Two Early Efforts toward Using Deep Learning in Syntax and Semantics

Sam Bowman
NYU

CLASP Seminar, University of Gothenburg

Context: Deep learning in NLP

As in vision and elsewhere, deep learning techniques have yielded very fast progress on a few important data-rich tasks:

- **Reading comprehension questions**
  - Near human performance (but brittle)
- **Translation**
  - Large, perceptually obvious improvements
- **Syntactic parsing**
  - Measurable improvements on longstanding state of the art

The Question

Given that these models incorporate no substantial prior knowledge about language, what can their (partial) successes tell us about language?

The Question

Given that these models incorporate no substantial prior knowledge about language, what can their (partial) successes tell us about language?

Today: Two attempts at answering this question.

- **Part I:** Discovering tree structure
- **Part II:** Learning to match expert acceptability judgments

Part I

Learning to Parse
from a Semantic Objective

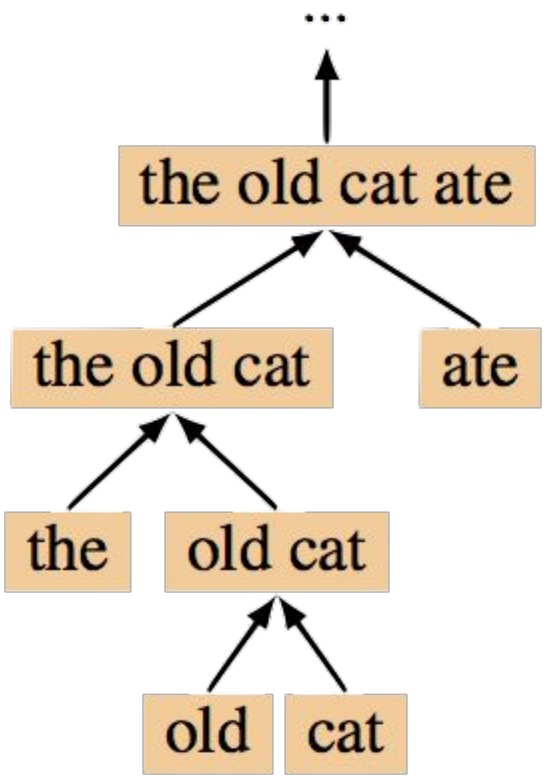
Adina Williams
Andrew Drozdov
Samuel R. Bowman

TACL 2018 (@NAACL)

Nikita Nangia
Samuel R. Bowman

Under Submission

Background: TreeRNNs



What?

- Run a (binary constituency) parser
- Use parse tree as computation graph
  - Generally with *TreeLSTM* function at each node

Why?

- Theoretically appealing
- Some empirical advantage

Socher et al. ‘10, Tai et al. ‘15, Bowman et al. ‘16, Liu et al. ‘17

Goal: *Latent Tree Learning*

What?

- Build one model that can:
  - Parse sentences
  - Use resulting parses in a TreeRNN text classifier
- Train the full model on a language understanding task

Why?

- Engineering objective:

  Better parsing strategies for NLU
- Scientific objective:

  What compositional structures are both valuable and learnable?

Goal: *Latent Tree Learning*

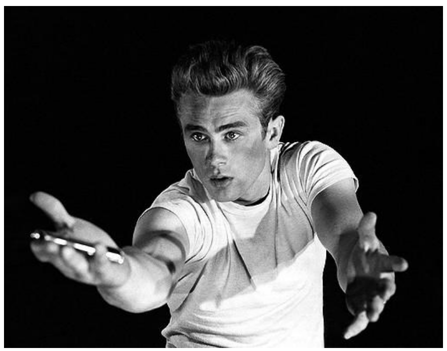
Today:

- What do *existing* methods for this task learn?
  - New evaluations, red flags, and negative results.
- Is the problem the task setting, the learning algorithms, or both?
  - Likely both.

Natural Language Inference as a Research Task

Natural language inference (NLI)

also known as recognizing textual entailment (RTE)



James Byron Dean refused to move without blue jeans

{**entails**, contradicts, neither}

James Dean didn't dance without pants

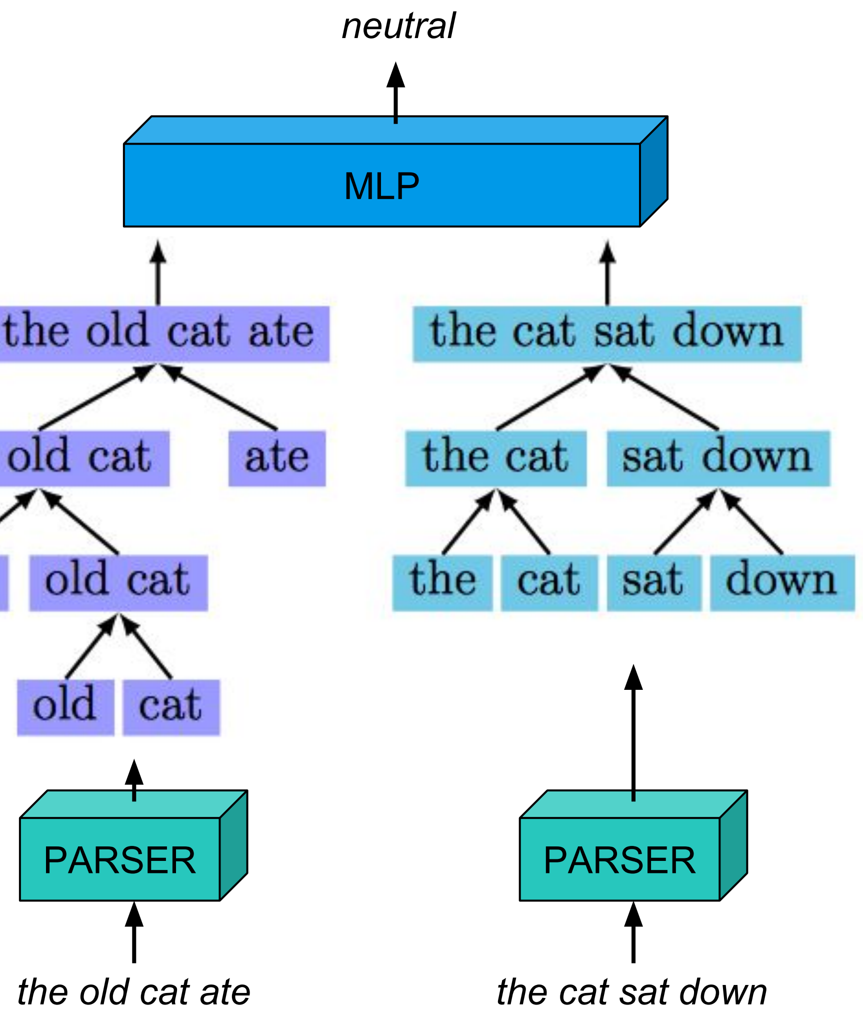
The Stanford NLI Corpus
& The MultiGenre NLI Corpus

Met my first girlfriend that way.	FACE-TO-FACE **contradiction** C C N C	I didn't meet my first girlfriend until later.
He turned and saw Jon sleeping in his half-tent.	FICTION **entailment** N E N N	He saw Jon was asleep.
8 million in relief in the form of emergency housing.	GOVERNMENT **neutral** N N N N	The 8 million dollars for emergency housing was still not enough to solve the problem.
Now, as children tend their gardens, they have a new appreciation of their relationship to the land, their cultural heritage, and their community.	LETTERS **neutral** N N N N	All of the children love working in their gardens.

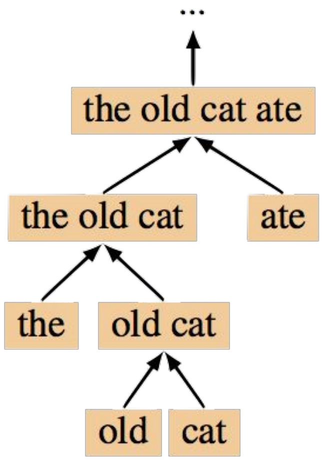
x 1,000,000

More on this in Thursday's talk!

The Setup

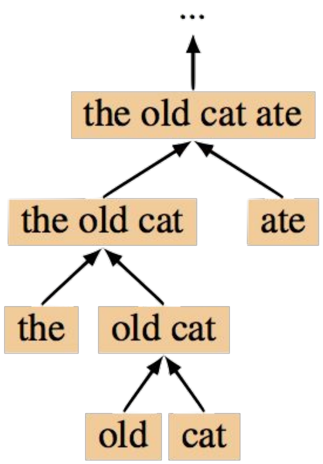

neutral
MLP
the old cat ate
the cat sat down
the old cat
ate
the cat
sat down
the
old cat
the
cat
sat
down
old
cat
PARSER
PARSER
the old cat ate
the cat sat down

Two Design Decisions



- How do we train the parser?
  - Backpropagation training generally won't work
  - Need some workaround
- How do we build the parser?
  - Must be compatible with training strategy above
  - Where possible, should share parameters with NLU model

Results to Date



Three 2017 papers on SNLI report that TreeLSTMs learned trees outperform ones based on trees from an external parser:

- Yogatama et al.:
  - Shift-reduce parser + REINFORCE
- Maillard et al.:
  - Chart parser + soft gating
- Choi et al.:
  - Novel parser + Straight Through + Gumbel softmax

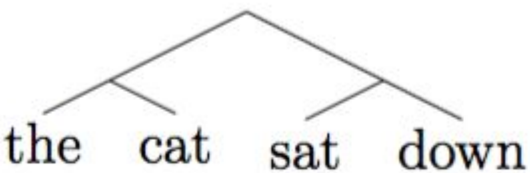
Limited analysis of the resulting parses so far.

Two Models:
RL-SPINN (Yogatama)
ST-Gumbel (Choi)

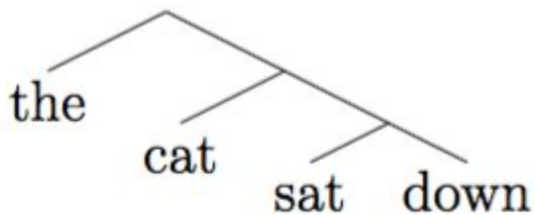
RL-SPINN (Yogatama)

Background: SPINN

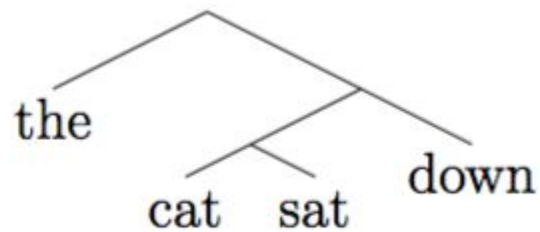
Using transitions to process trees



[SHIFT, SHIFT, REDUCE, SHIFT, SHIFT, REDUCE, REDUCE]

