STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

CS584 Natural Language Processing

Dependency Parsing

Ping Wang
Department of Computer Science
Stevens Institute of Technology

Today's plan

- Syntactic structure
- Dependency grammar
- Transition-based dependency parsing
- Neural dependency parsing

Two views of linguistic structure:

1. Constituency (phrase structure)

- Phrase structure organizes words into nested constituents.
  - can represent the grammar with Context-Free Grammars (CFG) rules
- **Starting units**: words are given a category (part of speech = POS)

  the, cuddly, cat, by, the, door

  Det Adj N P Det N
- **Words combine into phrases** with categories

  the cuddly cat, by the door

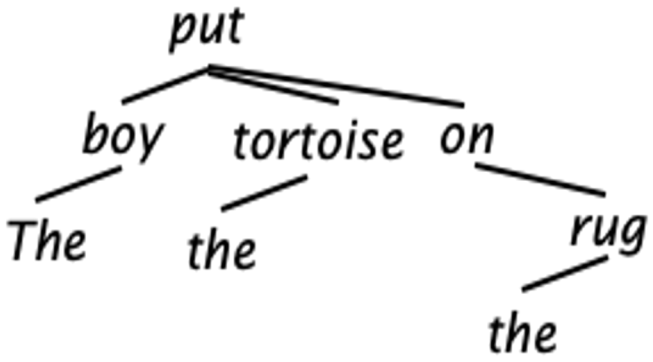
  NP -> Det Adj N PP->P NP
- **Phrases can combine into bigger phrases** recursively

  the cuddly cat by the door

  NP -> NP PP

Two views of linguistic structure:
2. Dependency structure

- Dependency structure shows which words depend on (modify or are arguments of) which other words.



Why do we need sentence structure?

- We need to understand sentence structure in order to be able to interpret language correctly.
- Humans communicate complex ideas by composing words together into bigger units to convey complex meanings.
  - "*Inside the carriage, which is built on several levels, he sits in velveteen darkness, with nothing to smoke, feeling metal nearer and farther rub and connect, steam escaping in puffs, a vibration in the carriage frame, a poising, an uneasiness, all the others pressed in around, feeble ones, second sheep, all out of luck and time: drunks, old veterans still in shock from...*" - *Gravity's Rainbow*
- We need to know what is connected to what

Prepositional phrase attachment ambiguity

- Arises in language when a prepositional phrase can be attached to more than one part of a sentence, resulting in different possible interpretations of the sentence's meaning.

- “San Jose cops kill man with knife”
- "The man saw the woman with the telescope."
- “Scientists count whales from space”

PP attachment ambiguities multiply

- A key parsing decision is how we 'attach' various constituents
  - PPs, adverbial or participial phrases, infinitives, coordinations

- *The board approved [its acquisition] [by Royal Trustco Ltd.] [of Toronto] [for $27 a share] [at its monthly meeting]*

Coordination scope ambiguity

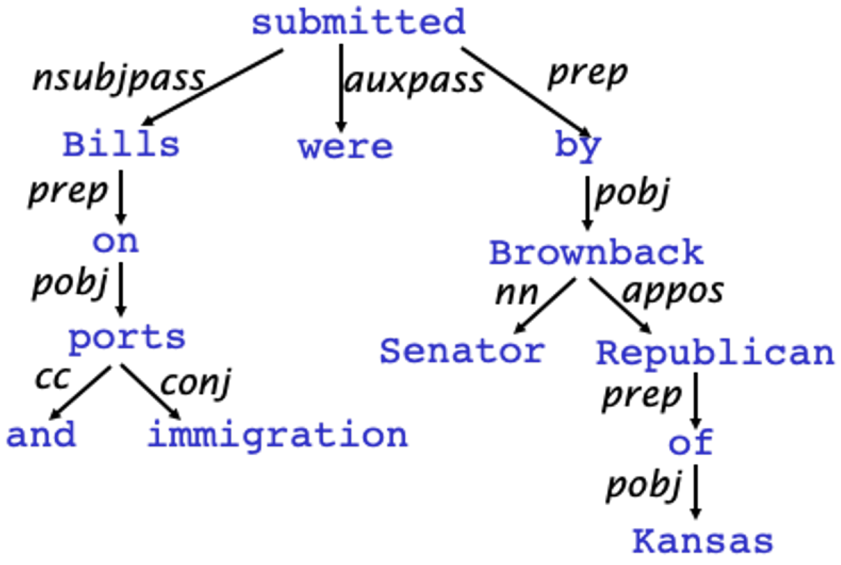
- A type of syntactic ambiguity that arises in sentences containing coordinate structures.
- Coordinate structures are sentences or phrases where two or more elements of similar syntactic type are joined together using coordinating conjunctions like "and," "or," or "but."
- The ambiguity occurs when it's not clear which elements are being coordinated with each other.
- An example: "Sue and John like apples and oranges."

