Logistics

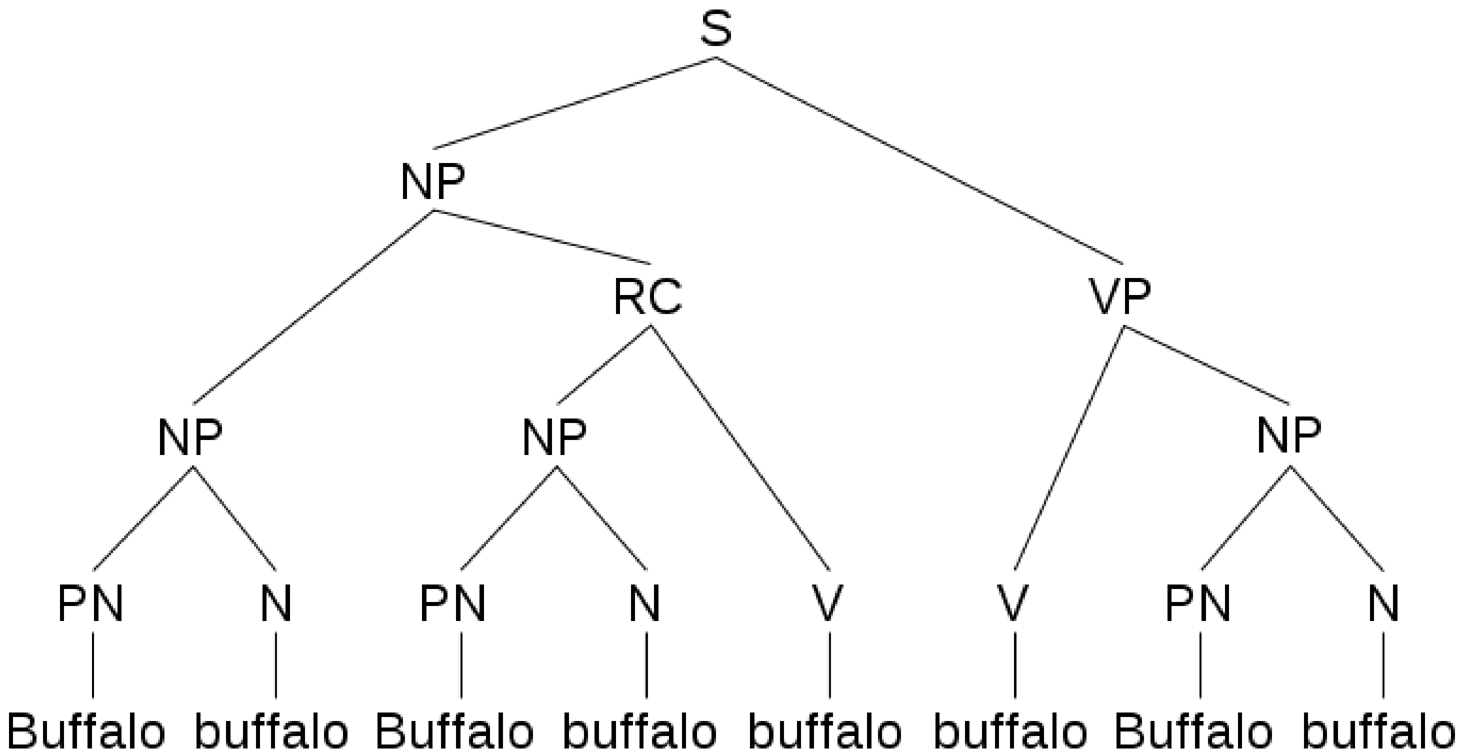
- CISE Career Development Workshop (CDW) with ~35 companies including ..
- AWS re:Invent conference live stream
  - AWS credits to be distributed (this week)
  - Setup AWS account and tutorials (next week)
  - Lab 3
- Preliminary guidelines of final project
  - More than 2 data sources (i.e., >=2)
  - Pipeline: Data collection and cleaning, alignment/integration, modeling, processing and analytics, visualization, evaluation

Review

- Basic objects that can be extracted using NLP from clinical narratives with examples and applications
- Modeling Text Documents
  - BoW, N-grams, vector, parse vector
  - Applications in IR, clustering, classification, summarization, visualization
- POS, Grammar
- Parse trees

Recursion in Grammars

"Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo".



