be estimated from its **inside probability and the highest probability categories for the remaining words**.
- Since supertagging is over 90% accurate, **the upper-bound will often be exact**.

## A<sup>\*</sup> CCG Parsing

- Intuitively, the parser begins by assigning all possible categories to all words ranked by probability, then finds the highest probability (Viterbi) parse by the following algorithm:
  - It first tries to find a parse over a graph consisting of the highest probability categories for all words;
  - If it fails then it adds the highest probability category among the next highest probability categories for all words to the graph;
  - Repeating until it finds a parse.
- This algorithm allows the parser to consider an unbounded number of categories for each word, without building a complete chart for all.
- Accuracy is competitive with C&C and **an order of magnitude faster**.

## LSTM CCG Parsing

- Because the parsing model is now confined to the supertagger and can be defined entirely in terms of matrix operations it is a suitable case for treatment using LSTM with generic library code
- Lewis *et al.* (2016) use two stacked LSTMs and improve slightly on Xu *et al.* (2015) in terms of accuracy.
- This is all **without using a head dependency model**.
- Lewis *et al.* achieve **a further order of magnitude in terms of speed** by running the parser on a GPU.
- They also show a further improvement from **“tri-training”**



## Shift-Reduce Parsing

- Neural Network-based Supertag-factored modeling also suggests itself for non- $A^*$  CCG parsing, such as the Shift-reduce parser proposed in Ades and Steedman, 1982; Steedman, 2000 and investigated by Xu *et al.* (2014) and Ambati *et al.* (2016b).

## III: Building Interpretations with CCG Parsers

## Building Interpretations with CCG Parsers

- The interpretation of the combinatory rules as type raising and composition guarantees “surface compositionality” with **any** compositional semantic representation.
- This in turn means that the process of interpretation building can be built into the categories and combinatory rules, and can be done in parallel to derivation, as discussed last week
- To make such a semantics wide-coverage involves specifying a semantics or a morphological stem-based semantic schema for the 400-500 most frequent category types (Hockenmaier *et al.* 2004; Bos *et al.* 2004)
- We use first order logics such as FOPL or DRT, (or the language of the Google Knowledge graph Reddy *et al.* (2014)), using the lambda calculus as a “glue language”.

```

-----
| x1          | | x2 x3      | |
|-----| |-----| |
(| company(x1) |A| say(x2)      | |
| single(x1)  | | agent(x2,x1) | | | | | |
|-----| | theme(x2,x3) | |
|            | | proposition(x3) | |
|            | |
|            | | x4          | | x5          | | x6 x7 x8    | |
| x3: |-----| |-----| |-----| |
|      (| card(x4)=billion |;(| filter(x5) |A| with(x4,x5)   |)) |
|      | 9.8(x4)          | | plural(x5) | | sell(x6)      | |
|      | kent(x4)         | |-----| | patient(x6,x4) | |
|      | cigarette(x4)   | |            | | 1953(x7)     | |
|      | plural(x4)      | |            | | single(x7)    | |
|      |-----| |            | | 1955(x8)     | |
|            | |            | | single(x8)    | |
|            | |            | | to(x7,x8)     | |
|            | |            | | from(x6,x7)   | |
|            | |            | | event(x6)     | |
|            | |            | |-----| |
|            | |            | |
| event(x2)  | |            | |
|-----| |-----| |

```

## The Poverty of Logicism

- Parsing with C&C 2004, and feeding such logical forms to a battery of FOL theorem provers, Bos and Markert (2005) attained quite high precision of 76% on the 2nd PASCAL RTE Challenge Problems.
- ◇ However, recall was only 4% ;@(
  - MacCartney and Manning (2007) argue that entailment must be computed much more directly, from the surface form of sentences, or from the strings themselves.
  - ◇ Rather, we need a quite different semantics, **less tied to linguistic form**.
  - The answer to your question is out there in the web, but **probably not in the form (or even the language) of the question itself**.

## Moral

- You can have the linguistic expressivity that is needed to build interpretable structure **and** parse efficiently with wide coverage—with an automatically induced CCG lexicon and a supertagging language model—
- —But we **need a better semantics!**

## References

- Ades, Anthony and Steedman, Mark, 1982. “On the Order of Words.” *Linguistics and Philosophy* 4:517–558.
- Ambati, Bharat Ram, Deoskar, Tejaswini, and Steedman, Mark, 2013. “Using CCG Categories to Improve Hindi Dependency Parsing.” In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia, 604–609.
- Ambati, Bharat Ram, Deoskar, Tejaswini, and Steedman, Mark, 2016a. “Hindi CCGbank: CCG Treebank from the Hindi Dependency Treebank.” *Language Resources and Evaluation* 51:1–34.

Ambati, Bharat Ram, Deoskar, Tejaswini, and Steedman, Mark, 2016b. “Shift-Reduce CCG Parsing using Neural Network Models.” In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. San Diego, CA, 447–453.

Auli, Michael and Lopez, Adam, 2011. “Efficient CCG Parsing: A\* versus Adaptive Supertagging.” In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, OR: ACL, 1577–1585.

Bangalore, Srinivas and Joshi, Aravind, 1999. “Supertagging: An Approach to Almost Parsing.” *Computational Linguistics* 25:237–265.

Bos, Johan and Markert, Katja, 2005. “Combining Shallow and Deep NLP Methods for Recognizing Textual Entailment.” In *Proceedings of*



---

*the First PASCAL Challenge Workshop on Recognizing Textual Entailment.*  
<http://www.pascal-network.org/Challenges/RTE/>: Pascal, 65–68.