Dependency grammar and structure

- Dependency syntax assumes that syntactic structure consists of lexical items, normally binary asymmetric relations ("arrows") called dependencies

- The arrows are commonly typed with the name of grammatical relations (subject, prepositional object, apposition, etc.)

- Usually, dependencies form a tree (connected, acyclic, single-head)



Dependency Grammar and Dependency Structure

- The idea of dependency structure goes back a long way.
- Some people draw the arrows one way; some the other way!
  - In modern dependency work, people had them point from head to dependent – we follow that convention
  - The dependent modifies the head.
- We usually add a fake ROOT so every word is a dependent of precisely 1 other node

The rise of annotated data and Universal Dependencies treebanks

The Penn Treebank

```
( (S
        (NP-SBJ (DT The) (NN move))
        (VP (VBD followed)
        (NP
        (NP (DT a) (NN round))
        (PP (IN of)
        (NP
        (NP (JJ similar) (NNS increases))
        (PP (IN by)
        (NP (JJ other) (NNS lenders)))
        (PP (IN against)
        (NP (NNP Arizona) (JJ real) (NN estate) (NNS loans))))))
        (, ,)
        (S-ADV
        (NP-SBJ (-NONE- *))
        (VP (VBG reflecting)
        (NP
        (NP (DT a) (VBG continuing) (NN decline))
        (PP-LOC (IN in)
        (NP (DT that) (NN market)))))))
    (. .)))
```

Marcus et al. 1993, Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics. https://aclweb.org/anthology/J93-2004

The rise of annotated data

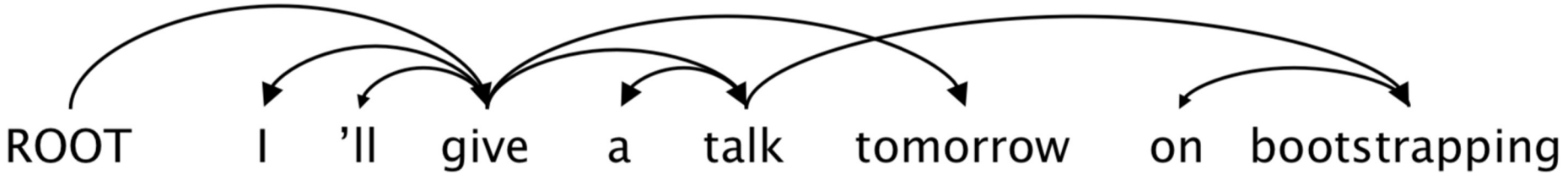
- Starting off, building a treebank seems a lot slower and less useful than building a grammar

- But a treebank gives us many things
  - Reusability of the labor
    - Many parsers, POS taggers, etc. can be built on it
    - Valuable resource for linguistics
  - Broad coverage
  - Frequencies and distributional information
  - A way to evaluate systems

Marcus et al. 1993, Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics. https://aclweb.org/anthology/J93-2004

Dependency parsing

- The process to analyze the grammatical structure in a sentence and find out related words as well as the type of the relationship between them.
- Usually some constraints:
  - Only one word is a dependent of ROOT
  - Don't want cycles A->B,B->A
- This makes the dependencies a tree.
- Final issue is whether arrows can cross (non-projective) or not



Projectivity

- Definition of a projective parse: There are no crossing dependency arcs when the words are laid out in their linear order, with all arcs above the words.
- But dependency theory normally does allow non-projective structures to account for complex sentence structures.

