ECE594: Mathematical Models of Language

Spring 2022

Lecture 4: Models of Meaning

Logistics

- Project proposals due February 18
- Read resources and related papers, discuss initial ideas
- Readings related to lecture---Yay or Nay
- Assignments posted this week

Project Proposal

1. Find a relevant (key) research paper for your topic

2. Write a summary of that research paper and what you took away from it as key ideas that you hope to use

3. Write what you plan to work on and how you can innovate in your final project work

4. Describe as needed

- A project plan, relevant existing literature, the kind(s) of models you will use/explore; the **data** you will use (and how it is obtained), and how you will **evaluate** success

From Words to Word Meaning

- Words as units of text
  - BoW prominent assumption
  - Feature extraction for classification
  - Alternatives to explicit feature extraction sought
- Word representation
  - Numerical representation for words
    - Embed words in a vector space
    - Permit comparing words

Words as vectors

- Sentiment analysis:
  - Feature is a word identity
    - Feature 5: 'The previous word was "terrible"'
    - requires **exact word** to be in training and test
  - With **embeddings**:
    - Feature is a word vector
    - Previous word was vector [35,22,17…]
    - In the test set we might see a similar vector [34,21,14…]
    - Generalize with **similar but unseen** words

Word-Level Models of Meaning

- Language described from 3 perspectives
  - Relations between words
  - Compositionality of how words are formed
  - Distributional properties of word co-occurrence

Word Meaning to Sentence Meaning

Starting unit: words

the, cat, cuddly, by, door

Words combine into phrases

the cuddly cat, by the door

Phrases can combine into bigger phrases

the cuddly cat by the door

Word-Level Models of Meaning

- Language described from 3 perspectives
  - Relations between words
  - Compositionality of how words are formed
  - Distributional properties of word co-occurrence

Distributional Hypothesis

Distributional hypothesis, stated by linguist John R. Firth (1957) as:

"You shall know a word by the company it keeps."

≈ "words that occur in similar contexts have similar meanings"

- One way to define "usage":

  words are defined by their environments (the words around them)

- Zellig Harris (1954):
  - **If A and B have almost identical environments we say that they are synonyms**.

Idea 1: Defining meaning by linguistic distribution

- Let's define the meaning of a word by its distribution in language use, meaning its neighboring words or grammatical environments.
  - Distributional representation of words

Context counting

(C1) A bottle of ________ is on the table

(C2) People like ________.

(C3) Don't have ________ before you drive.

(C4) ________ is made out of corn

	C1	C2	C3	C4
tesgüino	**1**	**1**	**1**	**1**
loud	0	0	0	0
Motor oil	1	0	0	1
tortillas	0	1	0	1
choices	0	1	0	0
wine	1	1	1	1

Idea 2: Meaning as a point in space (Osgood et al. 1957)

- 3 affective dimensions for a word
  - **valence**: pleasantness
  - **arousal**: intensity of emotion
  - **dominance**: the degree of control exerted

	Word	Score		Word	Score
Valence	love	1.000		toxic	0.008
	happy	1.000		nightmare	0.005
Arousal	elated	0.960		mellow	0.069
	frenzy	0.965		napping	0.046
Dominance	powerful	0.991		weak	0.045
	leadership	0.983		empty	0.081

NRC VAD Lexicon (Mohammad 2018)

- Hence the connotation of a word is a vector in 3-space

We'll discuss 2 kinds of embeddings

- Distributional
- Distributed

Distributional Embeddings

- Context counting
  - Words are represented by **counts** of nearby words (left and right context window)
  - Weighted by PPMI (positive pointwise mutual information)
  - Intuition: weigh the association between two words by asking how much more the two words co-occur in our corpus than we would have a priori expected them to appear by chance

Pointwise Mutual Information

Do outcomes x and y co-occur more than if they were independent?

$$\text{PMI}(X,Y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

PMI between two words: (Church & Hanks 1989)

Do words x and y co-occur more than if they were independent?

$$\text{PMI}(word_1, word_2) = \log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}$$

Positive Pointwise Mutual Information

- PMI ranges from $-\infty$ to $+\infty$
- But negative values are problematic
  - Things are co-occurring **less than** we expect by chance
  - Unreliable without enormous corpora
    - Imagine w1 and w2 whose probability is each 10^{-6}
    - Hard to be sure p(w1,w2) is significantly different than 10^{-12}

- Positive PMI (**PPMI**) between word1 and word2:

$$\text{PPMI}(word_1, word_2) = \max\left(\log_2 \frac{P(word_1, word_2)}{P(word_1)P(word_2)}, 0\right)$$

Distributional Embeddings

- Context counting
  - Words are represented by **counts** of nearby words (left and right context window)
  - Weighted by PPMI (positive pointwise mutual information)
  - **Sparse** vectors, dimensionality |V|

Computing PPMI on a term-context matrix

- Matrix F with W rows (words) and C columns (contexts)
- f_{ij} is # of times w_i occurs in context c_j

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \qquad p_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}} \qquad p_{*j} = \frac{\sum_{i=1}^{W} f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

	computer	data	result	pie	sugar	count(w)
cherry	2	8	9	442	25	486
strawberry	0	0	1	60	19	80
digital	1670	1683	85	5	4	3447
information	3325	3982	378	5	13	7703
count(context)	4997	5673	473	512	61	11716

$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*}p_{*j}} \qquad ppmi_{ij} = \begin{cases} pmi_{ij} & \text{if } pmi_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{W}\sum_{j=1}^{C} f_{ij}}$$

	computer	data	result	pie	sugar	count(w)
cherry	2	8	9	442	25	486
strawberry	0	0	1	60	19	80
digital	1670	1683	85	5	4	3447
information	3325	3982	378	5	13	7703
count(context)	4997	5673	473	512	61	11716

- p(w=information,c=data) = 3982/111716 = .3399
- p(w=information) = 7703/11716 = .6575
- p(c=data) = 5673/11716 = .4842

$$p(w_i) = \frac{\sum_{j=1}^{C} f_{ij}}{N} \qquad p(c_j) = \frac{\sum_{i=1}^{W} f_{ij}}{N}$$

	p(w,context)					p(w)
	computer	**data**	**result**	**pie**	**sugar**	**p(w)**
cherry	0.0002	0.0007	0.0008	0.0377	0.0021	0.0415
strawberry	0.0000	0.0000	0.0001	0.0051	0.0016	0.0068
digital	0.1425	0.1436	0.0073	0.0004	0.0003	0.2942
information	0.2838	0.3399	0.0323	0.0004	0.0011	0.6575
p(context)	0.4265	0.4842	0.0404	0.0437	0.0052	

$$pmi_{ij} = \log_2 \frac{p_{ij}}{p_{i*}p_{*j}}$$

	p(w,context)					p(w)
	computer	data	result	pie	sugar	p(w)
cherry	0.0002	0.0007	0.0008	0.0377	0.0021	0.0415
strawberry	0.0000	0.0000	0.0001	0.0051	0.0016	0.0068
digital	0.1425	0.1436	0.0073	0.0004	0.0003	0.2942
information	0.2838	0.3399	0.0323	0.0004	0.0011	0.6575
p(context)	0.4265	0.4842	0.0404	0.0437	0.0052	