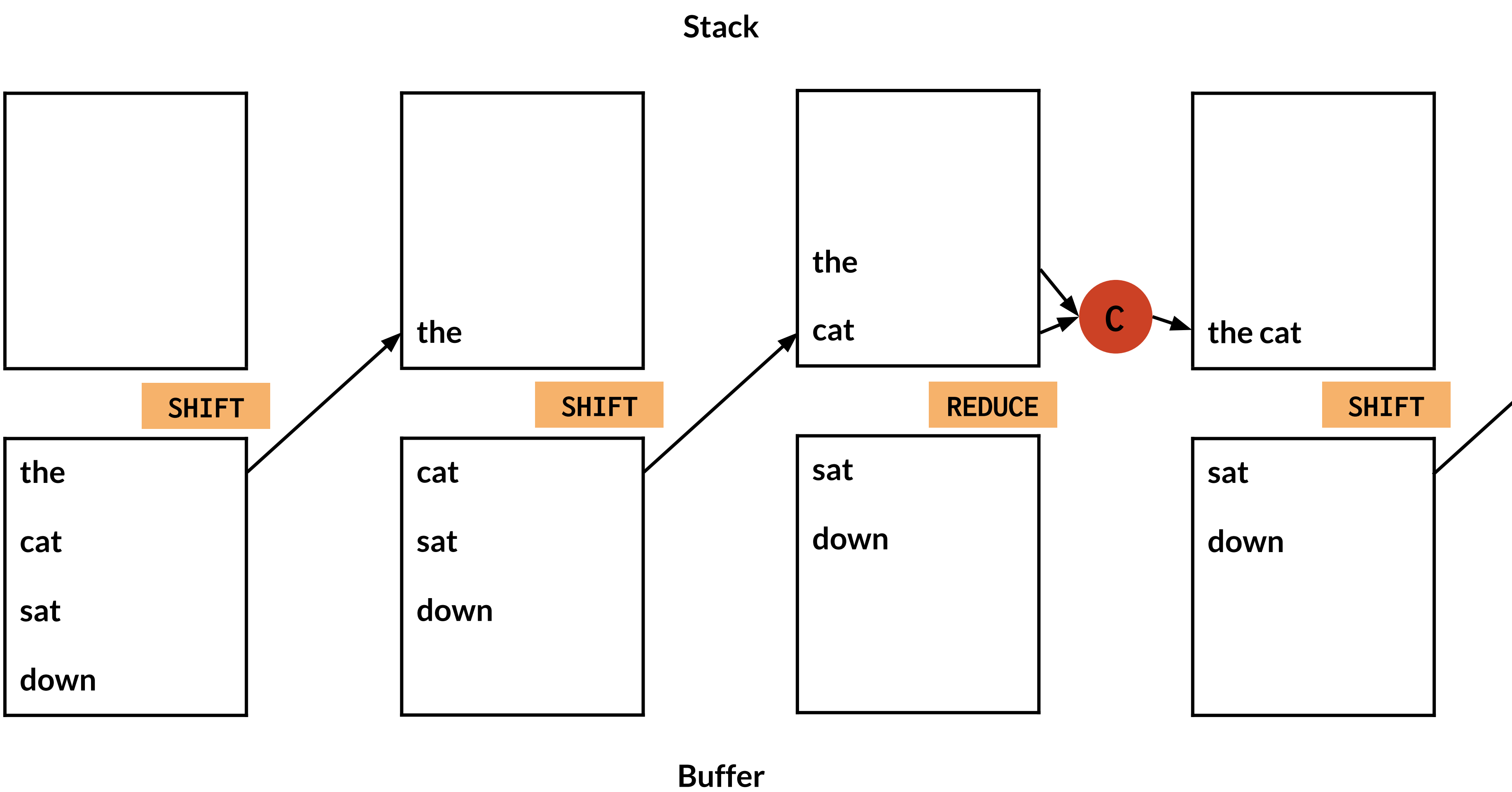


[SHIFT, SHIFT, SHIFT, SHIFT, REDUCE, REDUCE, REDUCE]

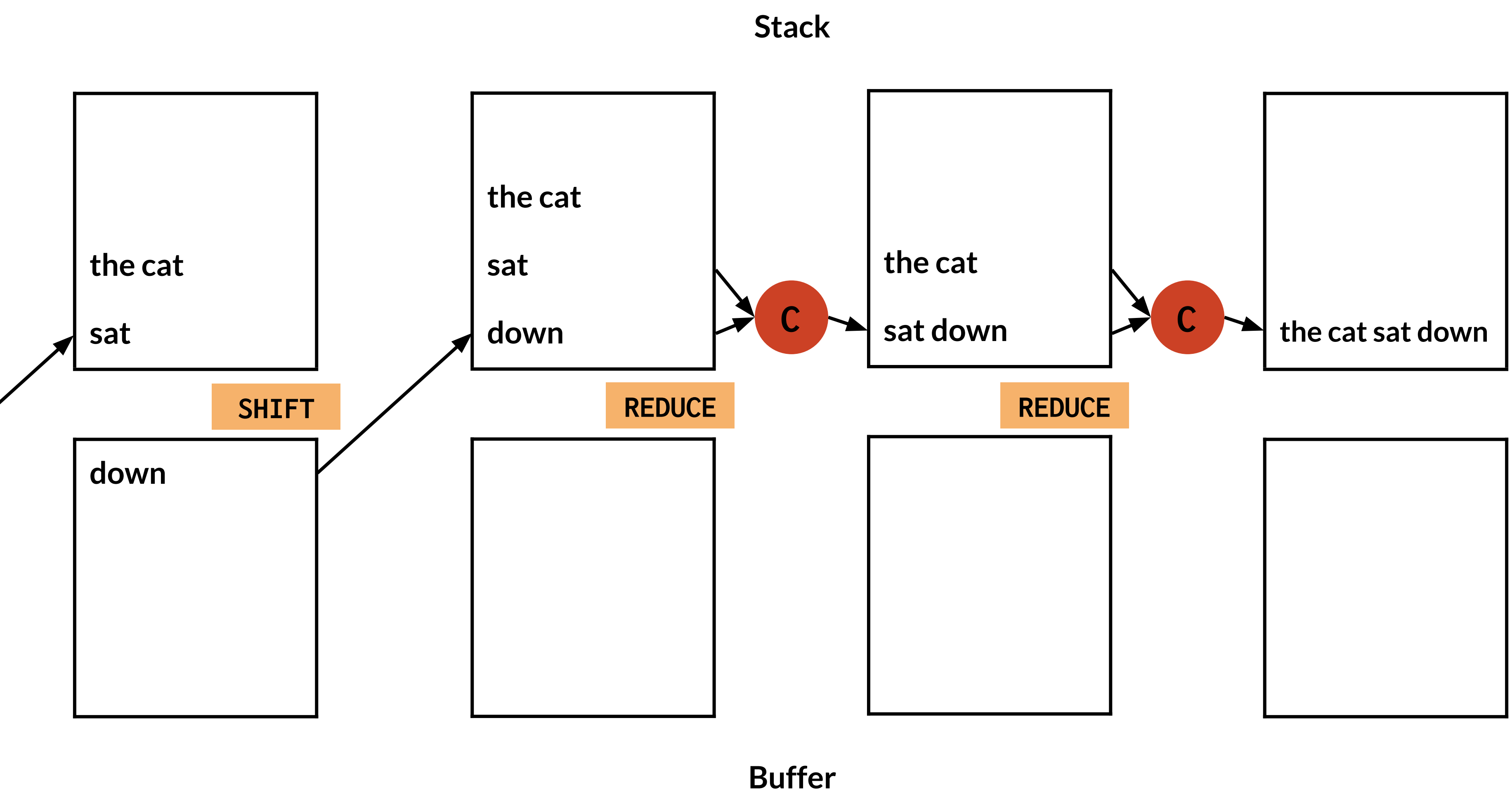


[SHIFT, SHIFT, SHIFT, REDUCE, SHIFT, REDUCE, REDUCE]