Greedy transition-based parsing

[Nivre 2003]

- A simple form of greedy discriminative dependency parser
- The parser does a sequence of bottom-up actions
  - Roughly like “shift” or “reduce” in a shift-reduce parser, but the "reduce" actions are specialized to create dependencies with head on left or right
- The parser has:
  - a stack σ, written with top to the right: which starts with the ROOT symbol
  - a buffer β, written with top to the left: which starts with the input sentence
  - a set of dependency arcs A: which starts off empty
  - a set of actions

Basic transition-based dependency parser

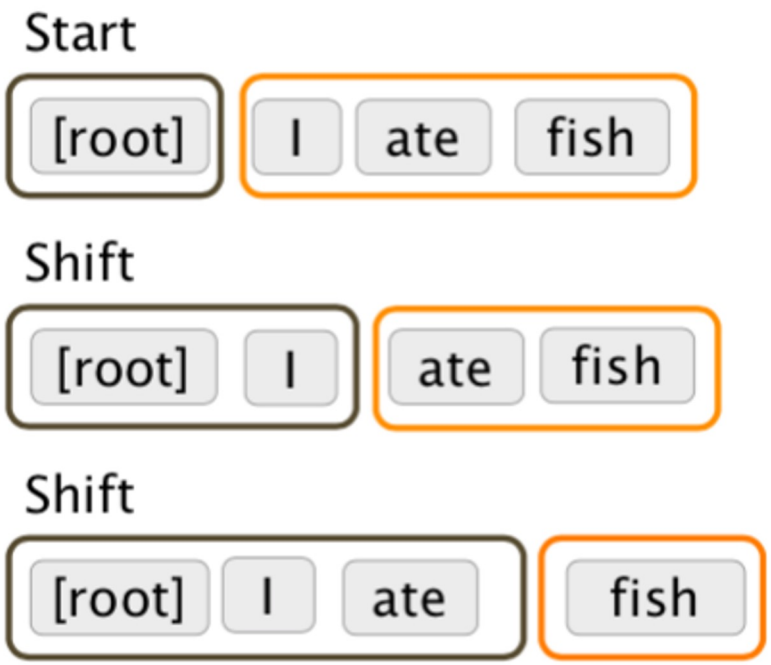
- Start σ = [ROOT], $\beta = w_1,\ldots,w_n$, $A = \emptyset$

1. Shift $\sigma, w_i \mid \beta, A \rightarrow \sigma \mid w_i, \beta, A$
2. Left-Arc$_r$ $\sigma \mid w_i \mid w_j, \beta, A \rightarrow \sigma \mid w_j, \beta, A \cup \{ r(w_j, w_i)\}$
3. Right-Arc$_r$ $\sigma \mid w_i \mid w_j, \beta, A \rightarrow \sigma \mid w_i, \beta, A \cup \{ r(w_i, w_j)\}$

- Finish: σ = [ROOT], $\beta = \emptyset$

Arc-standard transition-based parser

Analysis of "I ate fish"