- pmi(information,data) = $\log_2$ (.3399 / (.6575*.4842)) = .0944

Resulting PPMI matrix (negatives replaced by 0)

	computer	data	result	pie	sugar
cherry	0	0	0	4.38	3.30
strawberry	0	0	0	4.10	5.51
digital	0.18	0.01	0	0	0
information	0.02	0.09	0.28	0	0

Weighting PMI

- PMI is biased toward infrequent events
  - Very rare words have very high PMI values
- Two solutions:
  - Give rare words slightly higher probabilities
  - Use add-one smoothing (which has a similar effect)

Traditional Approach

- Context counting
    - Count left and right context in a window
    - Reweight with Pointwise Mutual Information
    - Reduce dimensionality with SVD or NNMF
        - Why?
        - Latent Semantic Analysis of documents [Deerwester et al. 1988]

Spare to Dense Vectors

Singular value decomposition (SVD) of PPMI weighted co-occurrence matrix

$$\underset{|V| \times |V|}{\begin{bmatrix} \\ \\ X \\ \\ \\ \end{bmatrix}} = \underset{|V| \times |V|}{\begin{bmatrix} \\ \\ W \\ \\ \\ \end{bmatrix}} \underset{|V| \times |V|}{\begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_V \end{bmatrix}} \underset{|V| \times |V|}{\begin{bmatrix} \\ \\ C \\ \\ \\ \end{bmatrix}}$$

$$\underset{|V| \times |V|}{\begin{bmatrix} \\ \\ X \\ \\ \\ \end{bmatrix}} = \underset{|V| \times k}{\begin{bmatrix} \\ \\ W \\ \\ \\ \end{bmatrix}} \underset{k \times k}{\begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_k \end{bmatrix}} \underset{k \times |V|}{\begin{bmatrix} \quad C \quad \end{bmatrix}}$$

Only keep the top k (e.g., 100) singular values!

Traditional Approach

- Context counting

Other methods include Brown clustering
hierarchical clustering based on bigram mutual information

Word Embeddings

- Context counting
- Prediction-based
  - Vector space representations learned on unlabeled linear context (i.e., left/right words)
  - **Dense** vectors

SENNA [Collobert and Weston, 2008; Collobert et al., 2011]: Multi-layer DNN w/ ranking-loss objective; BoW and sentence-level feature layers, followed by std. NN layers. Similar to [Bengio et al., 2003].

Distributed Embeddings

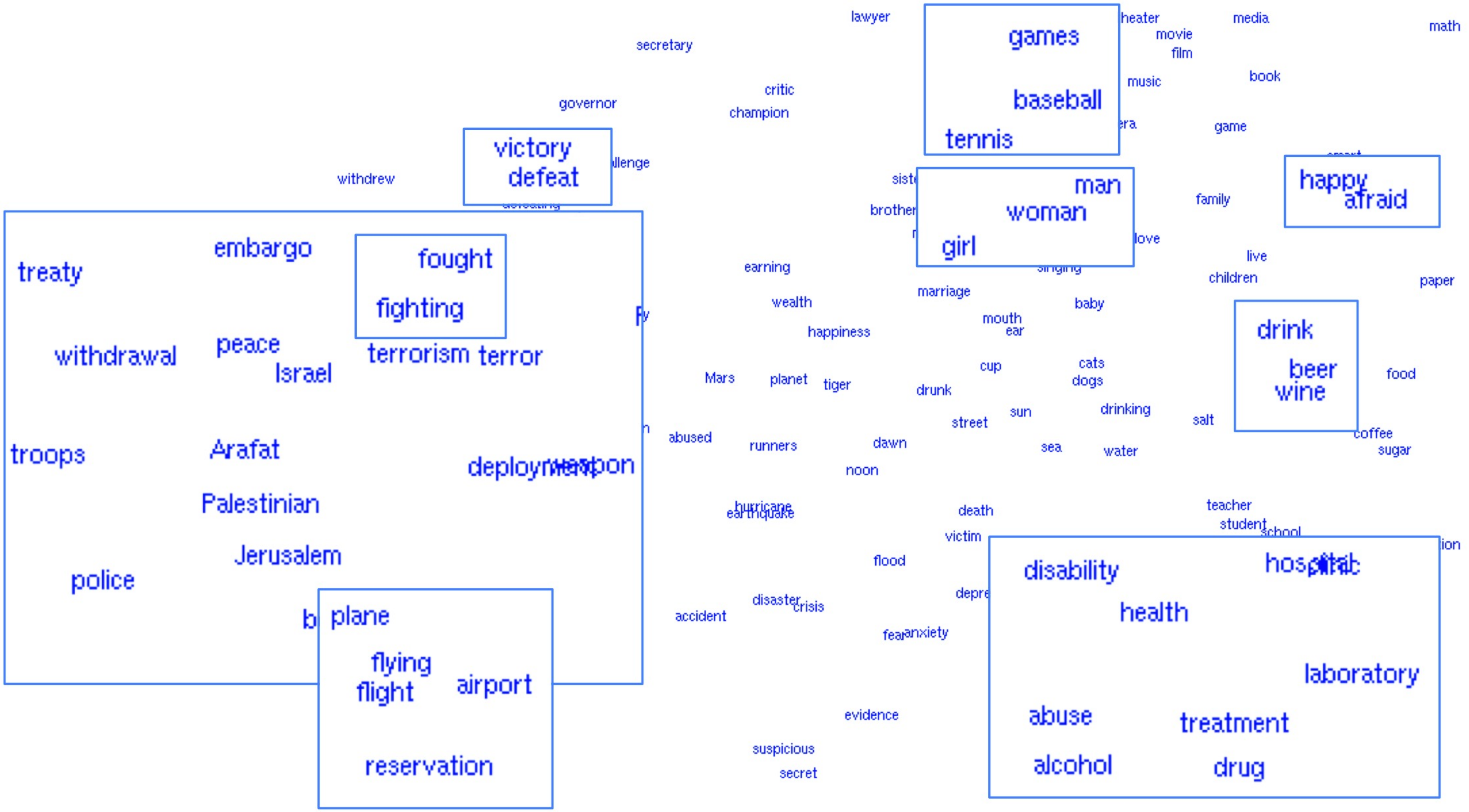
- Context counting
- Prediction-based
  - Vector space representations learned on unlabeled linear context (i.e., left/right words)
  - Representation created by training a classifier to **predict** whether a word is likely to appear nearby
  - Breakthrough idea word2vec [Mikolov et al 2013]
    - Continuous Bag of Words idea (using context words to predict target)
    - Skip-gram (predict surrounding context words given current word)
    - Demo: https://code.google.com/p/word2vec

Distributed Embeddings

- Context counting
- Prediction-based
  - Big idea: **self-supervision**:
    - A word c that occurs near w in the corpus acts as the gold "correct answer" for supervised learning
    - No need for human labels

    [Bengio et al. (2003); Collobert et al. (2011)]

Distributed Embeddings



Other Approaches

- Canonical Correlation Analysis

  Word-context correlation [Dhillon et al., 2011, 2012]