Background: SPINN
Using transitions to process trees

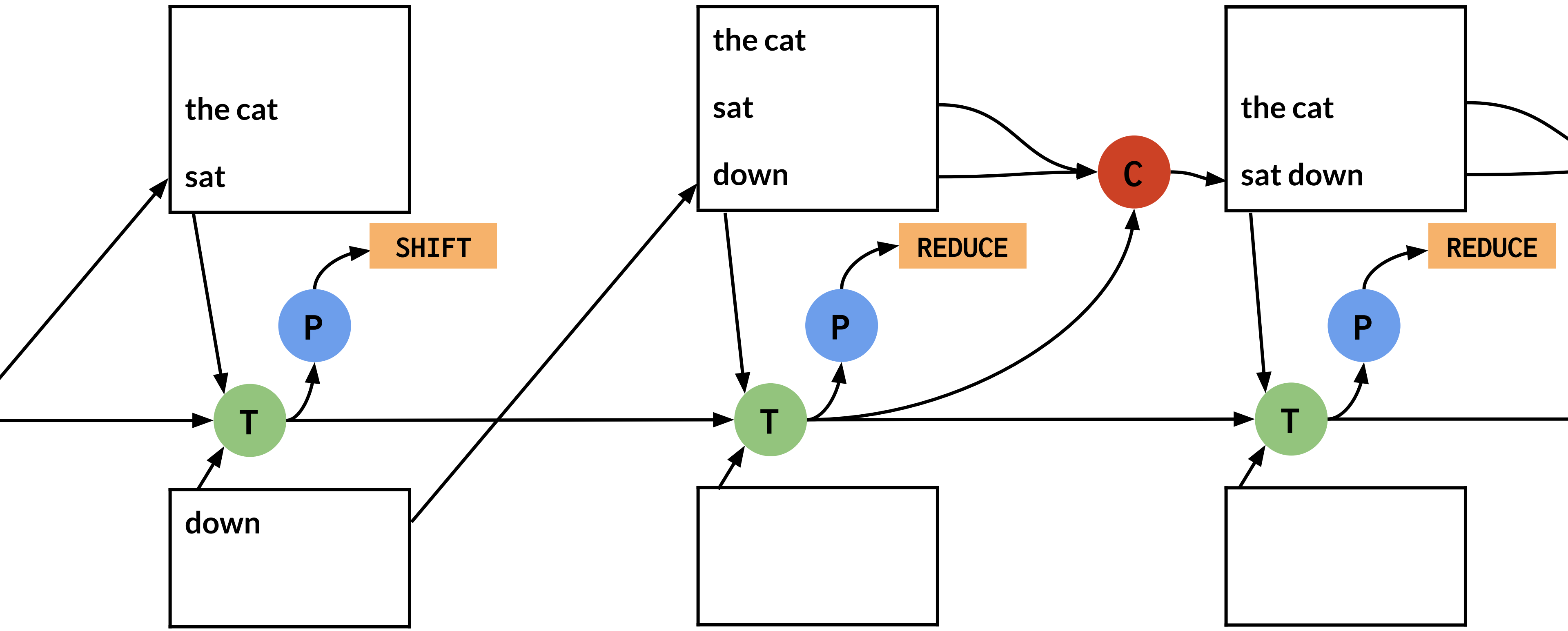


Background: SPINN
Using transitions to process trees



Background: SPINN

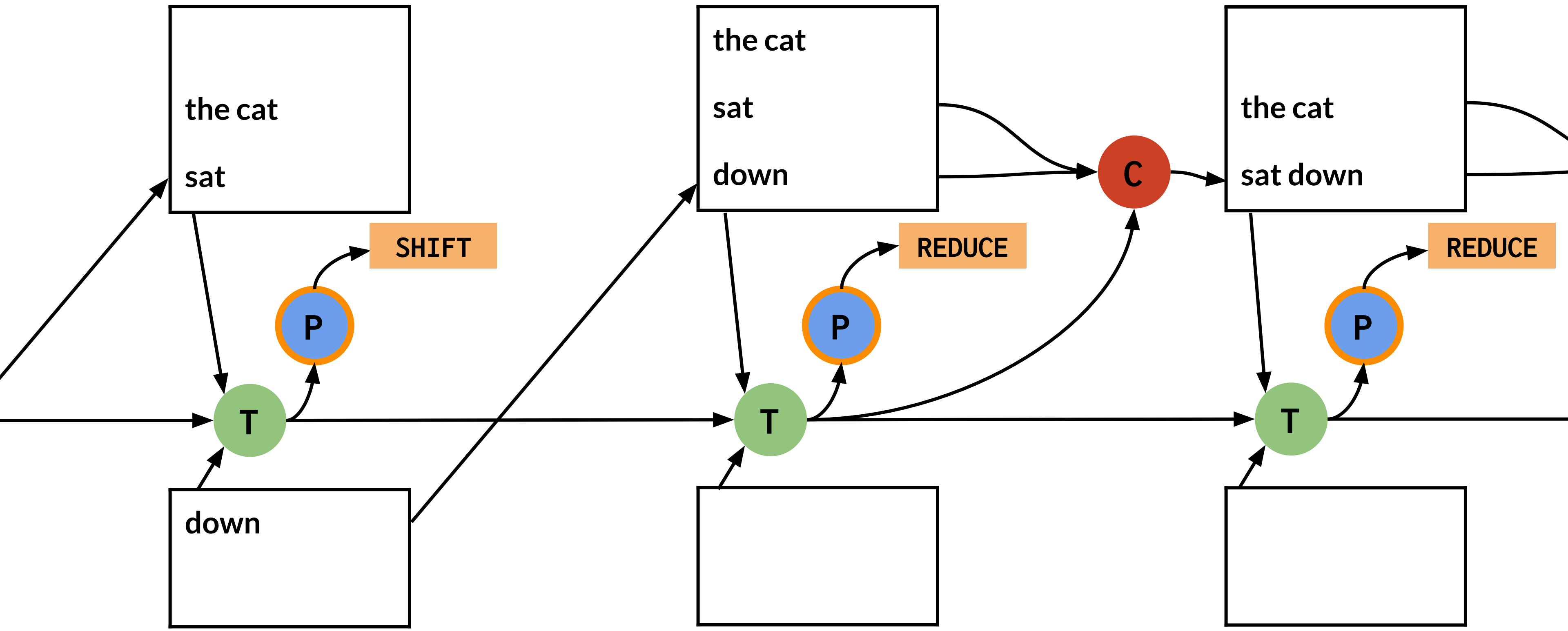
- Shift-reduce parser and TreeRNN share representations
- Supervised by existing parses at training time



Bowman et al. ‘16; cf. Socher et al. ‘11, Cho et al. ‘14

RL-SPINN

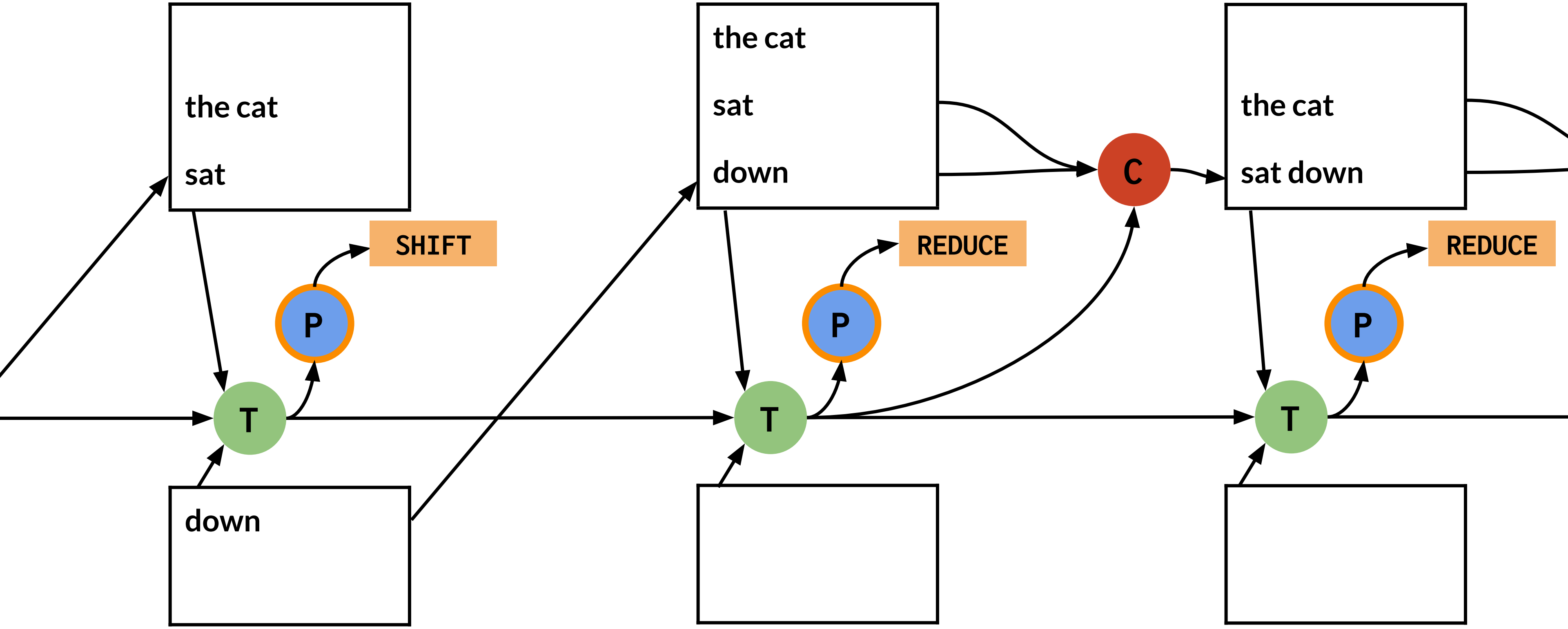
- Shift-reduce parser and TreeRNN share representations
- Parser trained using REINFORCE on NLI objective



Yogatama et al. '17

RL-SPINN

- 100D model only
- Improvements from latent trees!



Yogatama et al. '17

Work to date: ST-Gumbel

- At every layer:
  - Compute every possible merge
  - Score each merge
  - Use Gumbel Softmax to select best
- Straight-Through estimator for gradients

- O(N^2), but GPU-friendly
- Improvements from latent trees!

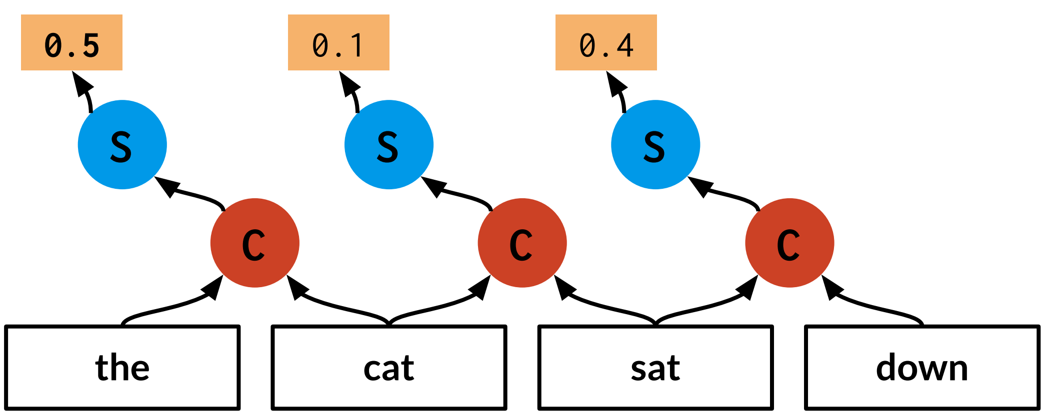
the | cat | sat | down

Choi et al. ‘17

Work to date: ST-Gumbel

- At every layer:
  - Compute every possible merge
  - Score each merge
  - Use Gumbel Softmax to select best
- Straight-Through estimator for gradients

- O(N^2), but GPU-friendly
- Improvements from latent trees!

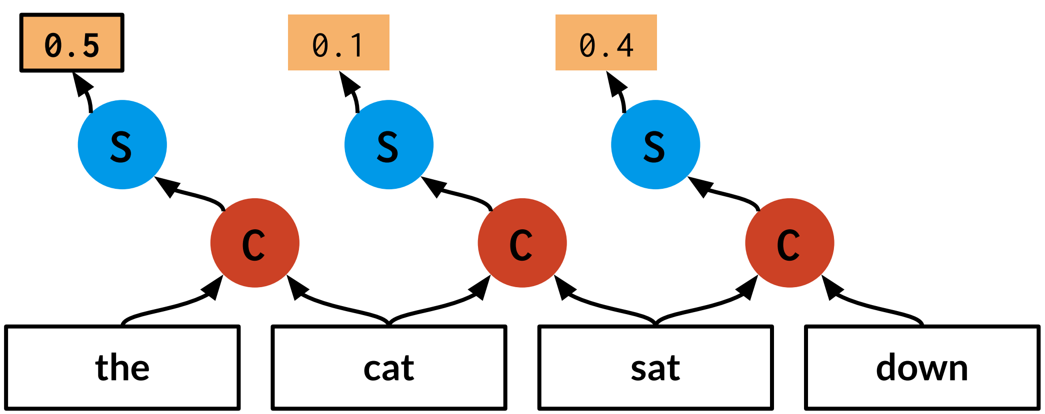


Choi et al. ‘17

Work to date: ST-Gumbel

- At every layer:
  - Compute every possible merge
  - Score each merge
  - Use Gumbel Softmax to select best
- Straight-Through estimator for gradients

- O(N^2), but GPU-friendly
- Improvements from latent trees!