Grammars

Its also possible to have "sentences" inside other sentences…

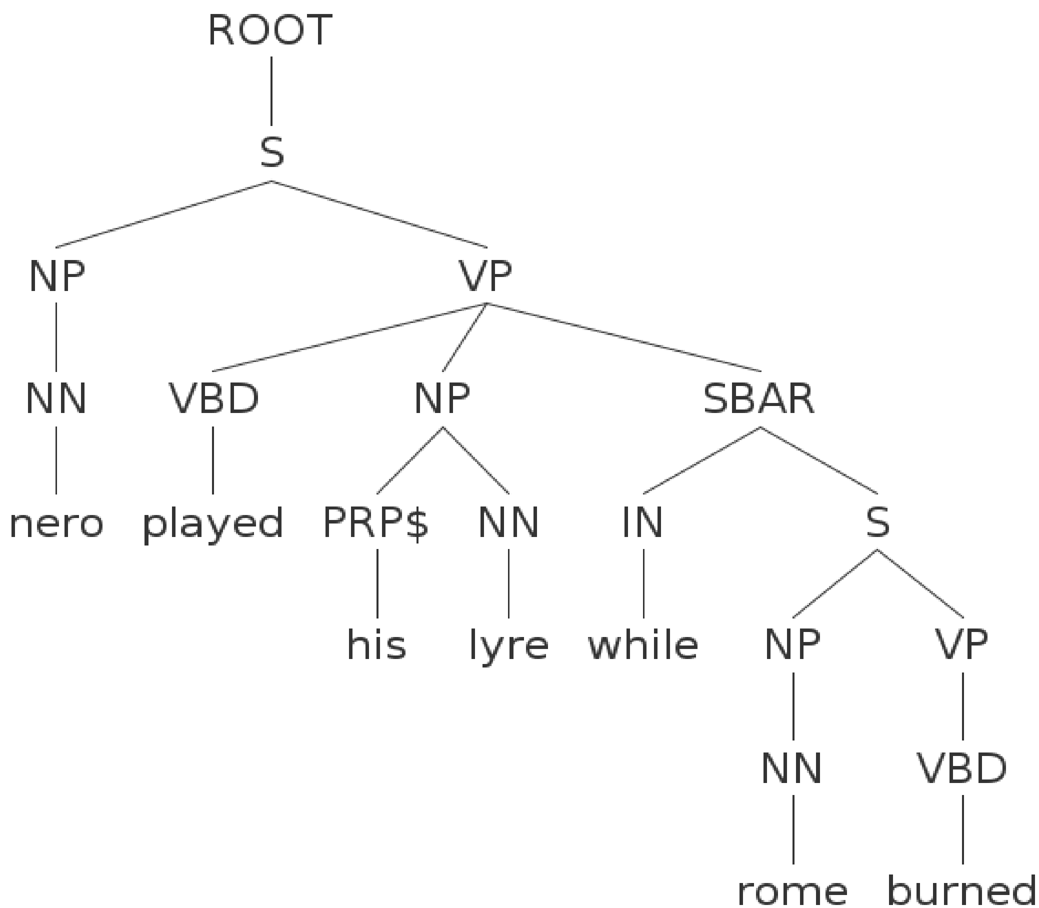
S → NP VP

VP → VB NP SBAR

SBAR → IN S

Recursion in Grammars

"Nero played his lyre while Rome burned".



Information Extraction

- POS/words (leaves in the parse tree)
- Chunking/Shallow parsing
- Parsing (state-of-the-art: PCFG)

- Named entity recognition
- Relationship extraction
- Event extraction
- Negation extraction
- Attribute extraction

PCFGs

Complex sentences can be parsed in many ways, most of which make no sense or are extremely improbable (like Groucho's example).

Probabilistic Context-Free Grammars (PCFGs) associate and learn probabilities for each rule:

S $\rightarrow$ NP VP 0.3

S $\rightarrow$ NP VP PP 0.7

The parser then tries to find the most likely sequence of productions that generate the given sentence. This adds more realistic "world knowledge" and generally gives much better results.

Most state-of-the-art parsers these days use PCFGs.

NLP Systems

- **NLTK:** Python-based NLP system. Many modules, good visualization tools, but not quite state-of-the-art performance.