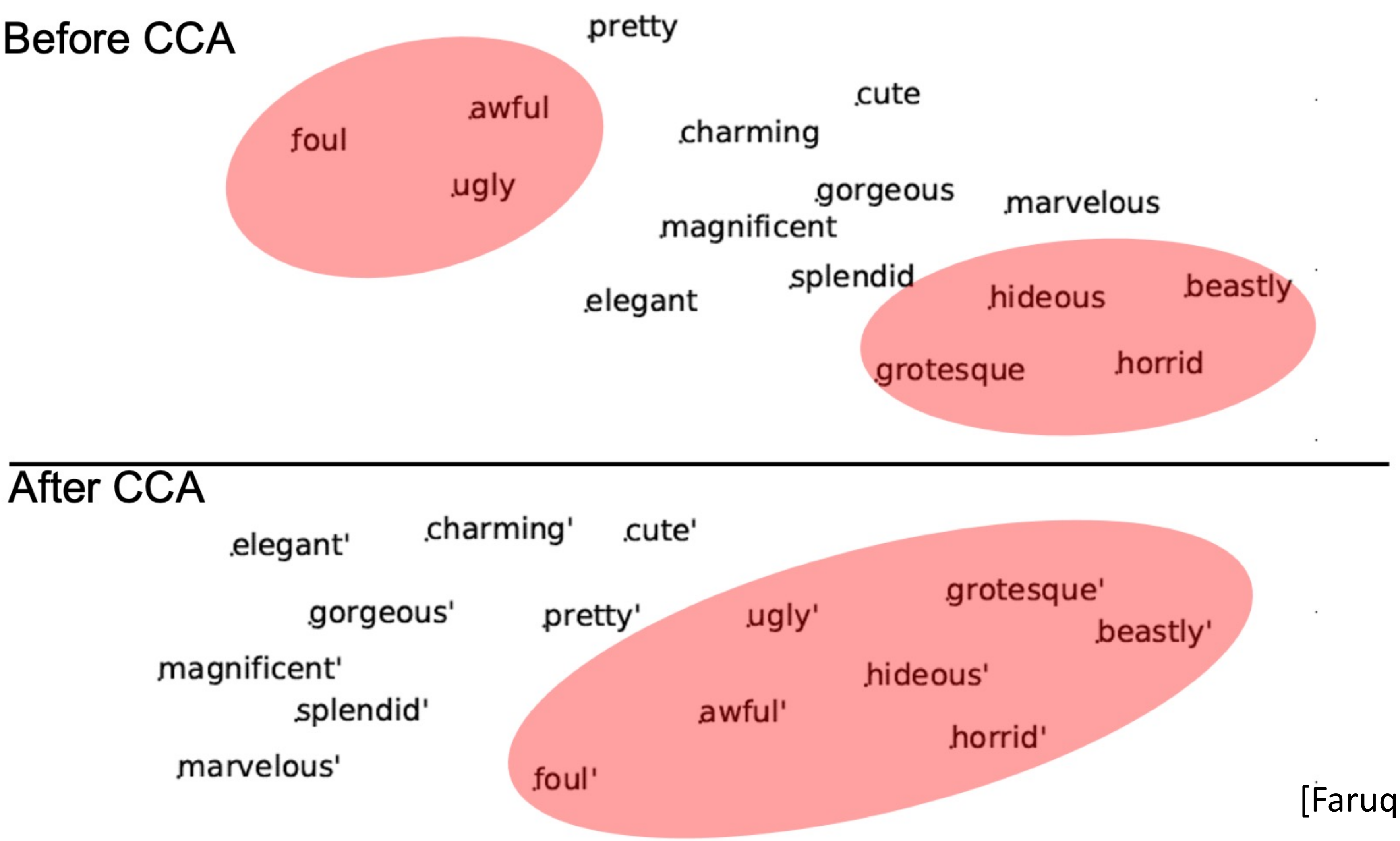
  Multilingual correlation [Faruqui and Dyer, 2014; Lu et al., 2015]

- Multi-sense embeddings [Reisinger and Mooney, 2010; Neelakantan et al., 2014]
- Task-tailored embeddings to capture specific types of similarity/semantics

  Lexicon evidence (PPDB, WordNet, FrameNet) [Xu et al., 2014; Yu and Dredze, 2014; Faruqui et al., 2014; Wieting et al., 2015]

  Combining advantages of global matrix factorization and local context window methods [GloVe; Pennington et al., 2014]

Multi-view Embeddings via CCA



[Faruqui and Dyer 2014]

Evaluation

- Extrinsic method
  - Use embeddings for a task and see if performance improves
  - Can be expensive (time) but still most important evaluation metric

  - Analogy: solve problems of the form *a:b :: a* :b**, given *a, b,* and *a*,* find *b**

Evaluation

- Measure of similarity
  - Cosine of angle between vectors --- length ignored
  - Vectors are normalized to unit length before they are used for similarity calculation, making cosine similarity and dot-product equivalent. [Levy et al., 2015]
  - Most applications of word embeddings explore not the word vectors themselves, but relations between them to solve, for example, similarity and word relation tasks. For these tasks, it was found that using normalised word vectors improves performance. [Wilson and Schakel, 2015]

Evaluation

- Importance of word length [Schakel and Wilson, 2015]
  - A word that is consistently used in a similar context will be represented by a longer vector than a word of the same frequency that is used in different contexts.
  - Not only the direction, but also the length of word vectors carries important information.
  - Word vector length furnishes, in combination with term frequency, a useful measure of word significance.

Evaluation

- Instrinsic method
  - Fast to compute, but not clear if it really helps downstream tasks
  - Similarity: compute correlation between an algorithm's word similarity scores and word similarity ratings assigned by humans.
  - WordSim-353 (Finkelstein et al., 2002) is a commonly used set of ratings from 0 to 10 for 353 noun pairs
  - (*plane*, *car*) had an average score of 5.77.
  - Analogy: solve problems of the form *a:b :: a* :b**, given *a, b,* and *a*,* find *b**

Limitation

- Variability
  - randomness in the initialization and sampling
    - word2vec may produce different results even from the same dataset, and individual documents in a collection impact the resulting embeddings [Tian et al. 2016, Hellrich and Hahn 2016, Antoniak and Mimno 2018]
    - Best practice to train multiple embeddings with bootstrap sampling over documents and average the results [Antoniak and Mimno, 2018]

Sentence Structure for Sentence Meaning

Humans communicate complex ideas by composing words together into bigger units to convey complex meanings

Listeners need to work out what modifies [attaches to] what

A model needs to understand sentence structure in order to be able to interpret language correctly

Models of composition

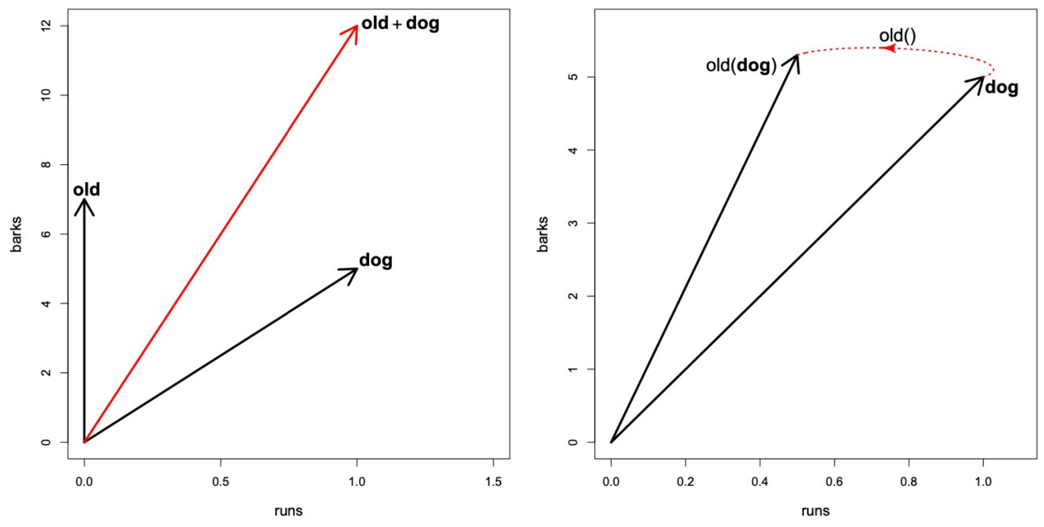
- Initial approaches
  - Point-wise sum, tensor product [Mitchell and Lapata, 2010; Smolensky 1990]