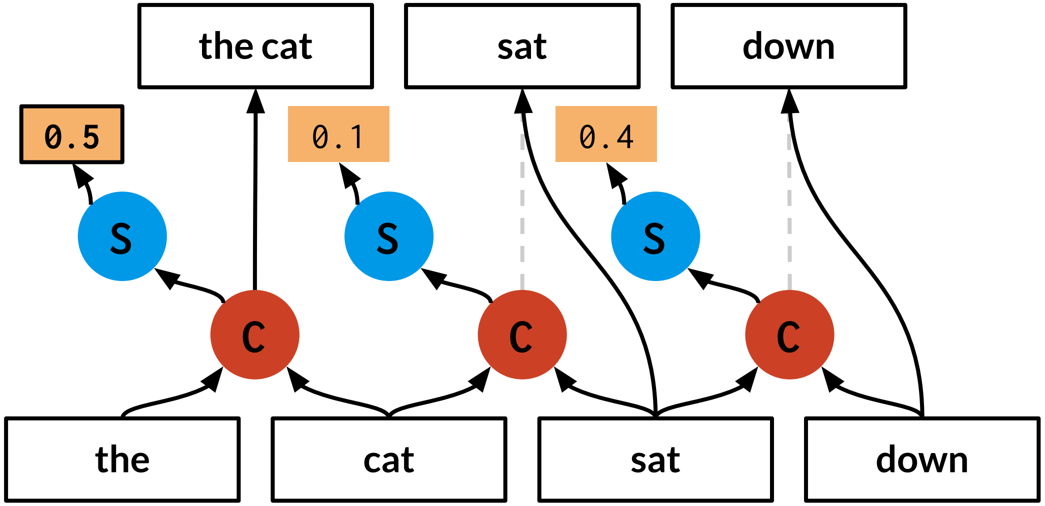


Choi et al. ‘17

Work to date: ST-Gumbel

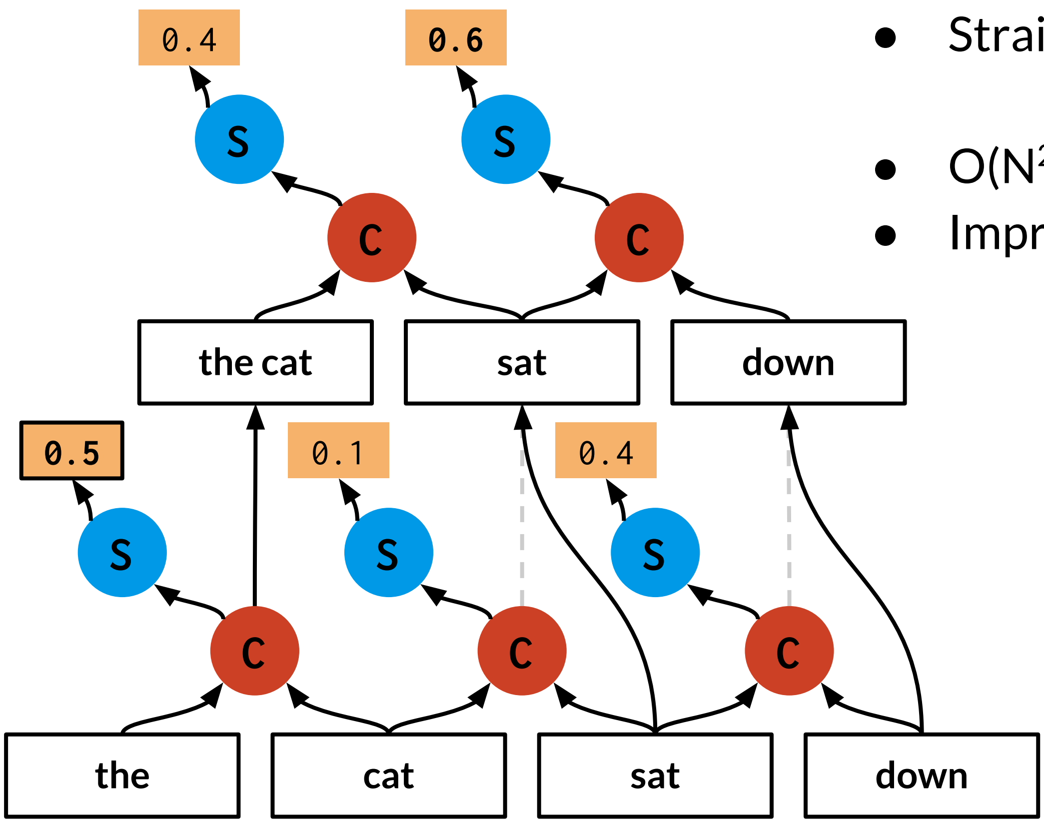
- At every layer:
  - Compute every possible merge
  - Score each merge
  - Use Gumbel Softmax to select best
- Straight-Through estimator for gradients

- $O(N^2)$, but GPU-friendly
- Improvements from latent trees!



Choi et al. ‘17

Work to date: ST-Gumbel

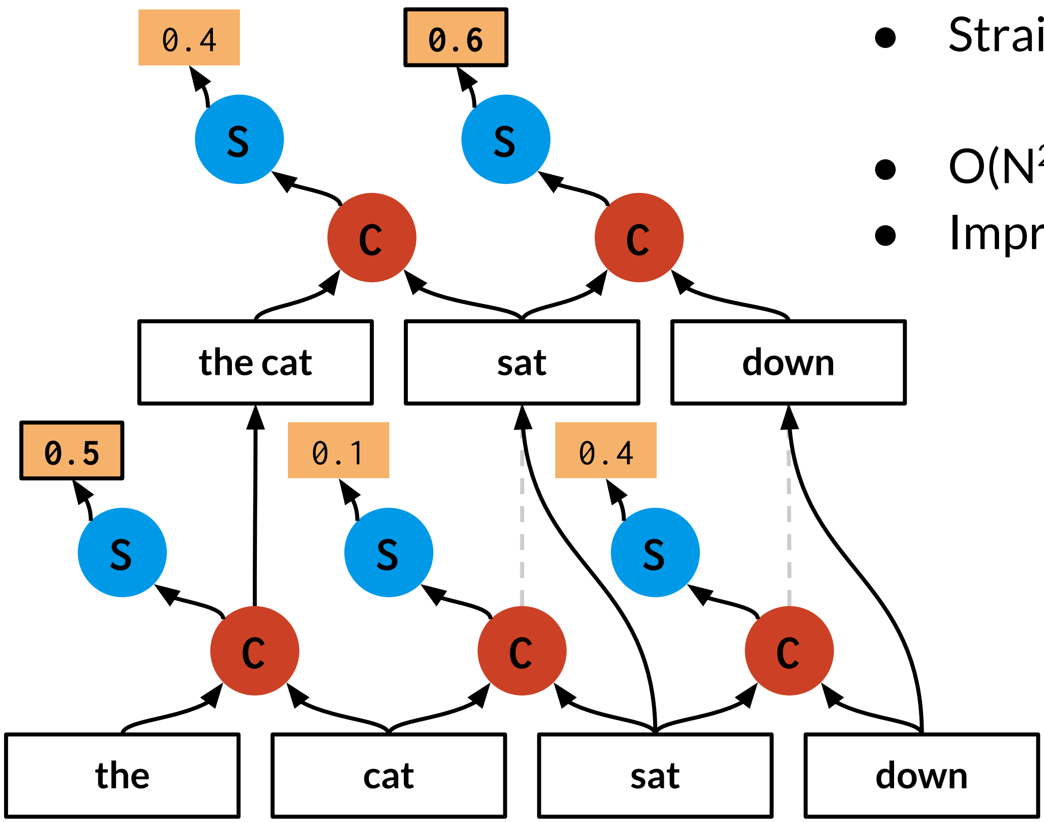


- At every layer:
  - Compute every possible merge
  - Score each merge
  - Use Gumbel Softmax to select best
- Straight-Through estimator for gradients

- O(N^2), but GPU-friendly
- Improvements from latent trees!

Choi et al. '17

Work to date: ST-Gumbel

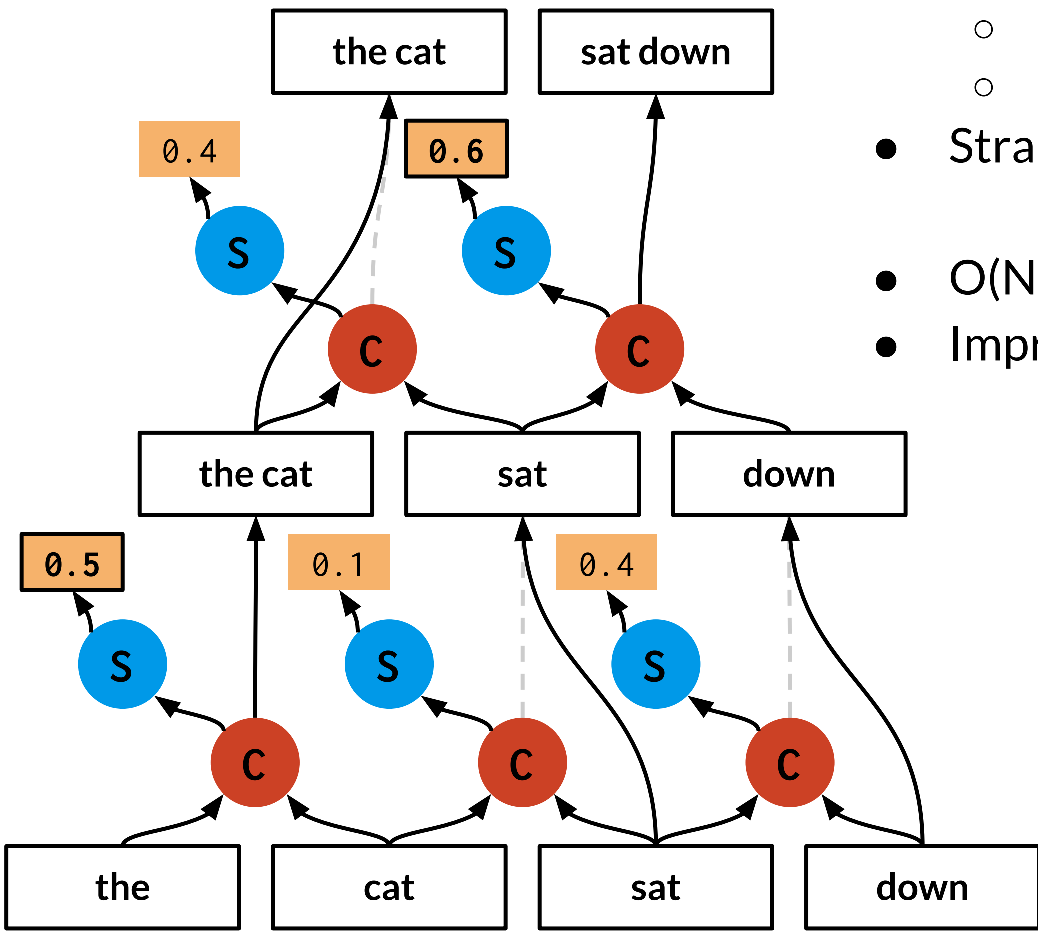


- At every layer:
  - Compute every possible merge
  - Score each merge
  - Use Gumbel Softmax to select best
- Straight-Through estimator for gradients

- $O(N^2)$, but GPU-friendly
- Improvements from latent trees!