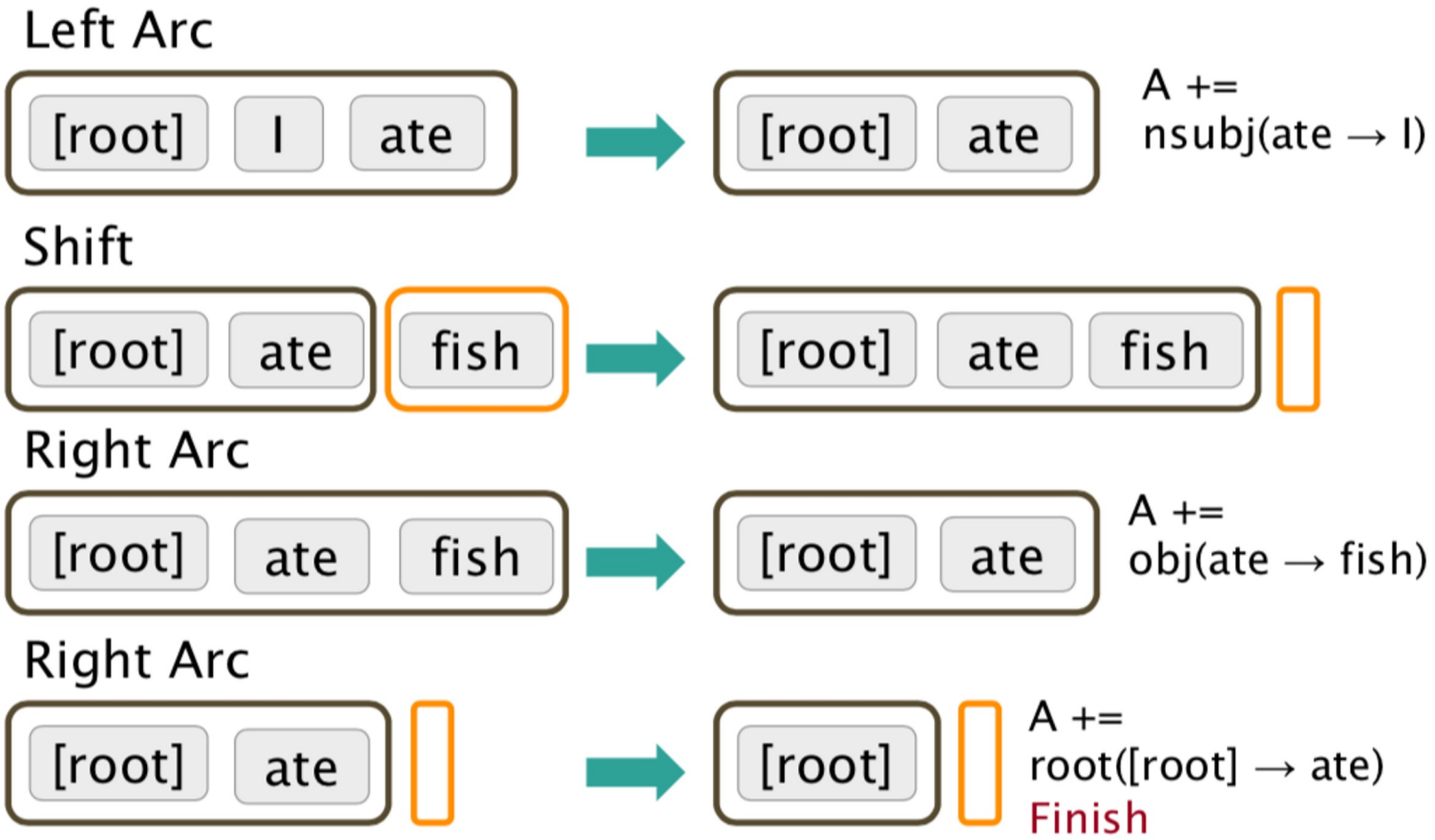
Start: $\sigma = [ROOT], \beta = w_1, \ldots, w_n, A = \emptyset$

1. Shift $\sigma, w_i|\beta, A \rightarrow \sigma|w_i, \beta, A$
2. Left-Arc$_r$ $\sigma|w_i|w_j, \beta, A \rightarrow \sigma|w_j, \beta, A \cup \{r(w_j, w_i)\}$
3. Right-Arc$_r$ $\sigma|w_i|w_j, \beta, A \rightarrow \sigma|w_i, \beta, A \cup \{r(w_i, w_j)\}$