    - Worked well for adjective-noun and noun-noun phrases

    - Fail to capture structural differences
      - Lice on dogs; lice and fleas
    - Fails on recursion
      - nice toilette-trained spayed short-haired Siamese cat

Models of composition

- Initial approaches
  - Matrix-vector compositionality [Baroni and Zamparelli, 2010; Zanzotto et al., 2010; Grefenstette and Sadrzadeh, 2011; Socher et al., 2011; Yessenalina and Cardie, 2011]
    - content words (such as nouns) are vectors
    - functional words (such as d
      expressions of one type ont
      other types.



Sentence Structure for Sentence Meaning

Humans communicate complex ideas by composing words together into bigger units to convey complex meanings

Listeners need to work out what modifies [attaches to] what

A model needs to understand sentence structure in order to be able to interpret language correctly

Two Views of Linguistic Structure

- Sentence interpreted via Constituency structure
  - Sets of rules of how words are grouped to form phrases

- Sentence represented as a Dependency structure
  - shows which words depend on (modify, attach to, or are arguments of) which other words

Constituency Structure

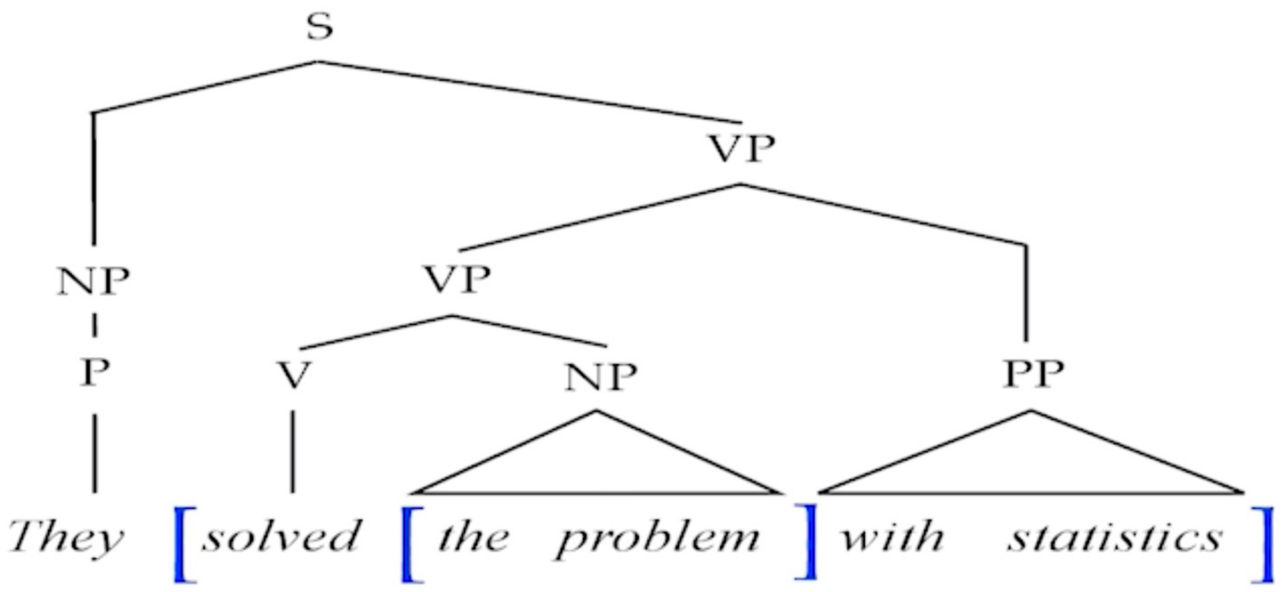
- A sentence as a set of constituents

- Sentence interpreted via Constituency Grammars
  - Sets of rules of how words are grouped to form phrases
  - Context-Free Grammars (CFG)
  - Popularized by Noam Chomsky

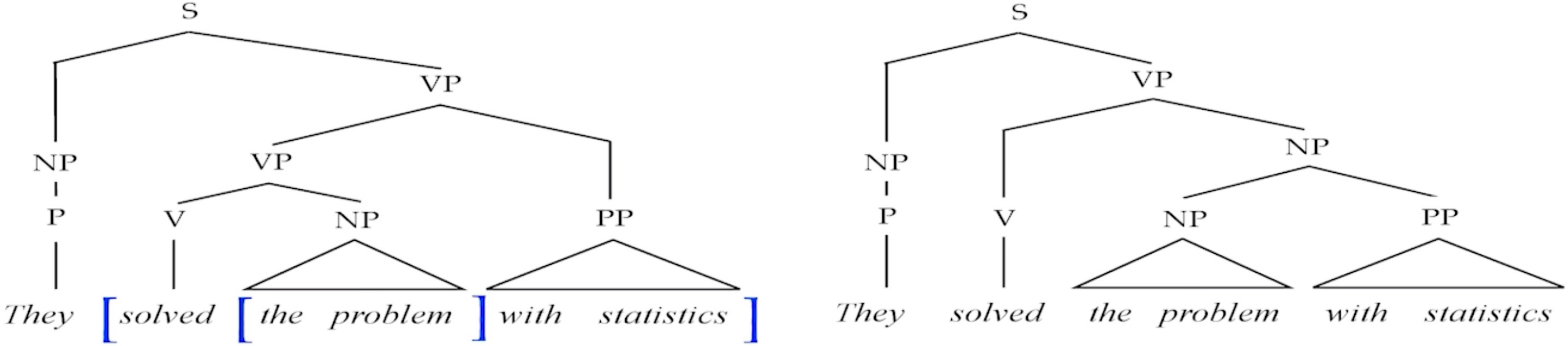
Constituency Parsing

- A sentence as a set of constituents
- **Constituency** parsing: task of recognizing a sentence and assigning a **constituency** structure to it