Choi et al. ‘17

Work to date: ST-Gumbel

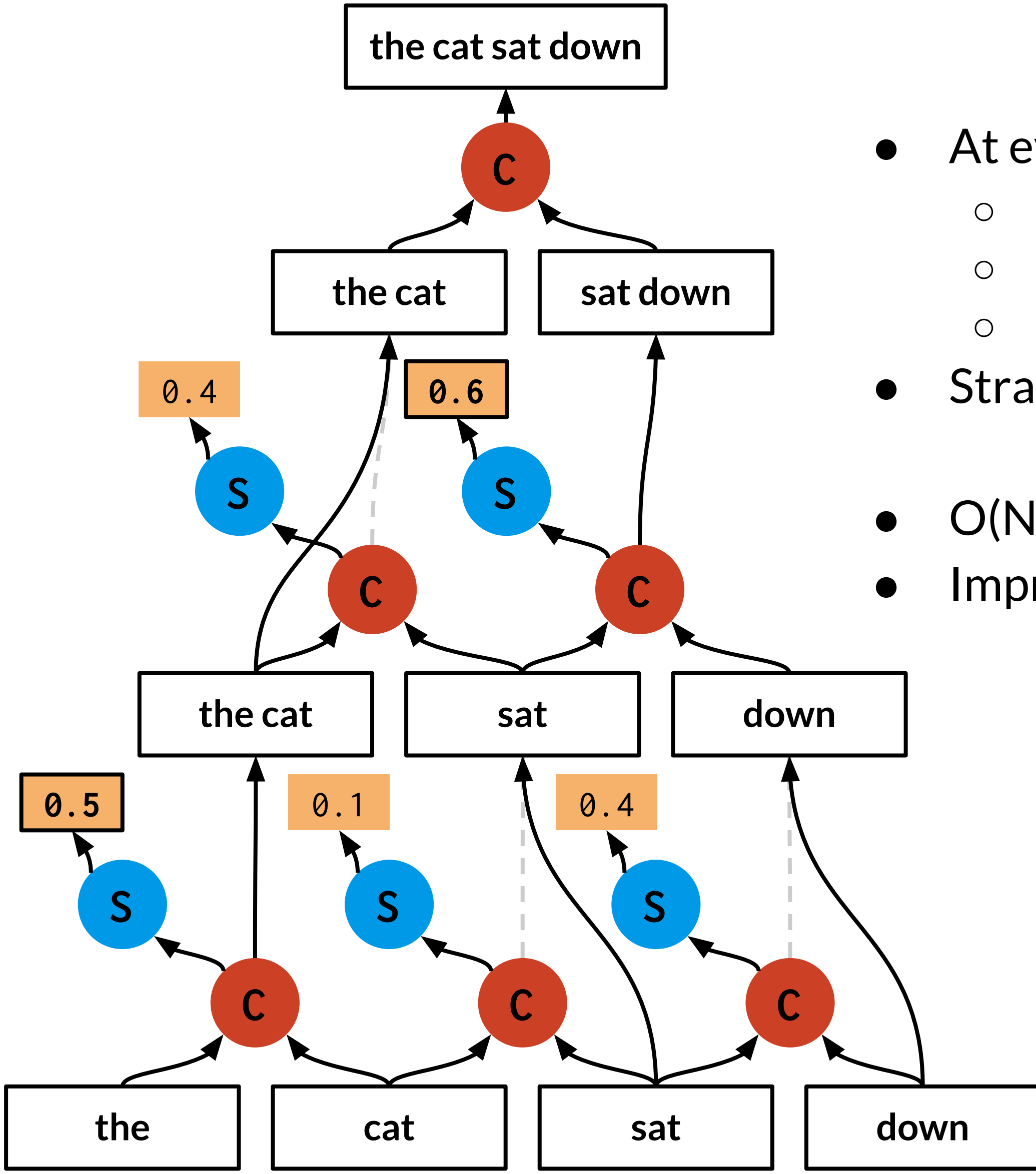


- At every layer:
  - Compute every possible merge
  - Score each merge
  - Use Gumbel Softmax to select best
- Straight-Through estimator for gradients

- $O(N^2)$, but GPU-friendly
- Improvements from latent trees!

Choi et al. ‘17

Work to date: ST-Gumbel



- At every layer:
    - Compute every possible merge
    - Score each merge
    - Use Gumbel Softmax to select best
- Straight-Through estimator for gradients

- $O(N^2)$, but GPU-friendly
- Improvements from latent trees!

Choi et al. ‘17

What grammar do these models learn?

Our findings: Task performance

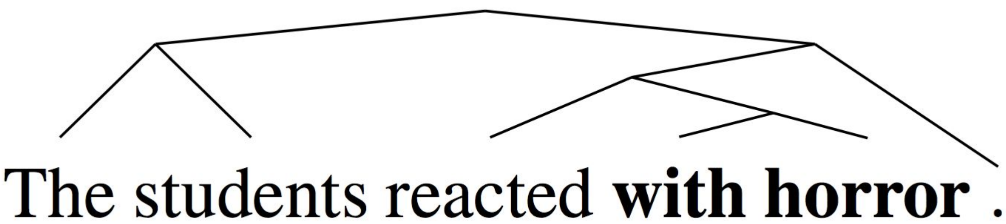
- 300D runs on MultiNLI and SNLI, extensively tuned:
  - Absolute performance on SNLI:
    - Outperform published RL-SPINN (1.8%)
    - Slightly underperform published ST-Gumbel (-0.9%)

Our findings: Major red flags

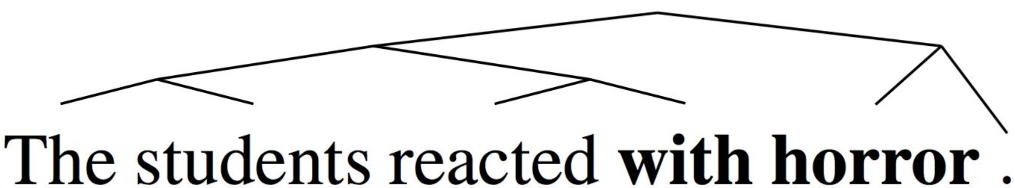
- Our TreeLSTM model (SPINN) does roughly equally well with:
    - Parser trees
    - *Balanced* trees
    - *Random* binary trees
- A plain LSTM (i.e., left-branching trees) does slightly *better.*

cf. Scheible & Schütze '13

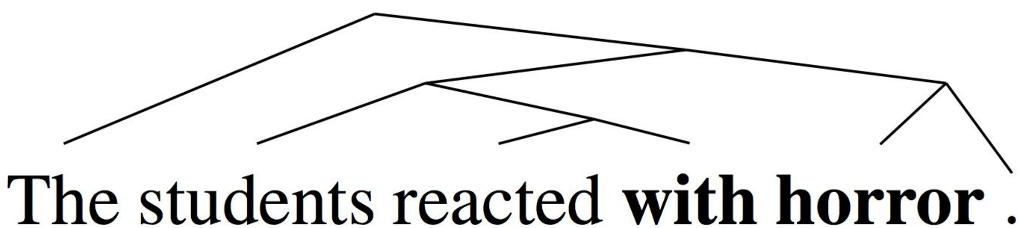
Examples



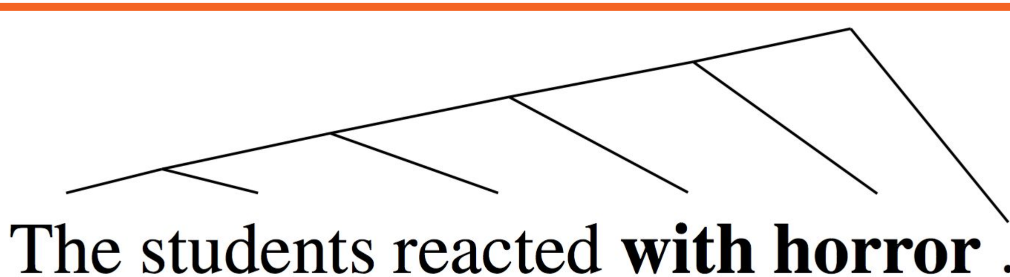
Parser



Balanced



Random



Left-branching (i.e., **recurrent NN)**

Our findings: Consistency

- Across five random restarts, measuring F1 between runs on the MultiNLI Dev Set:
  - RL-SPINN produces highly consistent trees
  - ST-Gumbel produces inconsistent trees, but better than chance

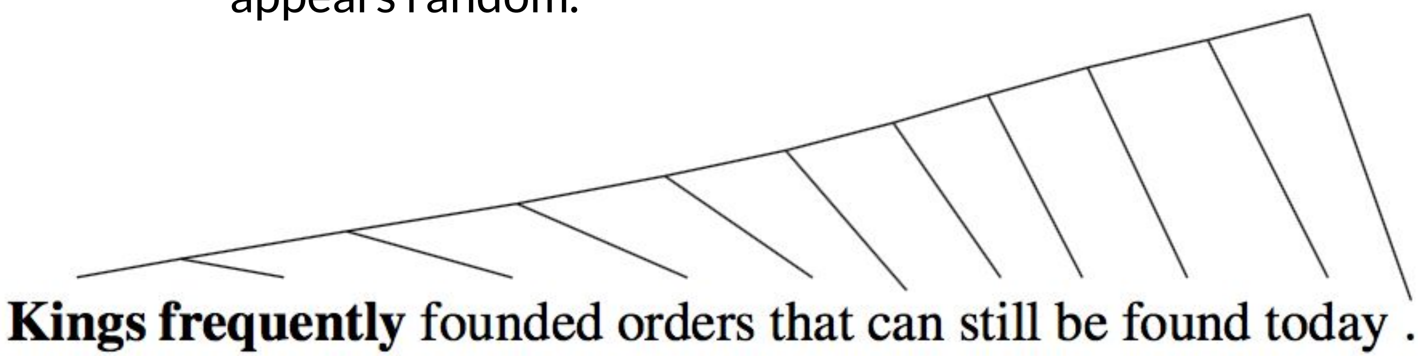
Our findings: PTB

Evaluating on ground-truth Wall Street Journal data:

- Baseline performance on PTB only so-so (~60% F1)
- ST-Gumbel barely above chance (~25% F1)
- RL-SPINN significantly **worse** than chance (~13% F1)

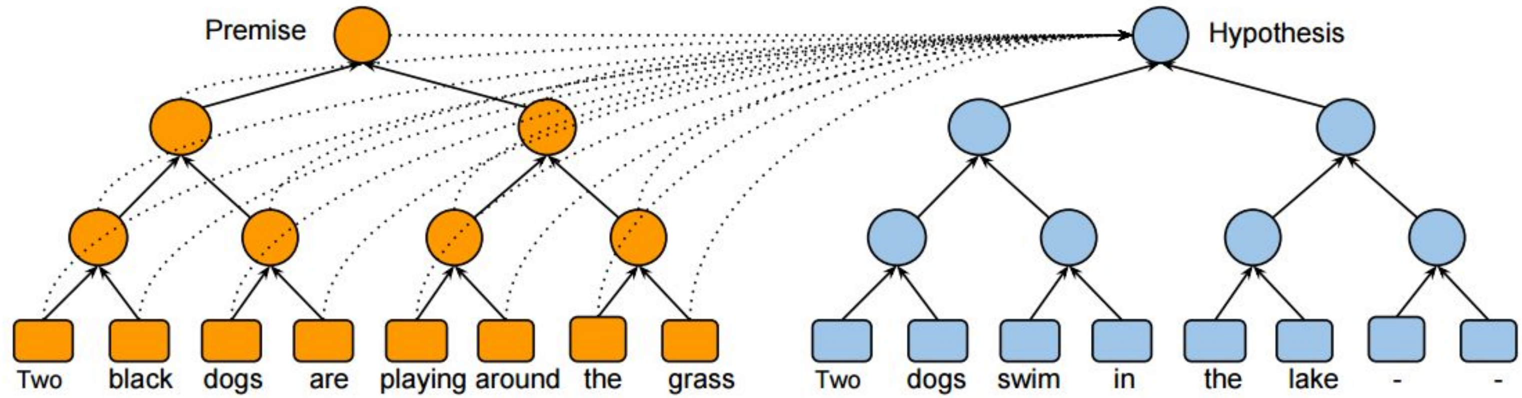
Our findings: Qualitative

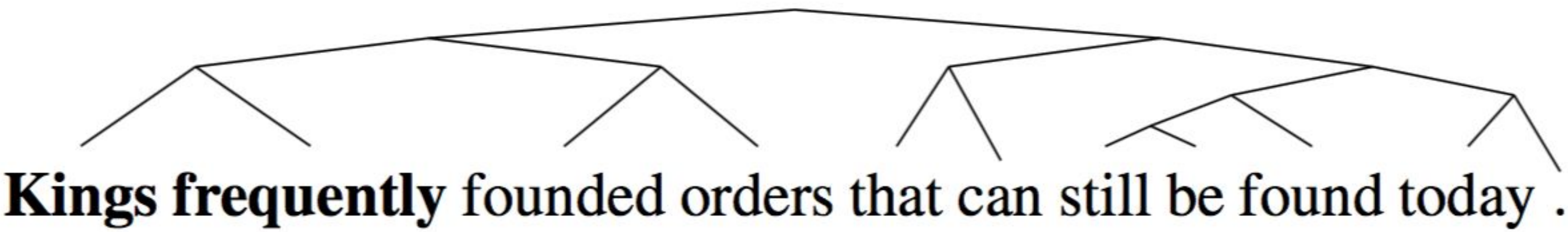
- The RL-SPINN runs that perform best use *strictly left-branching* parses!
- Some runs are less strict, but variation from this trend appears random.