- **Stanford Parser:** Another comprehensive suite of tools (also POS tagger), and state-of-the-art accuracy. Has the definitive dependency module.

- **Berkeley Parser:** Slightly higher parsing accuracy (than Stanford) but not as many modules.

- Note: high-quality parsing is usually very slow, but see: https://github.com/dlwh/puck

Outline

- Project Suggestions Overview
- N-grams
- Grammars
- Parsing
- Dependencies

Dependencies

In a constituency parse, there is no direct relation between the constituents and words from the sentence (except for leaf nodes which produce a single word).

In dependency parsing, the idea is to decompose the sentence into relations directly between words.

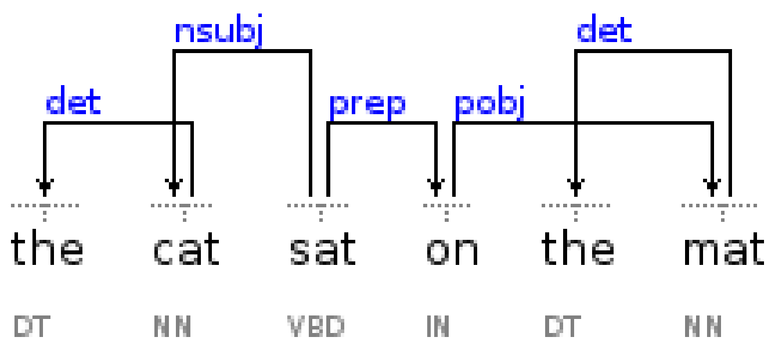
This is an older, and some argue more natural, decomposition of the sentence. It also often makes semantic interpretation (based on the meanings of the words) easier.

Lets look at a simple example:

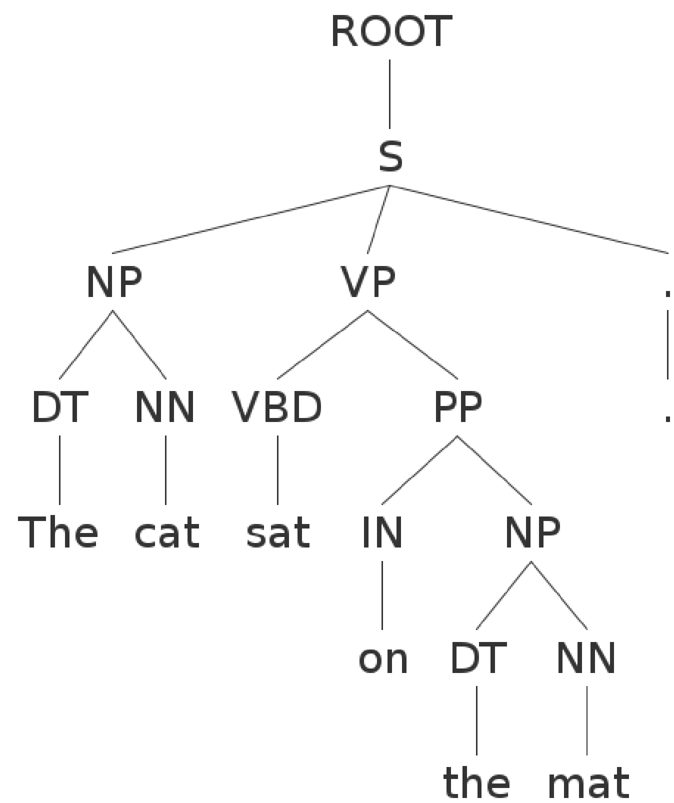
Dependencies

"The cat sat on the mat"

dependency tree



parse tree



constituency labels of leaf nodes

Dependencies

From the dependency tree, we can obtain a “sketch” of the sentence. i.e. by starting at the root we can look down one level to get:

“cat sat on”

And then by looking for the object of the prepositional child, we get:

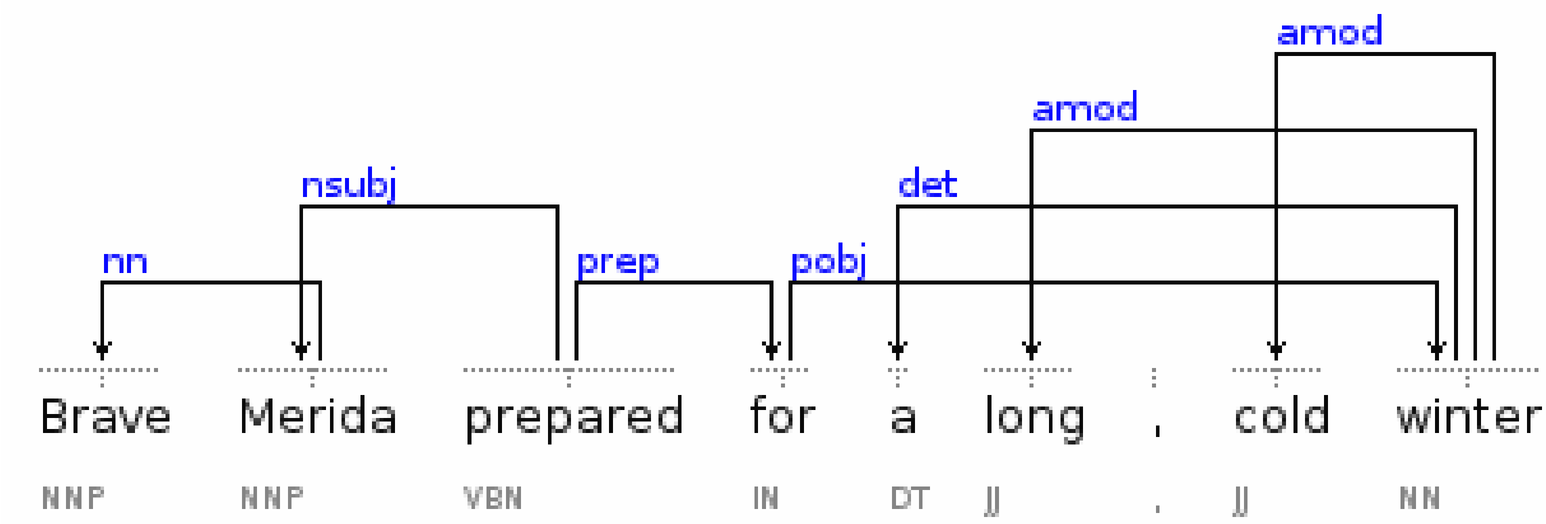
“cat sat on mat”

We can easily ignore determiners “a, the”.

And importantly, adjectival and adverbial modifiers generally connect to their targets:

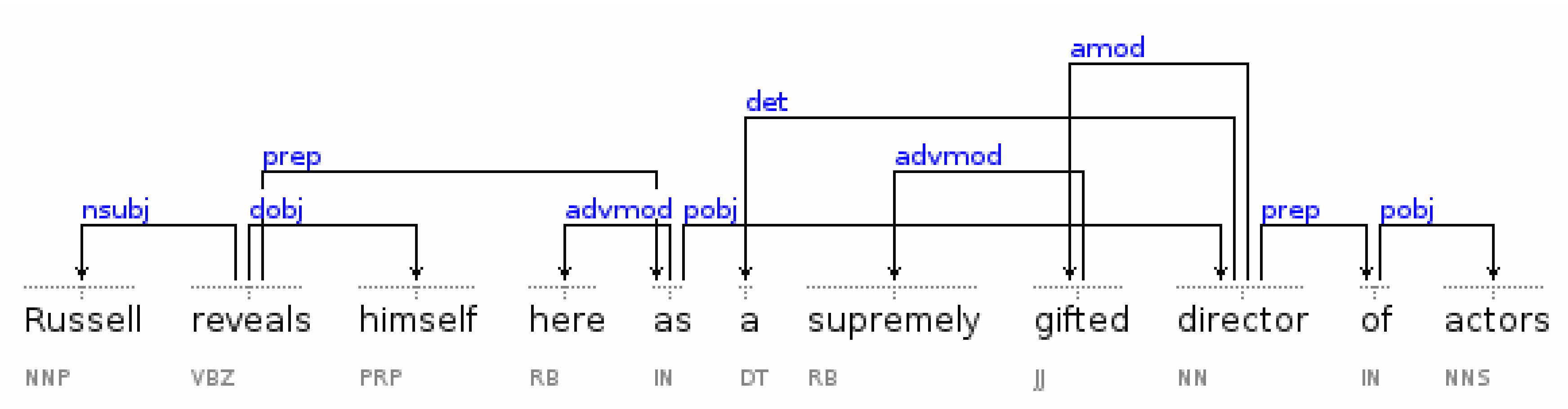
Dependencies

"Brave Merida prepared for a long, cold winter"



Dependencies

"Russell reveals himself here as a supremely gifted director of actors"



Dependencies

Stanford dependencies are constructed from the output of a constituency parser.