  - NP -> det N
  - VP ->V NP
  - PP -> prep NN
  - VP -> VP PP
  - S -> NP VP



Constituency Parsing

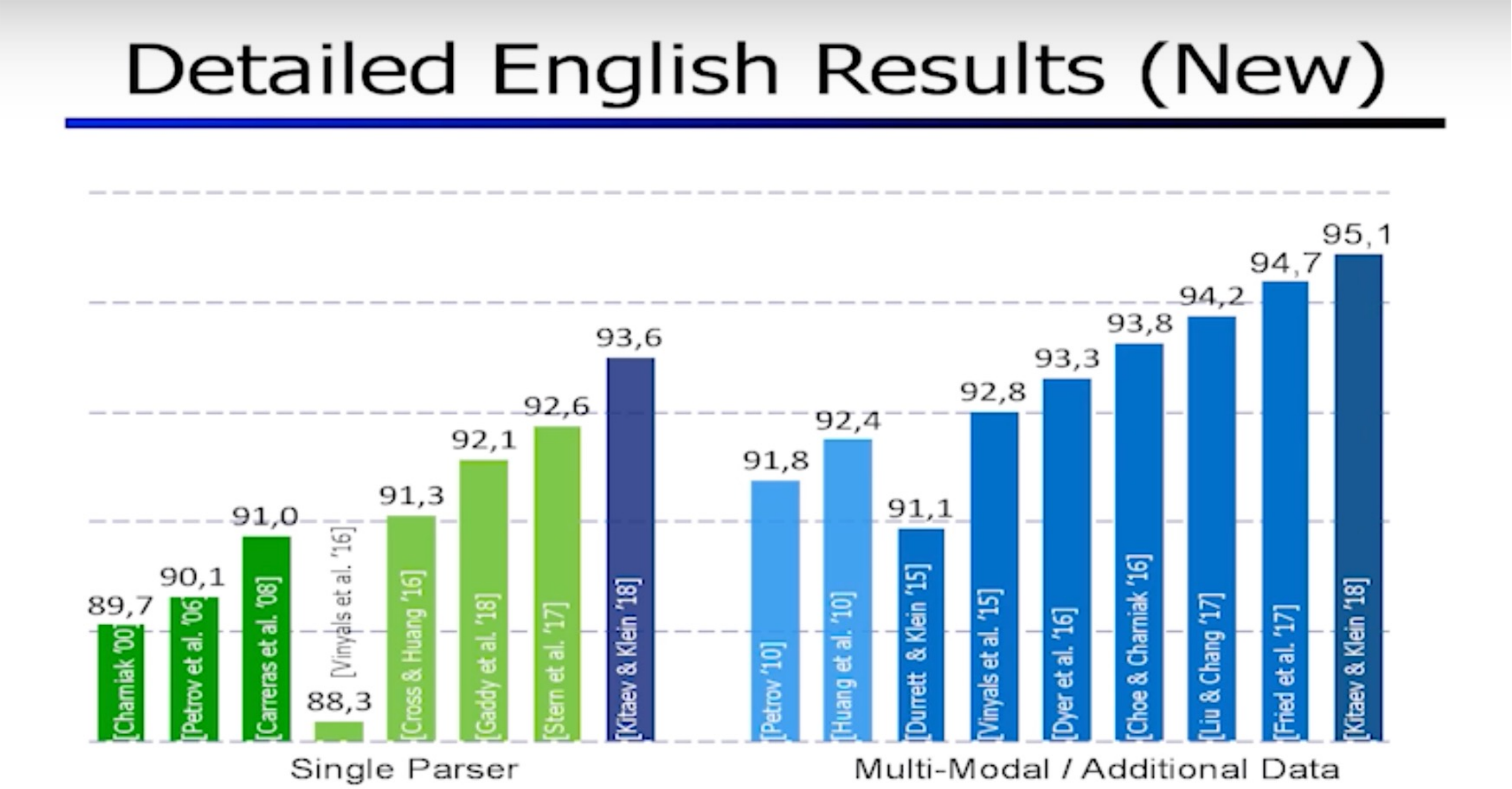


Why Constituency-based structures?

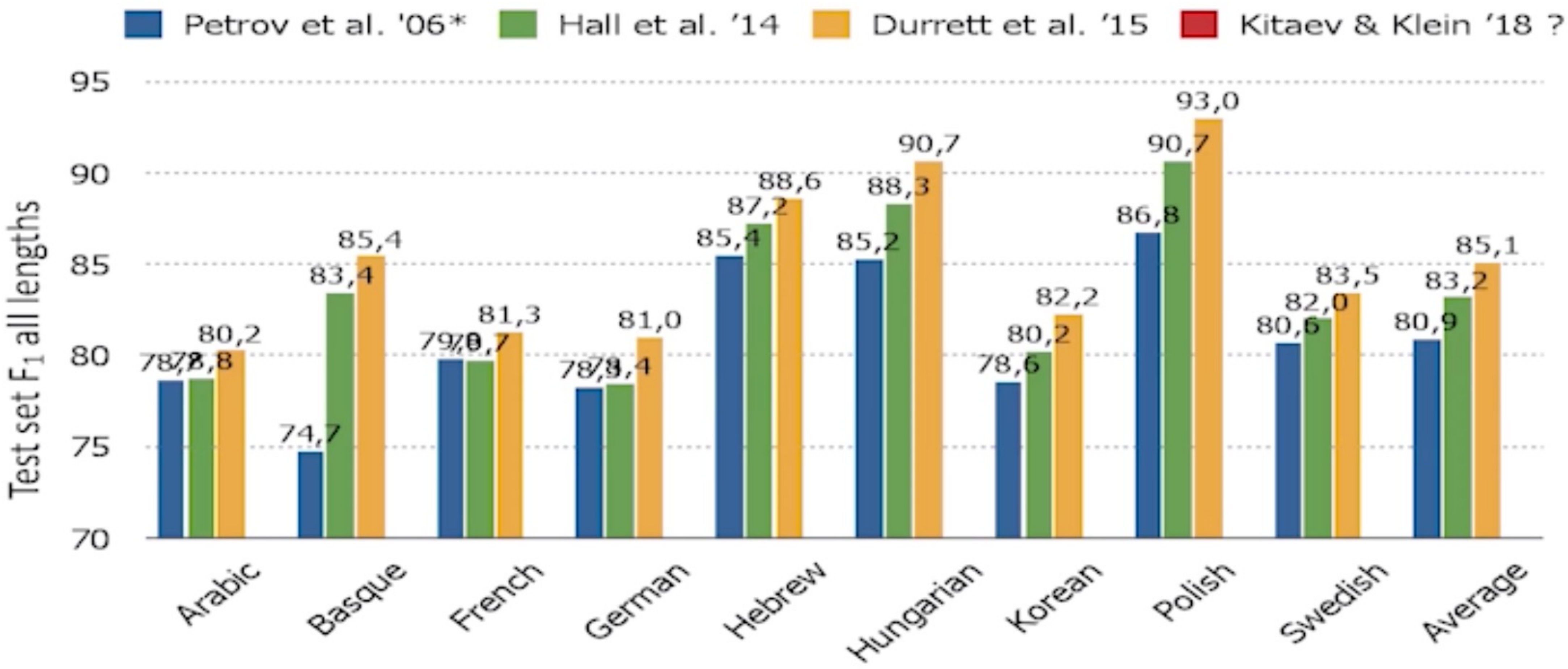
- Typically useful for fixed word-order languages
  - Grammar checking : If a sentence can’t be parsed, it may have grammatical errors (or at least hard to read)
  - Intermediate representations
    - Syntax-based understanding
    - MT (pre-NMT) & Low-resource MT
    - Information Extraction

- Several parsers (late1980- early 2000s)
  - CKY parser
  - Earley parser
  - Now NN-based parsers

Performance



Multi-Lingual Results



Treebanks

- Corpus of sentences with parse trees
  - Penn Treebank

```
((S
   (NP-SBJ (DT That)
     (JJ cold) (, ,)
     (JJ empty) (NN sky) )
   (VP (VBD was)
     (ADJP-PRD (JJ full)
       (PP (IN of)
         (NP (NN fire)
           (CC and)
           (NN light) ))))
   (. .) ))
```

(a)

```
((S
   (NP-SBJ The/DT flight/NN )
   (VP should/MD
     (VP arrive/VB
       (PP-TMP at/IN
         (NP eleven/CD a.m/RB ))
       (NP-TMP tomorrow/NN )))))
```

(b)

Figure 12.7 Parsed sentences from the LDC Treebank3 version of the (a) Brown and (b) ATIS corpora.

