# Statistical Learning Models for Text and Graph Data Sequence Labeling and Structured Output Learning: A First Look

Yangqiu Song

Hong Kong University of Science and Technology

*yqsong@cse.ust.hk*

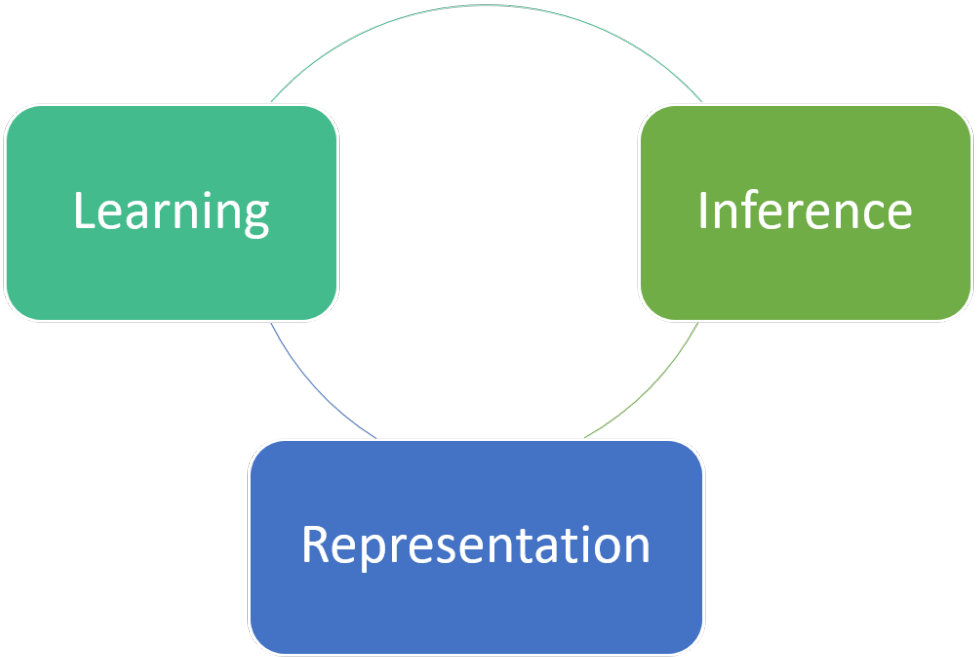
November 1, 2019

∗Contents are based on materials created by Dan Roth, Vivek Srikumar, Hongning Wang, Chris Manning

# Reference Content

- Dan Roth. CS546: Machine Learning and Natural Language . http://l2r.cs.uiuc.edu/~danr/Teaching/CS546-16/
- Vivek Srikumar. CS 6355 Structured Prediction. https://svivek.com/teaching/structured-prediction/spring2018/
- Hongning Wang. CS6501 Text Mining. http://www.cs.virginia.edu/~hw5x/Course/Text-Mining-2015-Spring/_site/
- Chris Manning. CS 224N/Ling 237. Natural Language Processing. https://web.stanford.edu/class/cs224n/

# Course Topics




- Representation: language models, word embeddings, topic models, knowledge graphs
- Learning: supervised learning,unsupervised learning, semi-supervised learning, distant supervision, indirect supervision, sequence models, deep learning, optimization techniques
- Inference: constraint modeling, joint inference, search algorithms

# Overview


1. Application of Sequence Labeling
2. Motivation of Structured Output Learning
3. Multiclass as a Structure: A Very Brief Digression
4. Discussion about Structured Prediction

# Overview


1. Application of Sequence Labeling
2. Motivation of Structured Output Learning
3. Multiclass as a Structure: A Very Brief Digression
4. Discussion about Structured Prediction

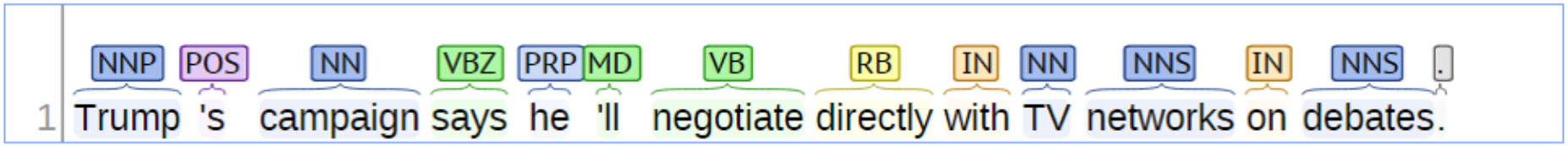
# Popular Statistical Machine Learning Algorithms for NLP