The mapping is based on hand-written regular expressions.

Dependency grammars have been widely used for sentiment analysis and for semantic embedding's of sentences.

Additional Details

- Rule-based IE
  - System T and AQL
- Statistical IE
  - HMM, Linear-CRF and Viterbi
- TFIDF and cosine similarity

Knowledge/Information Extraction (IE)

- “We are pleased that today's agreement guarantees our corporation will maintain a significant and long term presence in the Big Apple," McGraw-Hill president Harold McGraw III said in a statement.

--- *From New York Times April 24, 1997*

Knowledge/Information Extraction (IE)

- “We are pleased that today's agreement guarantees our corporation will maintain a significant and long term presence in the Big Apple," McGraw-Hill president Harold McGraw III said in a statement.

(prob=0.41)

--- From New York Times April 24, 1997

Labels:

Person Company Location Other

Knowledge/Information Extraction (IE)

- “We are pleased that today's agreement guarantees our corporation will maintain a significant and long term presence in the Big Apple," McGraw-Hill president Harold McGraw III said in a statement.

(prob=0.26)

--- *From New York Times April 24, 1997*

Labels:

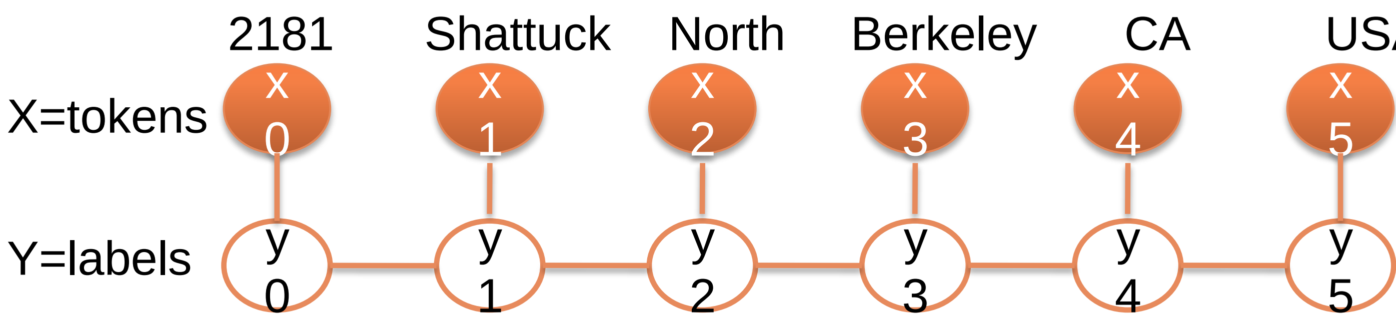
Person Company Location Other

A Graphical Model – Conditional Random Fields (CRF)

Text (address string):

E.g., *"2181 Shattuck North Berkeley CA USA"*

CRF Model:



Possible Extraction Worlds:

x	2181	Shattuck	North	Berkeley	CA	USA	
y1	apt. num	street name	city	city	state	country	(0.6)
y2	apt. num	street name	street name	city	state	country	(0.1)
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Viterbi Implemented in SQL

Viterbi Dynamic Programming Algorithm:

$$V(i,y) = \begin{cases} \max_{y'}\left(V(i-1,y') + \sum_{k=1}^{K} \lambda_k \bullet f_k \bullet f(y,y',x_i)\right), & \text{if } i \geq 0 \\ 0, & \text{if } i = -1. \end{cases}$$

	pos	street num	street name	city	state	country
2181	0	5	1	0	1	1
Shattuck	1	2	15	7	8	7
North	2	12	24	21	18	17
Berkeley	3	21	32	32	30	26
CA	4	29	40	38	42	35
USA	5	39	47	46	46	50

Summary

- N-grams
- Grammars
- Parsing
- Dependencies
- Details on
  - Rule-based IE
  - Statistical IE model
  - Document cosine similarity and TFIDF