Finish: $\beta = \emptyset$

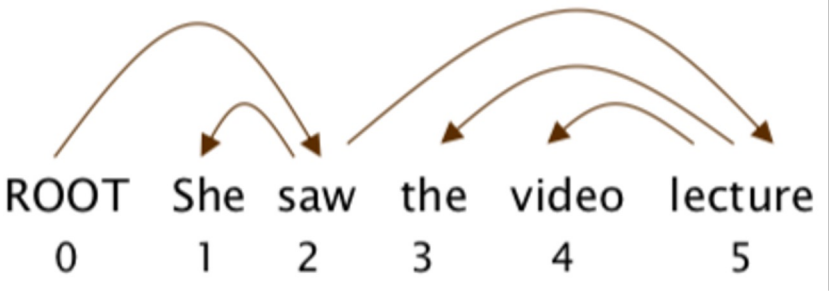
Arc-standard transition-based parser

Analysis of “I ate fish”



Evaluation of dependency parsing

(labeled) dependency accuracy



$$\text{Acc} = \frac{\text{\# correct deps}}{\text{\# of deps}}$$

UAS = 4 / 5 = 80%

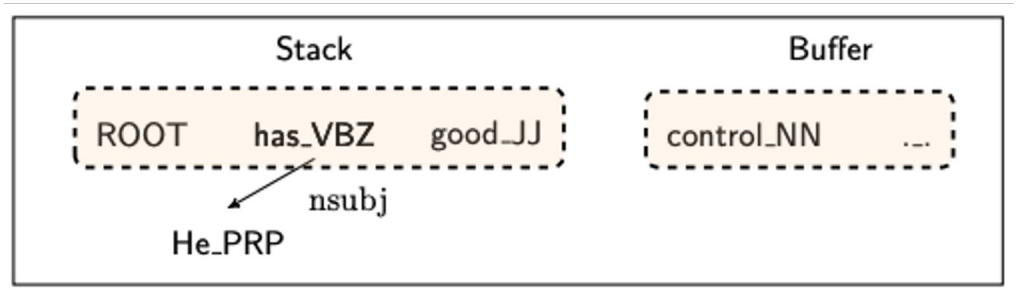
LAS = 2 / 5 = 40%

Gold			
1	2	She	nsubj
2	0	saw	root
3	5	the	det
4	5	video	nn
5	2	lecture	obj

Parsed			
1	2	She	nsubj
2	0	saw	root
3	4	the	nn
4	5	video	nsubj
5	2	lecture	ccomp

UAS: unlabeled attachment score
LAS: labeled attachment score

Conventional feature representation



binary, sparse
dim = $10^6 \sim 10^7$

Feature templates: usually a combination of 1 ~ 3 elements from the configuration.

Motivation of a neural dependency parser [Chen and Manning 2014]

- Problem 1: sparse
- Problem 2: incomplete
- Problem 3: expensive computation
- More than 95% of parsing time is consumed by feature computation
- Neural approach:
  - Learn a dense and compact feature representation
- English parsing to Stanford dependencies:
  - UAS = head
  - LAS = head and label

Parser	UAS	LAS	sent. / s
MaltParser	89.8	87.2	469
MSTParser	91.4	88.1	10
TurboParser	**92.3**	89.6	8
C & M 2014	92.0	**89.7**	**654**