- Mid-1970s: Hidden Markov Models (HMMs) for speech recognition $\rightarrow$ probabilistic models
- Early 2000s: Conditional Random Fields (CRFs) for part-of-speech tagging $\rightarrow$ structured prediction
- Early 2000s: Latent Dirichlet Allocation (LDA) for modeling text documents $\rightarrow$ topic modeling
- Mid 2010s: sequence-to-sequence models for machine translation $\rightarrow$ Deep Learning neural networks with memory/state
- Now: ??? for natural language understanding/generation
  - Reinforcement learning?

# Sequence Tagging: Part of Speech

## Example (Part of speech)

**Part-of-Speech:**




Tags:

- NN: common noun
- NNP: proper noun
- VB: verb, base form
- VBZ: verb, 3rd person singular
- ...

# POS Tagging

- POS tagging is (was) a prerequisite for further NLP analysis
  - Syntax parsing
    - Basic unit for parsing
  - Information extraction
    - Indication of names, relations
  - Machine translation
    - The meaning of a particular word depends on its POS tag
  - Sentiment analysis
    - Adjectives are the major opinion holders
    - Good v.s. Bad, Excellent v.s. Terrible

# Public Tag Sets in NLP

- Brown corpus (Francis and Kucera 1961)
  - 500 samples, distributed across 15 genres in rough proportion to the amount published in 1961 in each of those genres
  - 87 tags
- Penn Treebank (Marcus et al. 1993)
  - Hand-annotated corpus of Wall Street Journal, 1M words
  - 45 tags, a simplified version of Brown tag set
  - Standard for English now: most statistical POS taggers are trained on this Tagset

# Sequence Tagging: Part of Speech

Penn Treebank part-of-speech tags (including punctuation).

| Tag | Description | Example | Tag | Description | Example |
|---|---|---|---|---|---|
| CC | Coordin. Conjunction | *and, but, or* | SYM | Symbol | *+, %, &* |
| CD | Cardinal number | *one, two, three* | TO | "to" | *to* |
| DT | Determiner | *a, the* | UH | Interjection | *ah, oops* |
| EX | Existential 'there' | *there* | VB | Verb, base form | *eat* |
| FW | Foreign word | *mea culpa* | VBD | Verb, past tense | *ate* |
| IN | Preposition/sub-conj | *of, in, by* | VBG | Verb, gerund | *eating* |
| JJ | Adjective | *yellow* | VBN | Verb, past participle | *eaten* |
| JJR | Adj., comparative | *bigger* | VBP | Verb, non-3sg pres | *eat* |
| JJS | Adj., superlative | *wildest* | VBZ | Verb, 3sg pres | *eats* |
| LS | List item marker | *1, 2, One* | WDT | Wh-determiner | *which, that* |
| MD | Modal | *can, should* | WP | Wh-pronoun | *what, who* |
| NN | Noun, sing. or mass | *llama* | WP$ | Possessive wh- | *whose* |
| NNS | Noun, plural | *llamas* | WRB | Wh-adverb | *how, where* |
| NNP | Proper noun, singular | *IBM* | $ | Dollar sign | *$* |
| NNPS | Proper noun, plural | *Carolinas* | # | Pound sign | *#* |
| PDT | Predeterminer | *all, both* | " | Left quote | (' or ") |
| POS | Possessive ending | *'s* | " | Right quote | (' or ") |
| PRP | Personal pronoun | *I, you, he* | ( | Left parenthesis | ( [, (, {, <) |
| PRP$ | Possessive pronoun | *your, one's* | ) | Right parenthesis | ( ], ), }, >) |
| RB | Adverb | *quickly, never* | , | Comma | , |
| RBR | Adverb, comparative | *faster* | . | Sentence-final punc | (. ! ?) |
| RBS | Adverb, superlative | *fastest* | : | Mid-sentence punc | (: ; ... – -) |
| RP | Particle | *up, off* | | | |