Treebanks

- Robust grammars
  - Context-free grammar rules

$$
\begin{aligned}
VP &\rightarrow VBD\ PP \\
VP &\rightarrow VBD\ PP\ PP \\
VP &\rightarrow VBD\ PP\ PP\ PP \\
VP &\rightarrow VBD\ PP\ PP\ PP\ PP \\
VP &\rightarrow VB\ ADVP\ PP \\
VP &\rightarrow VB\ PP\ ADVP \\
VP &\rightarrow ADVP\ VB\ PP
\end{aligned}
$$

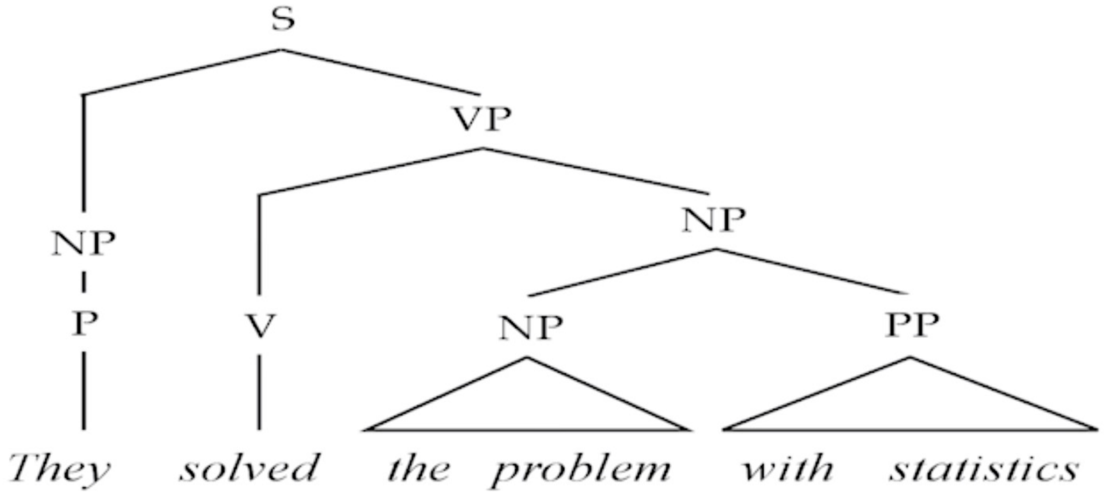
~4500 rules for VP

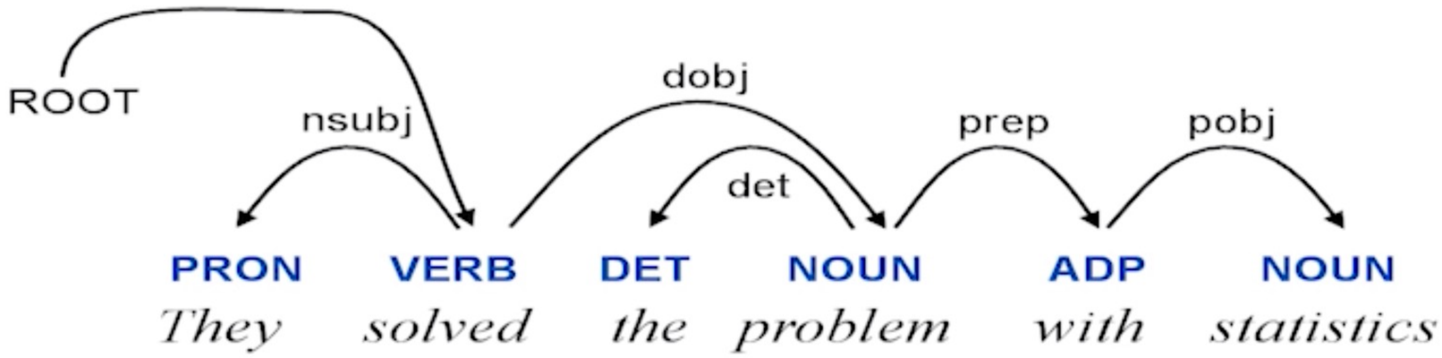
Two Views of Linguistic Structure

- Sentence interpreted via Constituency structure
  - Sets of rules of how words are grouped to form phrases

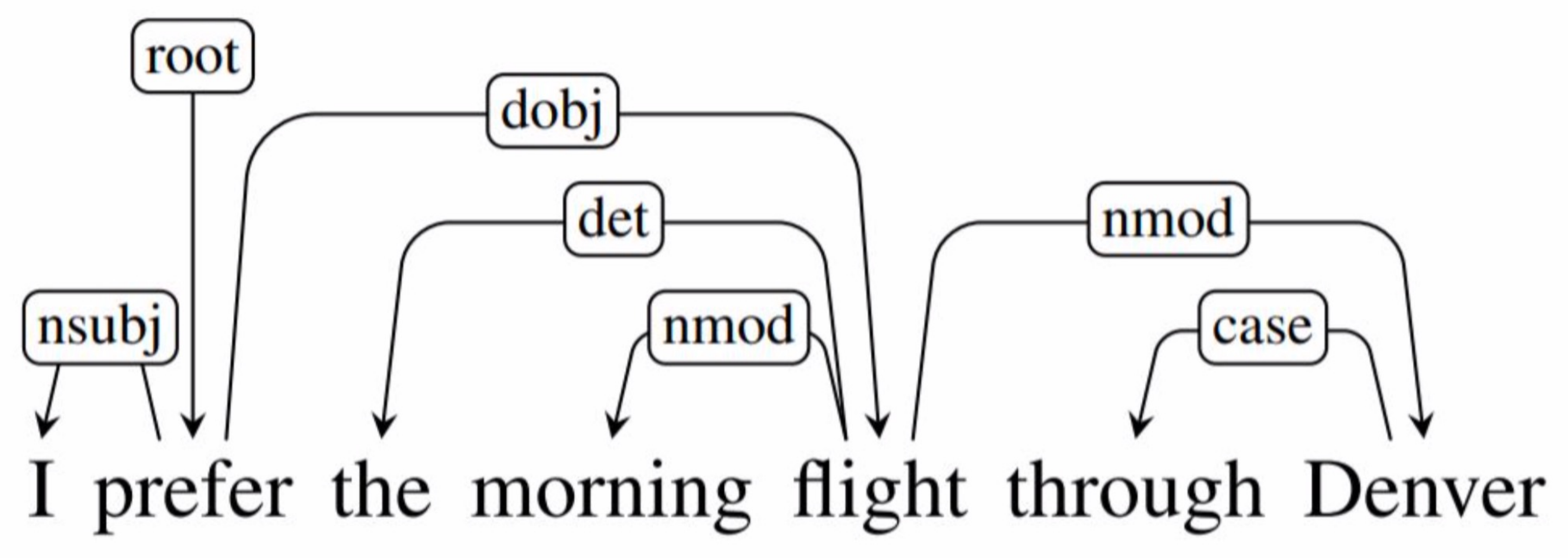
- Sentence represented as a Dependency structure
  - shows which words depend on (modify, attach to, or are arguments of) which other words