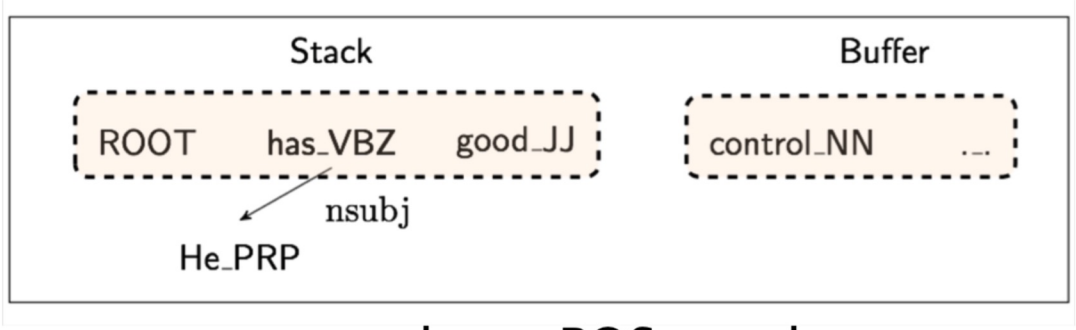
Distributed representations

- Represent each word as a d-dimensional dense vector (i.e., word embedding)
  - Similar words are expected to have close vectors.
- Meanwhile, part-of-speech tags (POS) and dependency labels are also represented as d-dimensional vectors.
  - The smaller discrete sets also exhibit many semantical similarities.

❑ NNS (plural noun) should be close to NN(singular noun)
❑ num (numerical modifier) should be close to amod (adjective modifier)

Extracting Tokens and then vector representations from configuration

- Extract a set of tokens based on the stack/buffer positions



		word		POS		dep.
s_1		good		JJ		∅
s_2		has		VBZ		∅
b_1		control		NN		∅
$lc(s_1)$	→	∅	+	∅	+	∅
$rc(s_1)$		∅		∅		∅
$lc(s_2)$		He		PRP		nsubj
$rc(s_2)$		∅		∅		∅

- We convert them to vector embeddings and concatenate them to get the neural representation of a configuration.

Model architecture

A simple feed-forward neural network multi-class classifier

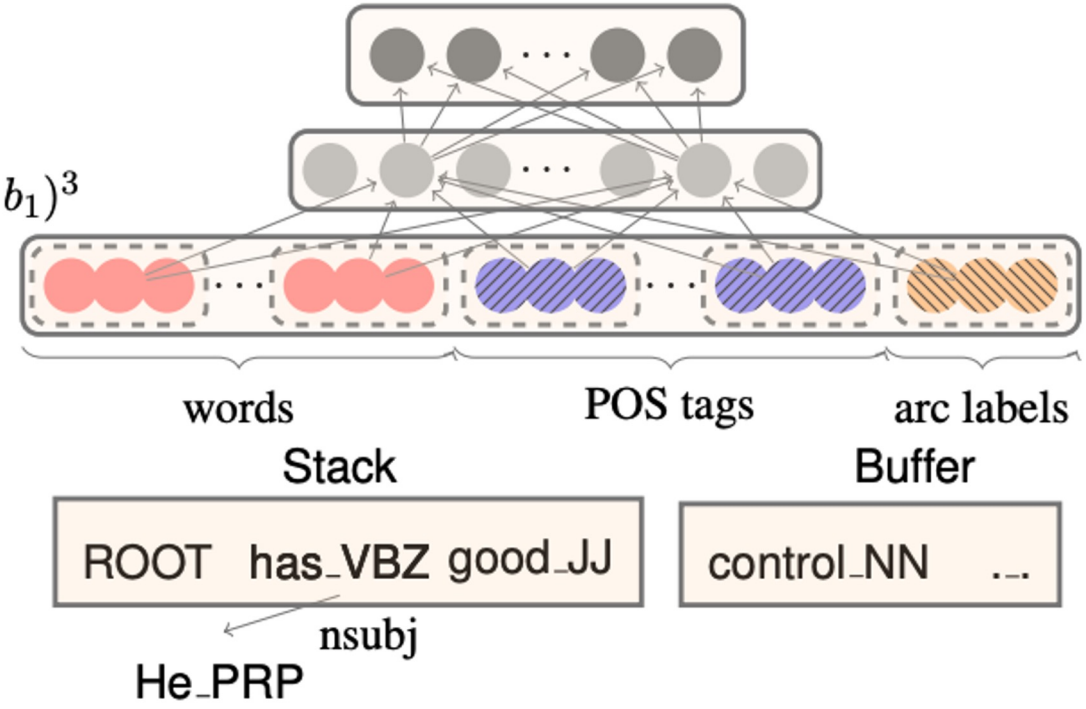
Softmax layer:

$$p = \texttt{softmax}(W_2 h)$$

Hidden layer:

$$h = (W_1^w x^w + W_1^t x^t + W_1^l x^l + b_1)^3$$

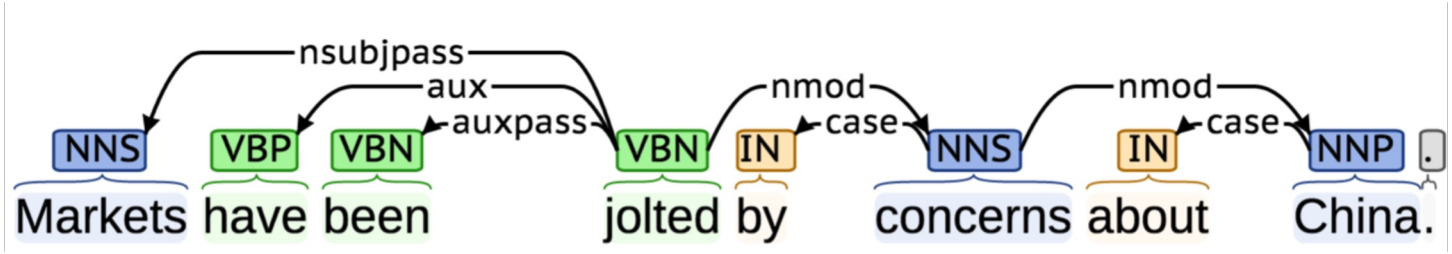
Input layer: $[x^w, x^t, x^l]$



Chen and Manning. EMNLP 2014. https://aclweb.org/anthology/papers/D/D14/D14-1082/

Dependency parsing for sentence structure

Neural networks can accurately determine the structure of sentences, supporting interpretation.



- [Chen and Manning 2014] was the first simple, successful neural dependency parser.
- The dense representations (and non-linear classifier) let it outperform other greedy parsers in both accuracy and speed

Chen and Manning. EMNLP 2014. https://aclweb.org/anthology/papers/D/D14/D14-1082/

Further developments in transition-based neural dependency parsing

This work was further developed and improved by others

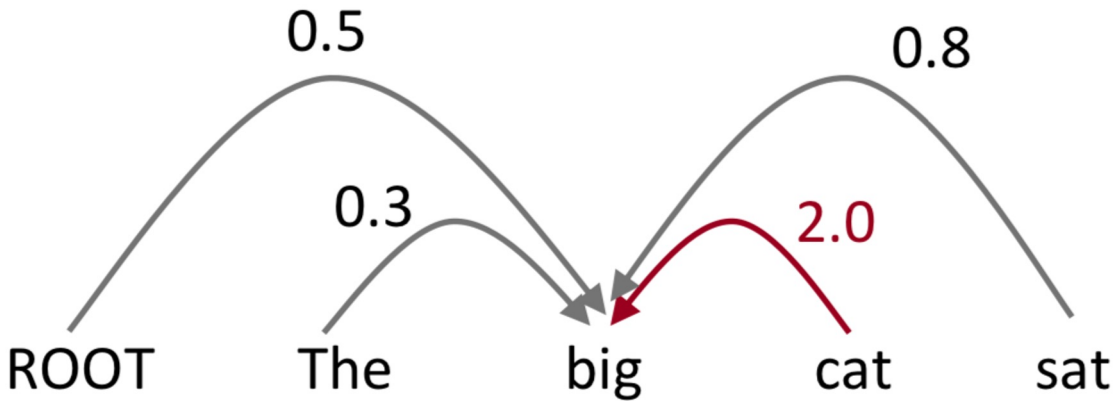
- ❑ Bigger, deeper networks with better tuned hyperparameters
- ❑ Beam search
- ❑ Global, conditional random field (CRF)-style inference over the decision sequence

Method	UAS	LAS (PTB WSJ SD 3.3)
Chen & Manning 2014	92.0	89.7
Weiss et al. 2015	93.99	92.05
Andor et al. 2016	94.61	92.79