# How much ambiguity is there?

- Statistics of word-tag pair in Brown Corpus and Penn Treebank

| | | 87-tag Original Brown | | 45-tag Treebank Brown | |
|---|---|---|---|---|---|
| **Unambiguous (1 tag)** | | **44,019** | **11%** | **38,857** | **18%** |
| **Ambiguous (2–7 tags)** | | **5,490** | | **8844** | |
| Details: | 2 tags | 4,967 | | 6,731 | |
| | 3 tags | 411 | | 1621 | |
| | 4 tags | 91 | | 357 | |
| | 5 tags | 17 | | 90 | |
| | 6 tags | 2 | (*well*, *beat*) | 32 | |
| | 7 tags | 2 | (*still*, *down*) | 6 | (*well*, *set*, *round*, *open*, *fit*, *down*) |
| | 8 tags | | | 4 | (*'s*, *half*, *back*, *a*) |
| | 9 tags | | | 3 | (*that*, *more*, *in*) |

# Sequence Tagging: Named Entities

## Example (Named Entity Recognition)

**Named Entity Recognition:**

1 Person
Trump's campaign says he'll negotiate directly with TV networks on debates.

2 Person Dur Location
The move by Trump, coming just hours after his and other campaigns huddled in a Washington suburb to craft a three-page letter of possible demands, thwarts an effort to find consensus after what most candidates agreed was a debacle hosted by Org Date
CNBC last week.

- Pers: Person
- Location
- Org: Organization
- Date/time

# Sequence Labeling as Learning

| VBG | NN | IN | DT | NN | IN | NN |
|---|---|---|---|---|---|---|
| Chasing | opportunity | in | an | age | of | upheaval |

**POS tagging**

| B | B | I | I | B | I | B | I | B | B |
|---|---|---|---|---|---|---|---|---|---|
| 而 | 相 | 对 | 于 | 这 | 些 | 品 | 牌 | 的 | 价 |

**Word segmentation**

| PERS | O | O | O | ORG | ORG |
|---|---|---|---|---|---|
| Murdoch | discusses | future | of | News | Corp. |

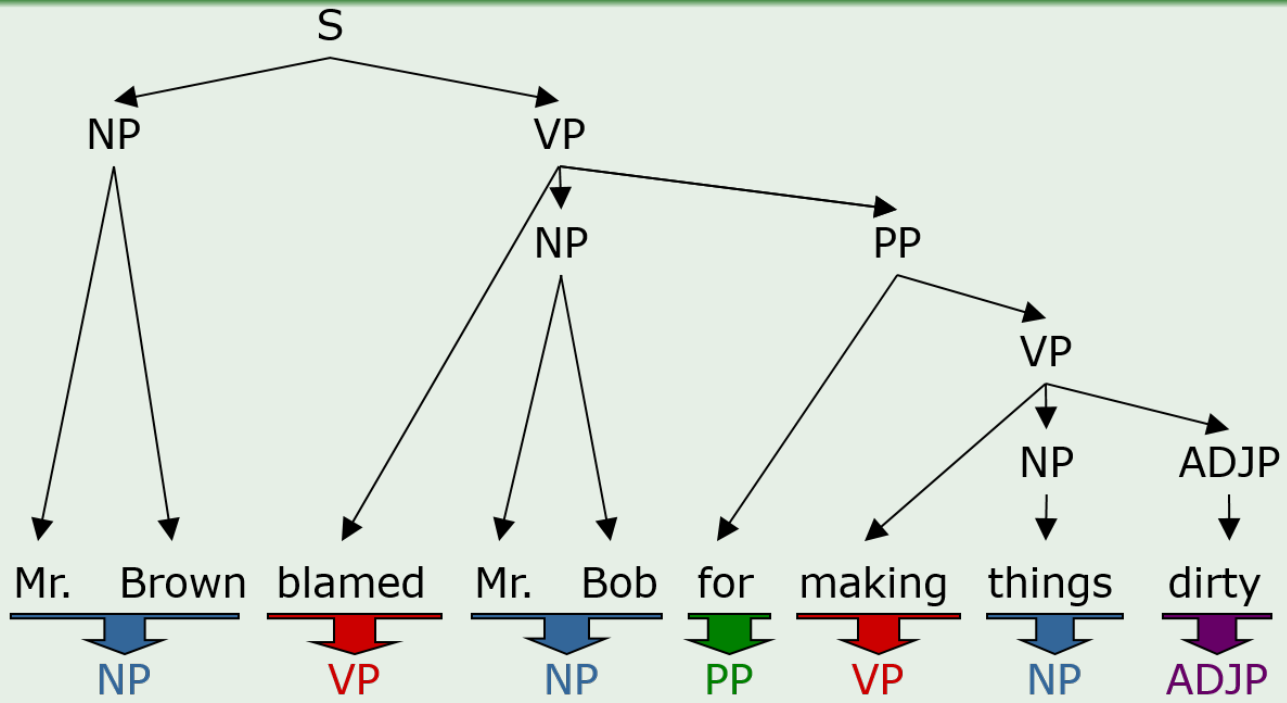
**Named entity recognition**


**Text segmentation**

- The BIO encoding
  - B-NP: beginning of a noun phrase chunk
  - I-NP: inside of a noun phrase chunk
  - O: outside of a noun phrase chunk