- Explains worse-than-chance parsing performance: English prefers *right*-branching trees.
- Model is *equivalent to* RNN, task performance shows that.

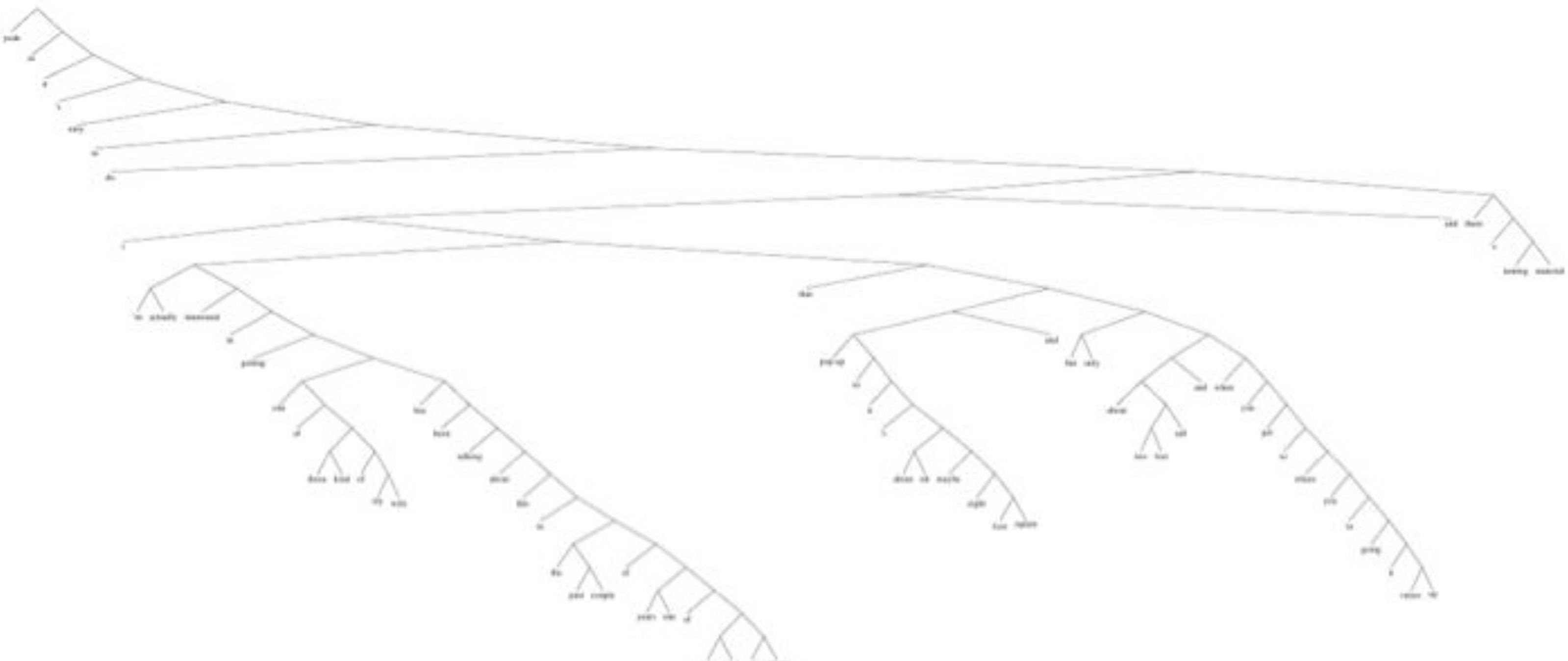
Our findings: Qualitative



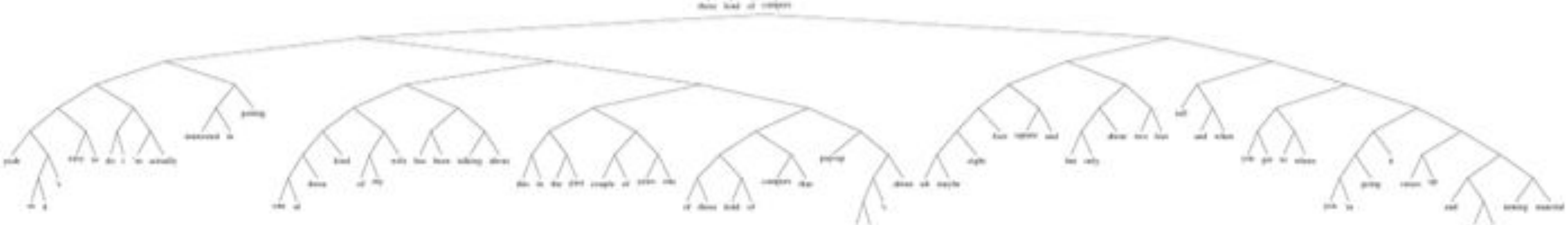


- Disappointing, but others have found these trees to be useful: *Munkhdalai & Yu '16*
- Something about the training procedure seems to help learning/overall performance.

Some Examples

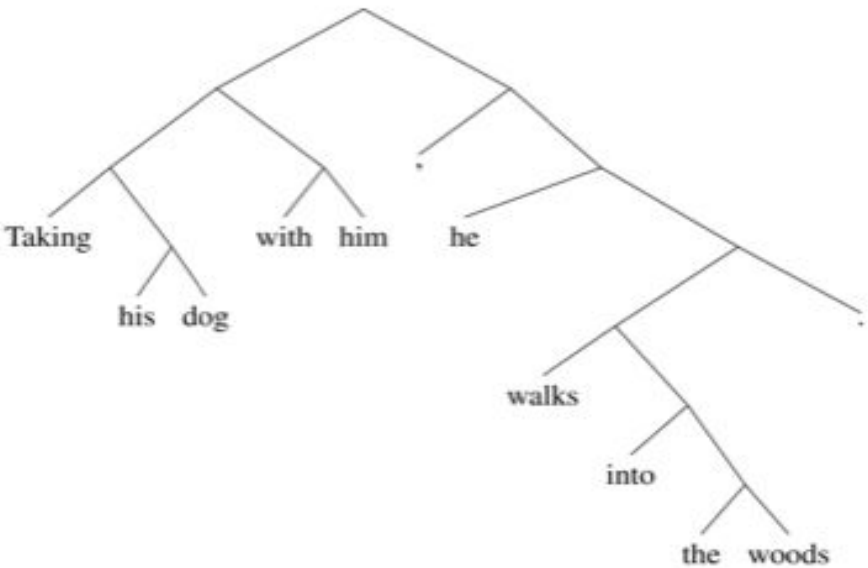


Parser

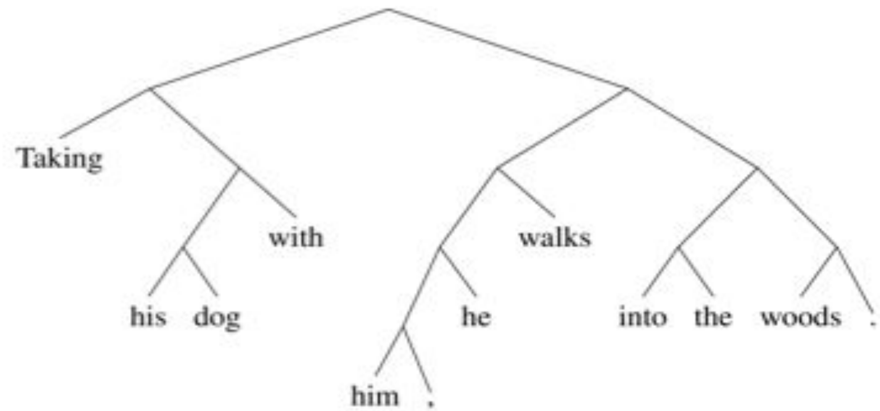


ST-Gumbel

Some Examples



Parser

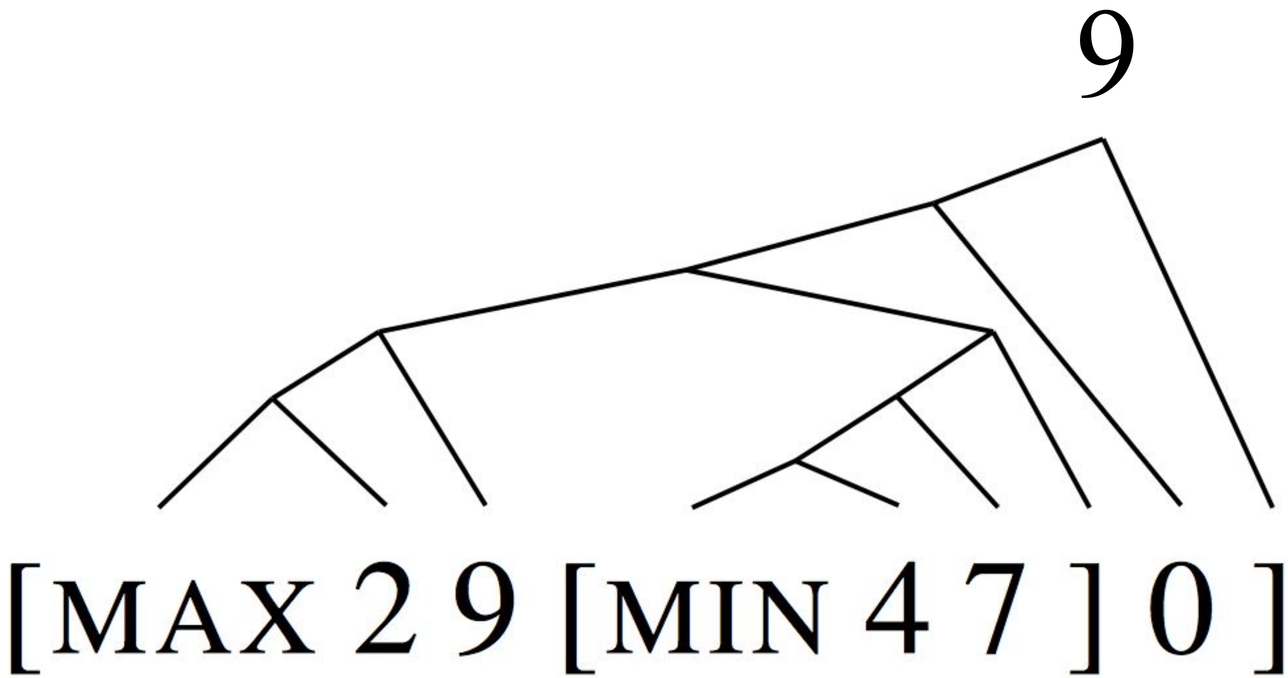


ST-Gumbel

Interim Discussion

- Key motivating hypothesis:
    - With current TreeRNN architectures and tasks, there exists some non-trivial, linguistically interesting parsing strategy that will provide a task advantage.
- Current latent tree learning models don't appear to identify any kind of nontrivial sentence structure.
    - Not surprising: Motivating hypothesis likely false.
    - Additional quantitative analysis (in paper) confirms.
- If motivating hypothesis were true, would these latent tree learning models succeed?

A Diagnostic Task: ListOps



ListOps

- 100k examples (90k train/10k test)
- Manually tuned...
  - Operation set
  - Maximum list length
  - Maximum recursion depth
- ...such that:
  - A tuned 128D TreeLSTM will succeed reliably (acc > 95%)
  - A tuned 128D LSTM RNN will fail reliably (acc < 75%)
- So:
  - Succeeding at the task requires discovering (roughly) the correct tree structure.