Buszkowski, Wojciech and Penn, Gerald, 1990. “Categorial Grammars Determined from Linguistic Data by Unification.” *Studia Logica* 49:431–454.

Çakıcı, Ruket, 2005. “Automatic Induction of a CCG grammar for Turkish.” In *Proceedings of the Student Workshop, 43rd Annual Meeting of the ACL, Ann Arbor MI*. ACL, 73–78.

Çakıcı, Ruket, 2009. *Parser Models for a Highly Inflected Language*. Ph.D. thesis, University of Edinburgh.

Charniak, Eugene, 1997. “Statistical Parsing with a Context-free Grammar and Word Statistics.” In *Proceedings of the 14th National Conference of the American Association for Artificial Intelligence, Providence, RI., July*. 598–603.

- Clark, Stephen, 2002. “A Supertagger for Combinatory Categorical Grammar.” In *Proceedings of the TAG+ Workshop*. Venice, 19–24.
- Clark, Stephen and Curran, James R., 2003. “Log-Linear Models for Wide-Coverage CCG Parsing.” In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Sapporo, Japan, 97–104.
- 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.
- Clark, Stephen and Curran, James R., 2006. “Partial Training for a Lexicalized Grammar Parser.” In *Proceedings of the Human Language Technology Conference and Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL '06)*. New York: ACL.

Clark, Stephen, Hockenmaier, Julia, and Steedman, Mark, 2002. “Building Deep Dependency Structures with a Wide-Coverage CCG Parser.” In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, 327–334.

Clark, Stephen, Steedman, Mark, and Curran, James R., 2004. “Object-Extraction and Question-Parsing Using CCG.” In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Barcelona: ACL, 111–118.

Collins, Michael, 1997. “Three Generative Lexicalized Models for Statistical Parsing.” In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*. Madrid: ACL, 16–23.

Collins, Michael, 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

---

Gildea, Dan, 2001. “Corpus Variation and Parser Performance.” In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*. Pittsburgh, PA, 167–202.

Harrison, Michael, 1978. *Introduction to Formal Language Theory*. Reading MA: Addison-Wesley.

Hindle, Donald and Rooth, Mats, 1993. “Structural Ambiguity and Lexical Relations.” *Computational Linguistics* 19:103–120.

Hockenmaier, Julia, 2003. *Data and Models for Statistical Parsing with CCG*. Ph.D. thesis, School of Informatics, University of Edinburgh.

Hockenmaier, Julia, Bierner, Gann, and Baldridge, Jason, 2004. “Extending the Coverage of a CCG System.” *Journal of Logic and Computation* 2:165–208.

- 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 Categorical Grammar.” In *Proceedings of the 40th Meeting of the Association for Computational Linguistics*. Philadelphia, 335–342.
- Honnibal, Matthew and Curran, James, 2009. “Fully Lexicalising CCGbank with Hat Categories.” In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. ACL, 1212–1221.
- Klein, Dan and Manning, Christopher D., 2003. “A\* Parsing: Fast Exact Viterbi Parse Selection.” In *Proceedings of Human Language Technology/North*

---

*American Chapter of the Association for Computational Linguistics*. Edmonton, Alberta: ACL, 40–47.

Kor, Kian Wei, 2005. *Improving Answer Precision and Recall of List Questions*. Ph.D. thesis, Edinburgh.

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, 2014a. “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.

Lewis, Mike and Steedman, Mark, 2014b. “Improved CCG Parsing with Semi-Supervised Supertagging.” *Transactions of the Association for Computational Linguistics* 2:327–338.

MacCartney, Bill and Manning, Christopher D., 2007. “Natural Logic for Textual Inference.” In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Prague: ACL, 193–200.

Marcus, Mitch, Santorini, Beatrice, and Marcinkiewicz, M., 1993. “Building a Large Annotated Corpus of English: The Penn Treebank.” *Computational Linguistics* 19:313–330.

McDonald, Ryan, Nivre, Joakim, Quirnbach-Brundage, Yvonne, Goldberg, Yoav, Das, Dipanjan, Ganchev, Kuzman, Hall, Keith, Petrov, Slav, Zhang, Hao, Täckström, Oscar, Bedini, Claudia, Bertomeu Castelló, Núria, and Lee,

Jungmee, 2013. “Universal Dependency Annotation for Multilingual Parsing.” In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Sofia: ACL, 92–97.

McDonald, Ryan, Pereira, Fernando, Ribarov, Kiril, and Hajic, Jan, 2005. “Non-Projective Dependency Parsing using Spanning Tree Algorithms.” In *Proceedings of the Joint Conference on Human Language Technologies and Empirical Methods in Natural Language Processing HLT/EMNLP*,. New Brunswick, NJ: ACL, 523–530.

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.

Steedman, Mark, 1996. *Surface Structure and Interpretation*. Linguistic Inquiry Monograph 30. Cambridge, MA: MIT Press.



---

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

Turian, Joseph, Ratinov, Lev-Arie, and Bengio, Yoshua, 2010. “Word Representations: A Simple and General Method for Semi-Supervised Learning.” In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL, 384–394.

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

Xia, Fei, 1999. “Extracting Tree-Adjoining Grammars from Bracketed Corpora.” In *Proceedings of the 5th Natural Language Processing Pacific Rim Symposium (NLPRS-99)*.

Xu, Wenduan, Auli, Michael, and Clark, Stephen, 2015. “CCG Supertagging with a Recurrent Neural Network.” In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing: ACL, 250–255.

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.