Dependency parsing progress: http://nlpprogress.com/english/dependency_parsing.html
https://ai.googleblog.com/2016/05/announcing-syntaxnet-worlds-most.html

Graph-based dependency parsers

- Compute a score for every possible dependency for each edge
  - Then add an edge from each word to its highest-scoring candidate head
  - And repeat the same process for each other word



e.g., picking the head for “big”

A neural graph-based dependency parser

[Dozat and Manning 2017; Dozat, Qi, and Manning 2017]

- Revived graph-based dependency parsing in a neural world
  - Design a biaffine scoring model for neural dependency parsing
    - Also using a neural sequence model
- Really great results!
  - But slower than simple neural transition-based parsers
    - There are n^2 possible dependencies in a sentence of length n

Method	UAS	LAS (PTB WSJ SD 3.3
Chen & Manning 2014	92.0	89.7
Weiss et al. 2015	93.99	92.05
Andor et al. 2016	94.61	92.79
Dozat & Manning 2017	**95.74**	**94.08**

Readings

- Chen and Manning. A Fast and Accurate Dependency Parser using Neural Networks. 2014
- Globally Normalized Transition-Based Neural Networks

Reminder

- Homework 3: Nov 17
- Homework 4: to be released
- Midterm Exam: Nov 20
- Project Presentation: Dec 4 and Dec 11
- Final report and codes submission: Dec 13

Midterm Exam

- Time: 3:00-5:30 PM on November 20, 2023
- Location: Burchard 103
- Closed-book exam; one A4 cheatsheet is allowed.
- Calculator is allowed, but phones, laptops, and other devices are not allowed.
- Work on the exam independently.
- You cannot share the calculator or cheatsheet during the exam.

Midterm Exam

Coverage: Focus on the **content in Lecture 1-10.**

- L2 machine learning basics:
  - Logistic regression, how to represent the probability of both positive and negative classes; what is the loss function;
  - Three types of gradient descent methos; how to set up the learning rate; forward propagation and backpropogation
- L3 vector semantics:
  - Different types of representing the meaning of a word along with their advantages and disadvantages;
  - Word2vec, how we formulate the learning task, what is the objective function, how to optimize; two model variants of word2vec; negative sampling.

Midterm Exam

Coverage: Focus on the **content in Lecture 1-10.**

- L4 language modeling:
  - 4 stages of development of LMs;
  - N-gram model; especially uni-gram and bi-gram model; how to estimate the probability; what are the sparsity problems in n-gram model and their solutions; evaluation metrics
  - RNN: the advantages to use RNN; model architecture and optimization; backpropogation through time
- L5 More on RNNs:
  - Limitations of vanilla RNN
  - Gradient vanishing/exploding problem
  - LSTM, GRU

Midterm Exam

Coverage: Focus on the **content in Lecture 1-10.**

- L6 CNN and tokenization:
  - Three unique properties of images that are suitable for CNN
  - CNN architecture, why less parameters
  - How to use CNN on text data
  - Three levels of tokenization
- L7 seq2seq:
  - Machine translation with seq2seq model;
  - Beam search decoding
- L8 attention + transformer:
  - Why do we incorporate attention in seq2seq model
  - Barriers/problems of self-attention;
  - Transformer encoder and decoder

Midterm Exam

Coverage: Focus on the **content in Lecture 1-10.**

- L9 Pretraining:
  - Factors that affect the performance of the pre-trained model.
  - Data sources for pretraining; data preprocessing
  - Pretraining architectures
  - In-context learning, three approaches
  - Finetuning methods
- L10 Dependency parsing:
  - Dependency grammar and structure: head, dependent, Projectivity
  - Greedy transition-based parsing
  - Evaluation of dependency parsing
  - Why neural dependency parser, what kind of information we can consider to construct the features

STEVENS

INSTITUTE *of* TECHNOLOGY

THE INNOVATION UNIVERSITY®

stevens.edu

Thank You