A Diagnostic Task: ListOps

```
[MAX 1 5 [MIN 9 2 5 ] ] = 5

[MAX [SM 8 5 [MIN 6 6 1 ] ] [MED 0 1 4 8 6 ] 7 7 ] = 7

[MED 4 1 0 ] = 1

[SM 7 3 2 [MIN 3 2 4 [SM 3 4 [SM 7 9 ] 9 ] ] 8 ] = 2
```

```
[MAX 1 [MIN [MIN 1 [MED 7 [MAX 0 [MED [MAX [MAX [MIN 9 [SM 1 4 0 [MED 9 6 6 ] [MIN 3 7 4 1 ] ] 6 7 [MED 3 0 8 1 [MED [SM 0 3
[MAX 9 0 ] [SM 3 8 5 ] 8 ] 5 8 ] ] ] 9 3 [MAX [SM 0 [SM 9 3 ] ] 4 7 6 [MAX 1 [MED 7 3 ] 8 5 0 ] ] 4 ] 2 0 [MAX [MAX [MAX 4 2
4 ] 7 [MAX 1 4 [MAX 4 4 2 ] [SM 1 9 5 7 8 ] 3 ] ] 7 9 6 ] ] 8 [MIN 2 7 ] ] [MAX [SM 8 [MIN 7 1 [MAX [MAX 4 [MAX [MIN [SM 4 6
[MAX [SM [MAX [MIN [MIN 4 4 ] 7 9 ] [MIN 5 6 ] ] 2 5 5 2 ] 3 9 ] 2 8 ] [SM [MAX 8 9 3 [SM 5 [MIN 4 [MIN 4 2 2 0 ] 6 ] 7 2 ] ]
9 ] [MED 4 [SM 0 [MAX 4 4 [MED [MIN 0 4 [MED [MAX 8 2 4 5 ] 2 1 ] ] 8 2 1 [MIN 1 7 ] ] 0 0 ] 4 7 ] ] ] 1 [MAX 1 4 ] 2 [SM 0 1
2 5 9 ] ] [MED 3 [SM 3 [SM [MIN [MIN [SM 6 [MED [SM 0 3 [SM 9 6 1 2 ] 7 [MIN 9 0 4 ] ] [MIN [MIN 4 9 6 ] 5 4 [SM 3 7 8 6 5 ]
] 2 9 ] 2 8 ] 9 [MIN 0 3 [SM 1 6 ] ] 1 ] 9 1 ] 7 [MAX [MIN [MED 0 [SM [MAX [MAX 9 5 5 ] 0 0 ] 4 3 [MED 8 7 5 2 0 ] 2 ] ] [MAX
5 7 9 ] ] [MED 7 6 ] 3 ] 7 ] 2 ] [MIN 8 8 ] 0 ] [MAX 2 2 7 1 9 ] 1 ] 7 ] ] ] 7 1 2 ] 3 [SM [MED [MAX 5 8 5 9 1 ] 8 [SM 8 7 ]
] 6 8 [MAX 4 7 ] [MED 7 0 4 3 [MIN 8 1 3 6 ] ] ] ] 8 [SM 4 [MAX 9 9 9 7 ] [MIN 1 6 [MAX [SM [SM [MIN 9 8 ] [SM 1 0 ] ] 7 ] 9
6 1 ] 2 ] ] ] 6 4 ] 4 ] 3 [MIN [MAX [SM 2 3 [MED [MAX [MED 4 5 [SM [SM 2 5 7 ] 1 6 ] ] [SM 7 3 2 [SM [MED [MAX 2 [MED 0 8 ] 6
[MED [MIN 2 2 ] 0 7 [SM [MIN 3 [MAX 3 9 4 [MIN 7 0 2 ] ] [MED 9 [SM 2 [MAX 9 3 4 2 ] 9 [MED 7 1 ] ] 3 6 ] 5 ] 5 4 7 ] ] ]
[MIN [MIN [SM 0 4 8 2 5 ] [MIN 3 4 ] 4 [SM 3 [MED [MED 2 3 5 2 1 ] 0 ] 5 6 0 ] 7 ] 8 7 3 ] ] 7 3 4 ] ] 9 8 ] [MAX 3 5 [MAX 1
5 ] 4 ] [SM 6 [SM [MED [SM 2 8 5 ] [MED [SM 0 3 9 [MIN 0 6 8 ] 4 ] 4 [MAX [MIN 0 1 1 ] 4 4 ] ] ] [MIN 5 [MIN 5 [SM [MAX 9 0 0
] 5 3 1 7 ] 7 0 ] 8 3 5 ] ] 4 0 9 ] [MED 9 7 2 ] 8 ] ] 9 8 8 ] 7 8 ] ]

= 7
```

ListOps

- Possible inputs:
  - `0 … 9`
  - `[min`
  - `[max`
  - `[median`
  - `[sm` (sum list, modulo 10)
  - `]`
- Possible outputs:
  - `0 … 9`

Results

- LSTM RNN: 73.3
- TreeLSTM: **98.7**
- RL-SPINN: 64.8
- ST-Gumbel: 59.9

Existing latent tree learning models do not appear to be able to identify useful structure, even when such structure is known to exist.

Open Questions

How do we get to effective latent tree learning?

- Better composition functions?
- Harder or richer tasks (cf. Shen et al. ‘18)?
- Warm start methods (cf. Yogatama et al. ‘17)?

Part II

Learning to Match Expert Acceptability Judgments

Alex Warstadt
Samuel R. Bowman

LSA 2018/in prep

(The) Open Question

To what extent is strong prior knowledge (strong universal grammar) needed to learn linguistic competence?

How do these sentences sound?

- The earth revolves around the sun.
- The earth revolves the sun.

- The earth circles around the sun.
- The earth circles the sun.

How do these sentences sound?

- The earth revolves around the sun.
- *The earth revolves the sun.

- The earth circles around the sun.
- ?The earth circles the sun.

Acceptability Judgments

The task:
Evaluate whether a sentence is *acceptable* in some natural language.

- Intended as a direct test of speaker *competence*.
- Primary form of empirical data in many areas of linguistic theory (Chomsky '65; Schütze '96).

Acceptability Judgments

Linguists working on their own native language(s) often provide their own judgments as data in published work.

Ross ‘67:

variables. And yet, just as was the case with rule (1.10), Extraposition from NP, it is easy to see that (1.13) is far too strong, for it will generate infinitely many non-sentences, such as those in (1.15).

(1.15) * What did Bill buy potatoes and?

* What did that Bill wore surprise everyone?

This Work:

Matching Expert Judgments

- New corpus of 7k expert judgments from published work on English.
- Modeling: Semi-supervised learning with NNs
  - Pretrain an RNN sentence classifier with a proxy *real-fake classification task* on the 100M-word British National Corpus.
  - Fine-tune the sentence classifier on a small sample of expert judgments.

Why Semi-Supervised Learning?

- Why any acceptability judgment supervision?
    - Ensures that the model learns the correct *definition* of acceptability.
    - Possible that this can be done analytically, but not clear how.
- Why not just use full supervision?
    - We want to give the model a chance to learn some phenomena *without* expert judgments.
        - If the system succeeds at these phenomena at test time, it has succeeded in learning them from the unlabeled data.

Prior work

- Lawrence et al. ‘00 train RNNs to perform a similar task, but over POS tags rather than words.
- Wagner et al. ‘09 learn to distinguish real English sentences from manipulated ones.
- **Lau, Clark, & Lappin ‘15/‘16** use *unsupervised* learning to predict gradient non-expert acceptability judgments.

Acceptability Judgments

In our work:

Morphosyntactic

- *Maryann should leaving.

Syntactic

- *What did Bill buy potatoes and?

Semantic

- *Kim persuaded it to rain.

Acceptability Judgments

Not used in our work:

Pragmatically awkward

- #Bill pushed Harry off the sofa for hours.

Prescriptively forbidden

- It’s easy to find prepositions to end a sentence with.

Semantically unavailable readings

- *He$_i$ loves John$_i$. *(intended: John$_i$ loves himself$_i$)*

The Draft Corpus

source	topic	size	1	0
Kim and Sells (2008)	syntax textbook	2036	1453	583
Levin (1993)	lexical semantics	1835	1339	496
Ross (1967)	extraction, etc.	1092	692	400
Baltin and Collins (2008)	syntax handbook	925	624	301
Sag et al. (1999)	syntax textbook	472	334	138
Culicover and Jackendoff (1999)	comparatives	260	161	99
Rappaport Hovav and Levin (2008)	dative alternation	162	115	47
Goldberg and Jackendoff (2004)	resultative	111	87	24
	Entire Corpus	6893	4805	2088

- Broad coverage of syntax/semantics/morphology
- Random train/dev/test split
- Original author's judgments used
- Original distribution of judgments preserved: ~70% acceptable

The Draft Corpus

source	topic	size	1	0
Kim and Sells (2008)	syntax textbook	2036	1453	583
Levin (1993)	lexical semantics	1835	1339	496
Ross (1967)	extraction, etc.	1092	692	400
Baltin and Collins (2008)	syntax handbook	925	624	301
Sag et al. (1999)	syntax textbook	472	334	138
Culicover and Jackendoff (1999)	comparatives	260	161	99
Rappaport Hovav and Levin (2008)	dative alternation	162	115	47
Goldberg and Jackendoff (2004)	resultative	111	87	24
	Entire Corpus	6893	4805	2088

- Removed unwanted judgment types (pragmatic, reading-based, …)
- Removed *questionable* examples (?, ??, *?)
- Replaced extremely rare words (especially proper names)

Random Sample

0	*At Mrs. Parker's lodged an old woman.*
1	*Everybody who has ever, worked in any office which contained any typewriter which had ever been used to type any letters which had to be signed by any administrator who ever worked in any department like mine will know what I mean.*
1	*Sharon shivered.*
1	*We believed John to try to leave the country.*
0	*It tried to bother me that Chris lied.*