Two Views of Structure

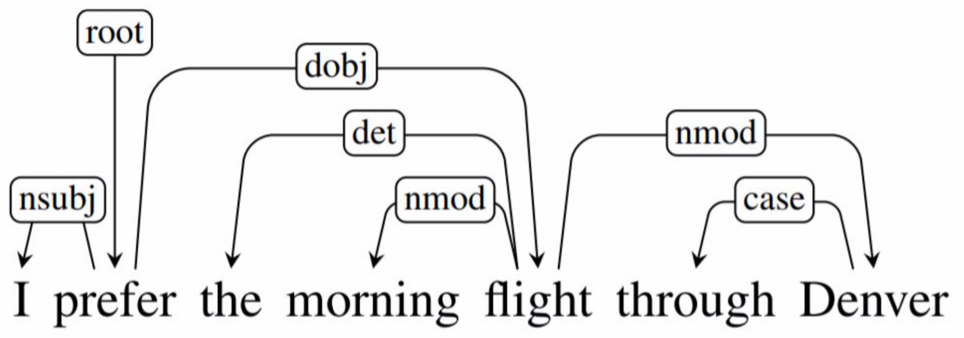




Dependency Structure

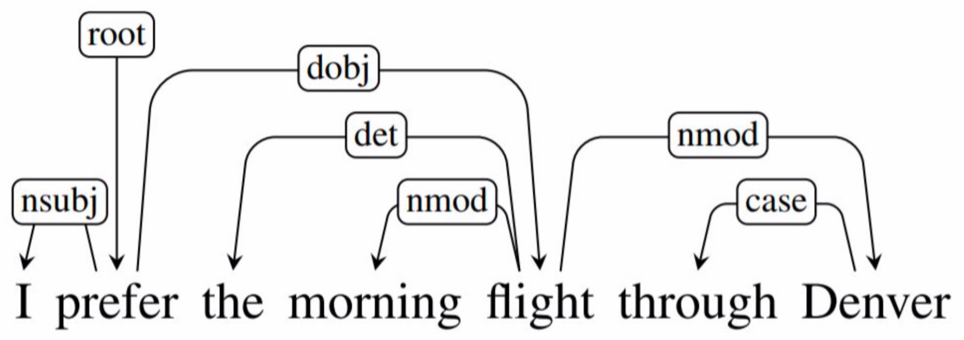


Dependency Structure



- Relations between words
  - binary, asymmetric
  - subject, prepositional object, apposition
- Denoted via arrows with labels
  - Arrow connects a head with a dependent
- Dependencies form a connected, acyclic, single-root graph

Dependency Parsing



- Syntactic parsing: task of recognizing a sentence and assigning a structure to it.
- **Dependency** parsing: the task of recognizing a sentence and assigning a **dependency** structure to it.

Pāṇini’s grammar (c. 5th century BCE)

Gallery: http://wellcomeimages.org/indexplus/image/L0032691.html
[CC BY 4.0](CC BY 4.0) File:Birch bark MS from Kashmir of the Rupavatra Wellcome L0032691.jpg

Dependency Grammars

- Characteristics
  - Lexical/syntactic dependencies between words
  - The top-level predicate of a sentence is the root
  - Simpler to parse than context-free grammars
  - Particularly useful for free word order languages

Techniques (1)

- Dynamic programming
  - CKY – similar to lexicalized PCFG, cubic complexity (Eisner 96)

Techniques (2)

- Constraint-based methods
  - Maruyama 1990, Karlsson 1990
  - Example
    - word(pos(x)) = DET ⇒(label(X) = NMOD, word(mod(x)) = NN, pos(x) < mod(x))
    - A determiner (DET) modifies a noun (NN) on the right with the label NMOD.
  - NP complete problem; heuristics needed
- Constraint graph
  - For initial constraint graph using a core grammar: nodes, domains, constraints
  - Find an assignment that doesn't contradict any constraints. If more than one assignment exists, add more constraints.

Techniques (3)

- Deterministic parsing
  - Covington 2001
  - MaltParser by Nivre
    - shift/reduce as in a shift/reduce parser
    - reduce creates dependencies with the head on either the left or the right
- Graph-based methods
  - Maximum spanning trees (MST)
    - MST Parser by McDonald et al.

MaltParser (Nivre 2008)

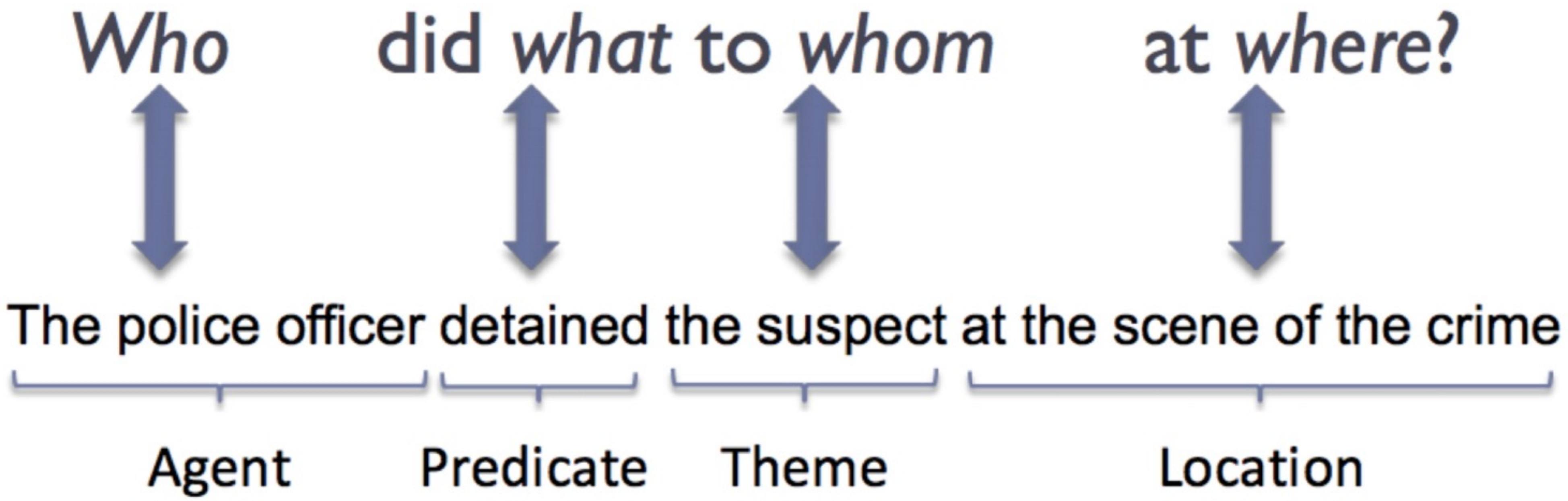
- Very similar to shift-reduce parsing.
- It includes the following components
  - A stack
  - A buffer
  - Set of dependencies (arcs)
- The reduce operations combine an element from the stack and one from the buffer
- Arc-eager parser
  - The actions are shift, reduce, left-arc, right-arc

Universal Dependencies

Universal Dependencies (UD) is a framework for consistent annotation of grammar (parts of speech, morphological features, and syntactic dependencies) across different human languages. UD is an open community effort with over 300 contributors producing nearly 200 treebanks in over 100 languages. If you're new to UD, you should start by reading the first part of the Short Introduction and then browsing the annotation guidelines.