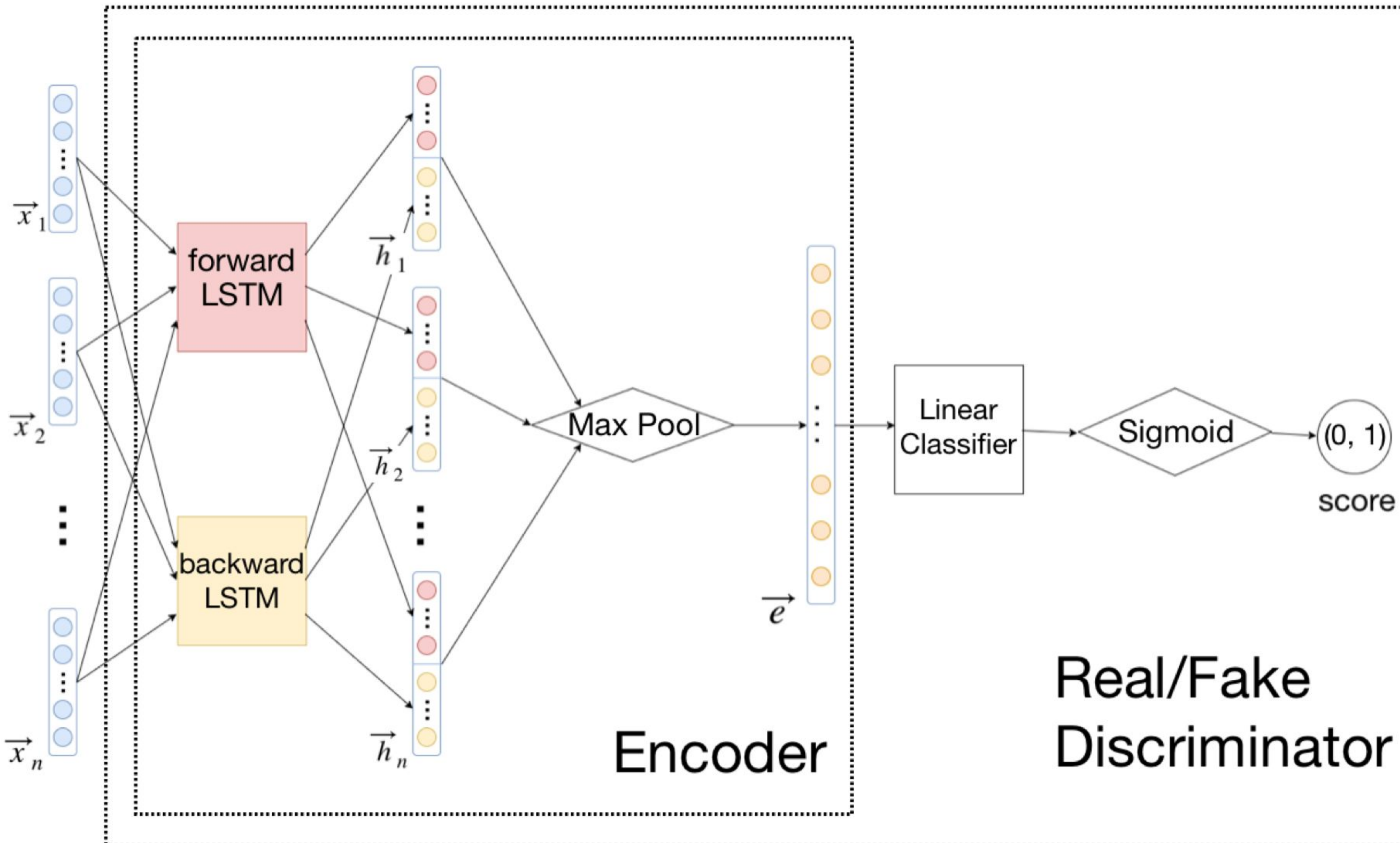
The Proxy Task

Discriminate real BNC sentences from fake sentences generated by either:

- training an RNN language model on BNC and sampling from it…
    - *Investors are powerful discounts, mostly they seem particularly.*
- ...or randomly permuting a few of the words in a real BNC sentence.
    - *The hard-to-handle stop spread rumour to was me working.*

~200M words of training data, no linguistic knowledge included.

The Model

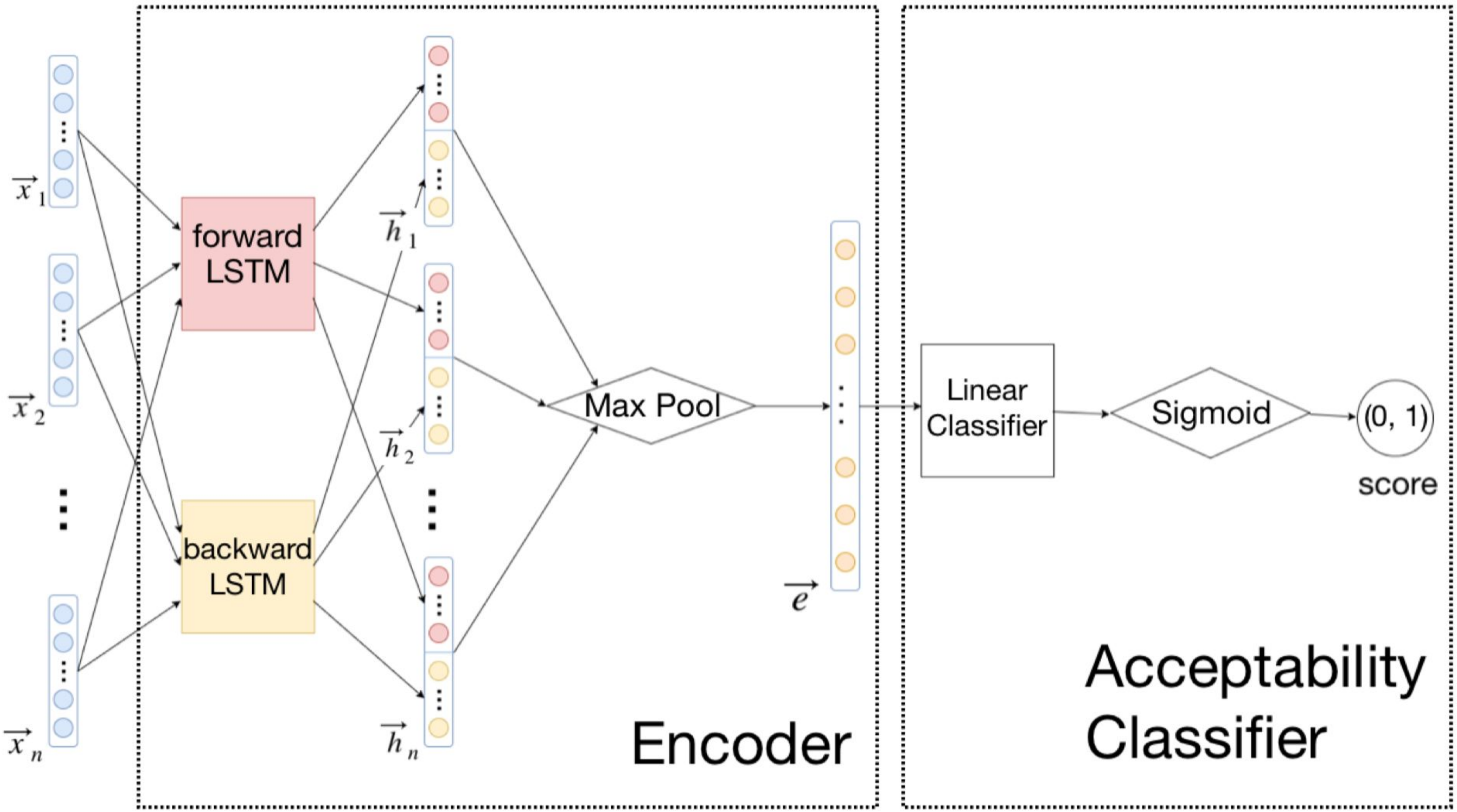


$\vec{x}_1$
$\vec{x}_2$
$\vec{x}_n$
forward LSTM
backward LSTM
$\vec{h}_1$
$\vec{h}_2$
$\vec{h}_n$
Max Pool
$\vec{e}$
Encoder
Linear Classifier
Sigmoid
(0, 1)
score
Real/Fake Discriminator

Proxy Task Results

fake data	hidden size	number of layers	accuracy (Matthews)
LM & permuted	1034	3	0.883 (0.88)
LM	1515	1	0.830 (0.84)
permuted	689	4	**0.884 (0.88)**

The Model



$\vec{x}_1$
$\vec{x}_2$
$\vec{x}_n$
forward LSTM
backward LSTM
$\vec{h}_1$
$\vec{h}_2$
$\vec{h}_n$
Max Pool
$\vec{e}$
Encoder
Linear Classifier
Sigmoid
(0, 1)
score
Acceptability Classifier

Acceptability Results

encoding data	encoding size	hidden size	accuracy (Matthews)
LM & permuted	1034	21	**0.75 (0.497)**
permuted	689	158	0.72 (0.438)
LM	279	93	0.66 (0.312)
CBOW	–	928	0.59 (0.223)

Random Samples

sentence	model rating	gold label
Tessa cut.	0.13	0
Harriet interconnected the pieces.	0.88	1
The boy whose playing the piano loudly I disliked was a student.	0.13	0
Which witnesses testified against defendants who incriminated themselves?	0.95	1

Minimal Pairs

sentence	model rating	gold label
I demand that the more John eat, the more he pays.	0.09	0
I demand that John pay more, the more he eats.	0.54	1

Results by Source (four largest)

dataset	accuracy (Matthews)
overall	0.75 (0.497)
Levin 1993	0.84 (0.650)
Ross 1967	0.79 (0.566)
Kim and Sells 2008	0.83 (0.66)
Baltin and Collins 2008	0.81 (0.618)

- The model has the easiest time with the mostly-local phenomena in Levin.
- The model has the hardest time with the mostly long-distance phenomena in Ross.
- Much better than chance (M=0) on all sources.

Dative Alternation & Ditransitives

sentence	model rating	gold label
Nora sent the book to Peter.	0.85	1
Bill sent Tom a package.	0.64	1
Nora sent the book from Paris to London.	0.60	1
Nora sent at the book to Peter.	0.41	0
Jake sent the box towards Carson.	0.27	0
Felicia sent the box out of the storeroom.	0.73	0
A review copy of the book was sent to her by the publisher.	0.66	1
I sent the devil the salesman.	0.36	0

Islands

sentence	model rating	gold label
Which problem do you wonder whether John said Mary solved?	0.09	0
This rock is too heavy for us to try to claim that we picked up.	0.40	0
What did Bill cook and wash the dishes?	0.76	0
The money which I have hopes that the company will squander amounts to $ 400,000.	0.45	1
Why do you wonder whether she will invite me?	0.75	1

Interim Conclusions

- Under semi-supervised training RNNs can learn to match expert acceptability judgments correctly much of the time.
- These models capture at least some structure—they outperform the simple order-insensitive CBOW.

Work in Progress (short term)

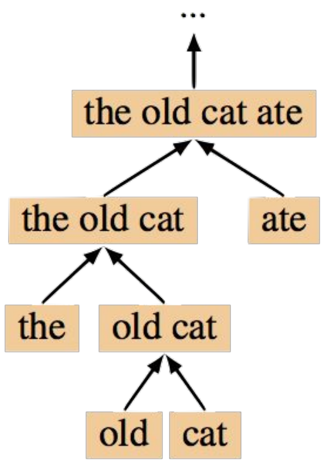
- How do these semi-supervised methods compare with the unsupervised methods of Lau et al.?
- Do non-expert judgments (from, e.g., Lau et al.) yield qualitatively different results?
- Can models learn to correctly judge entire families of phenomena without acceptability supervision?
- Are word vectors necessary or useful?

Open Questions (longer term)

- How much bias will we need to reach human-level performance?
- To what extent can supervision from semantic tasks like NLI or translation help?
- To what extent do existing popular models learn this information already?

Wrapping up

Recap



Two early efforts toward using evidence from NNs to inform linguistic questions.

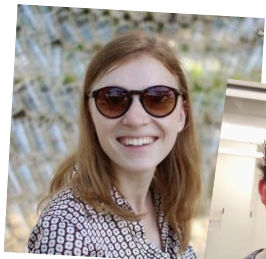
Part I:

- Latent tree learning promises to discover compositional structure in data.
- Early results appear to be misleading: Open problem.

Part II:

- Semi-supervised learning can produce a reasonably good model of acceptability, but not currently at human level on any major phenomena.

Thanks!




Code/data:

- Part I: http://nyu.edu/projects/bowman
- Part II: Work in progress. Contact *bowman@nyu.edu*

Plus:

- Adina Williams is seeking a postdoc position!

These projects were supported in part by a Google Faculty Research Award, gifts from Tencent Holdings and NVIDIA, and a grant from Samsung Research.