- Universal dependencies defined
  - Same annotation standard
  - Applicable to wider set of languages, including free word-order
  - Efficient parsing algorithms
  - Useful in applications including information extraction

Semantic Role Labeling



Semantic Roles

Thematic Role	Definition
AGENT	The volitional causer of an event
EXPERIENCER	The experiencer of an event
FORCE	The non-volitional causer of the event
THEME	The participant most directly affected by an event
RESULT	The end product of an event
CONTENT	The proposition or content of a propositional event
INSTRUMENT	An instrument used in an event
BENEFICIARY	The beneficiary of an event
SOURCE	The origin of the object of a transfer event
GOAL	The destination of an object of a transfer event

Figure 19.1 Some commonly used thematic roles with their definitions.

Use of Semantic Roles

- Infer meaning components even when structurally dissimilar
    - John broke the window.
    - John broke the window with a rock.
    - The rock broke the window.
    - The window broke.
    - The window was broken by John.

Semantic Roles

- Defined with respect to the predicates and nouns
    - Semantically related verbs/nouns
    - Available in manually created resources
        - FrameNet and PropBank

FrameNet (2004)

- Project at UC Berkeley led by Chuck Fillmore for developing a database of frames, general semantic concepts with an associated set of roles.
- Roles are specific to frames, which are “invoked” by the predicate, which can be a verb, noun, adjective, adverb
  - JUDGEMENT frame
    - Invoked by: V: blame, praise, admire; N: fault, admiration
    - Roles: JUDGE, EVALUEE, and REASON
- Specific frames chosen, and then sentences that employed these frames selected from the British National Corpus and annotated by linguists for semantic roles.
- Initial version: 67 frames, 49,013 sentences, 99,232 role fillers

PropBank := proposition bank (2005)

- Project at Colorado led by Martha Palmer to add semantic roles to the Penn treebank.
- Proposition := verb + a set of roles
- Annotated over 1M words of Wall Street Journal text with existing gold-standard parse trees.
- Statistics:
  - 43,594 sentences 99,265 propositions
  - 3,324 unique verbs 262,281 role assignments

Semantic Role Labeling

- POS tagging
- Parsing
- Feature extraction

Semantic Role Labeling

- POS tagging
- Parsing
- Feature extraction

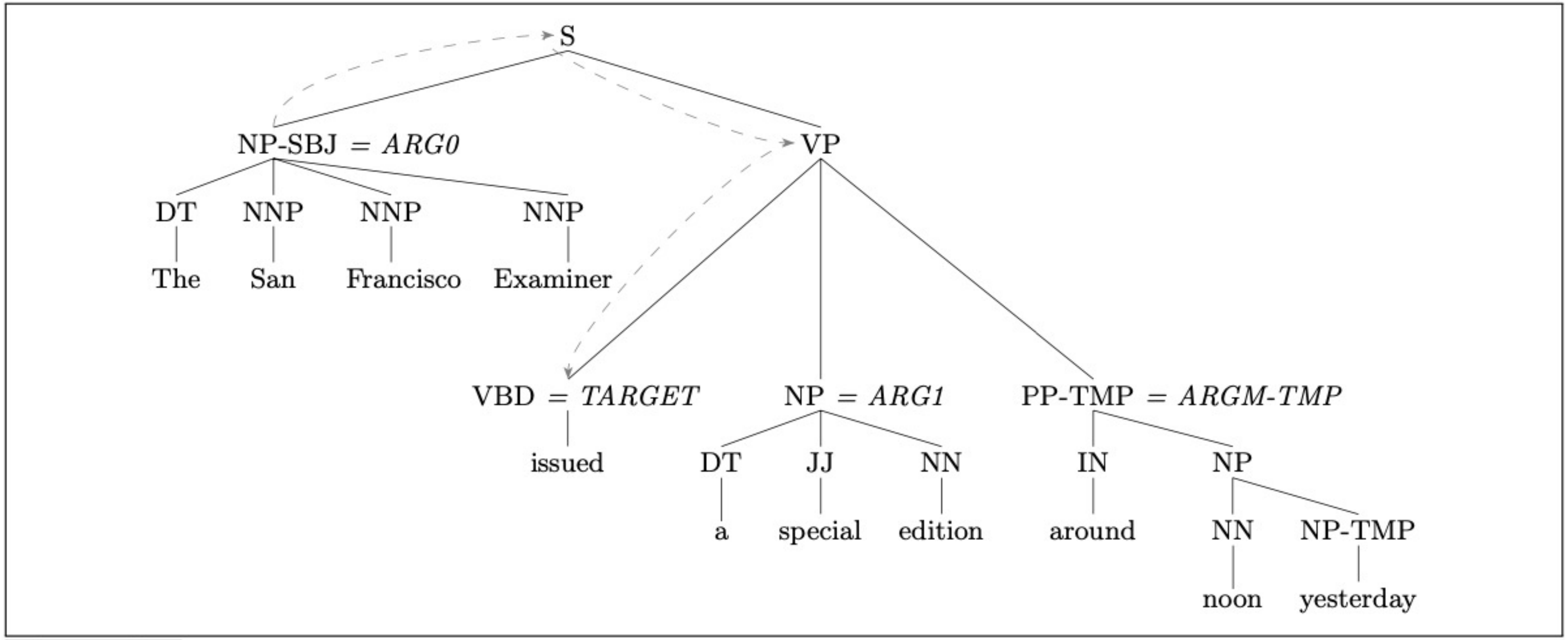


Figure 19.5 Parse tree for a PropBank sentence, showing the PropBank argument labels. The dotted line shows the **path** feature NP↑S↓VP↓VBD for ARG0, the NP-SBJ constituent *The San Francisco Examiner.*

Features for Semantic Role Labeling

- Common features:
  - Governing predicate
  - Constituent type
  - Head word of the constituent
  - Part of speech of the head word
  - Path in the parse tree from the constituent to the predicate
  - Whether the voice of the surrounding clause is active or passive
  - Whether the constituent appears before or after the predicate
  - Set of expected arguments for the verb phrase
  - Named entity type of the constituent
  - First and last word(s) of the constituent

Global Optimization

- Semantic roles are not independent of one another!
- Many approaches perform a second pass to address **global consistency**
  - Viterbi decoding
  - Reranking
  - Integer linear programming

Evaluation of Semantic Role Labeling

- **True positives:** Argument labels assigned to the correct word sequence or parse constituents
- Then, we can compute our standard NLP metrics:
  - Precision
  - Recall
  - F-measure

- Accuracy for such feature-based SRL models then highly depends on accuracy of underlying parse tree!
  - So quite high SRL results when using ground-truth parses
  - Much lower results with automatically-predicted parses!

CORE		ARGM	
F1	Acc.	F1	Acc.
92.2	80.7	89.9	71.8

CORE		ARGM	
F1	Acc.	F1	Acc.
84.1	66.5	81.4	55.6