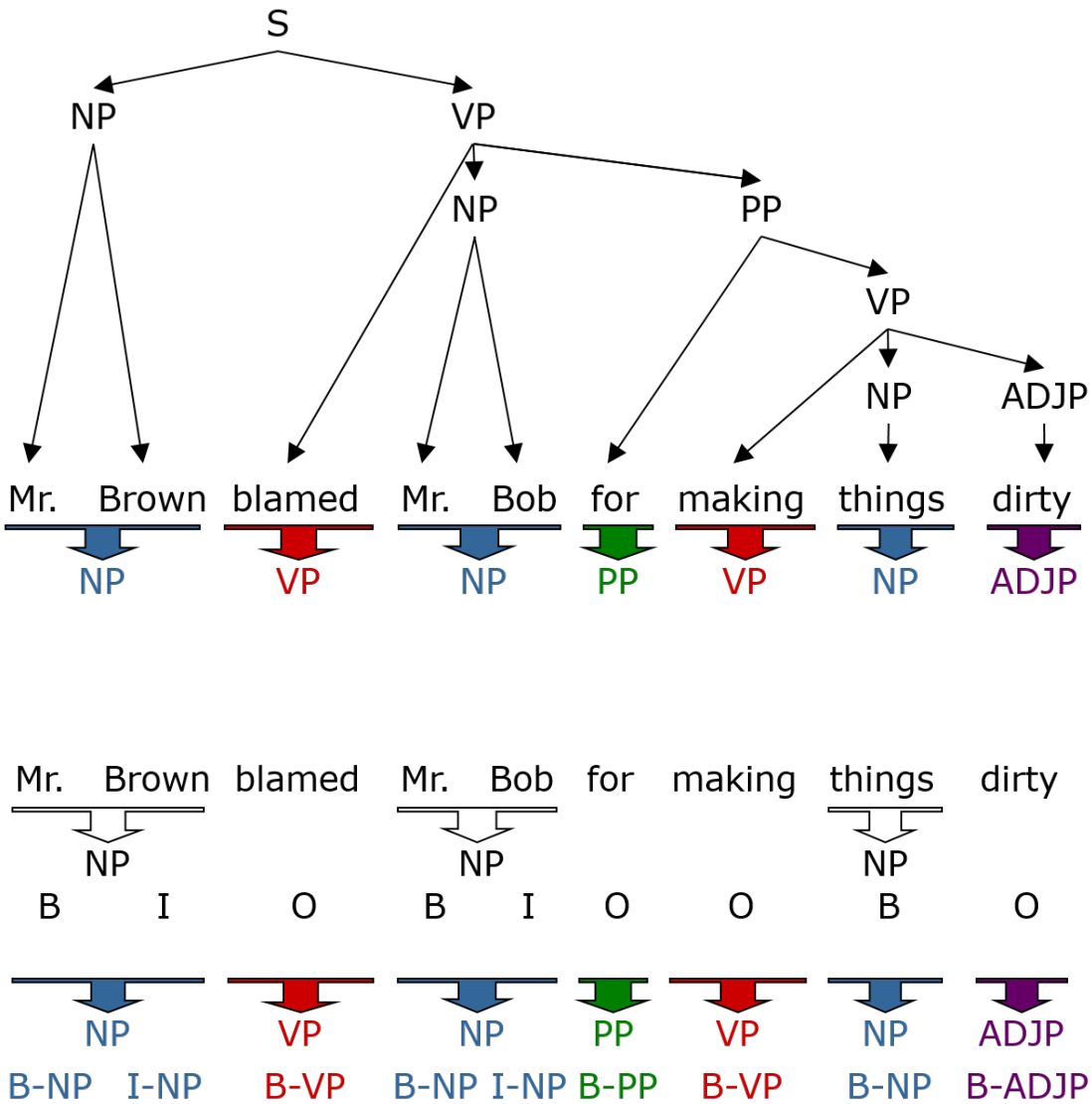
# Sequence Tagging: Shallow Parsing (Chunking)

## Example (Chunking)

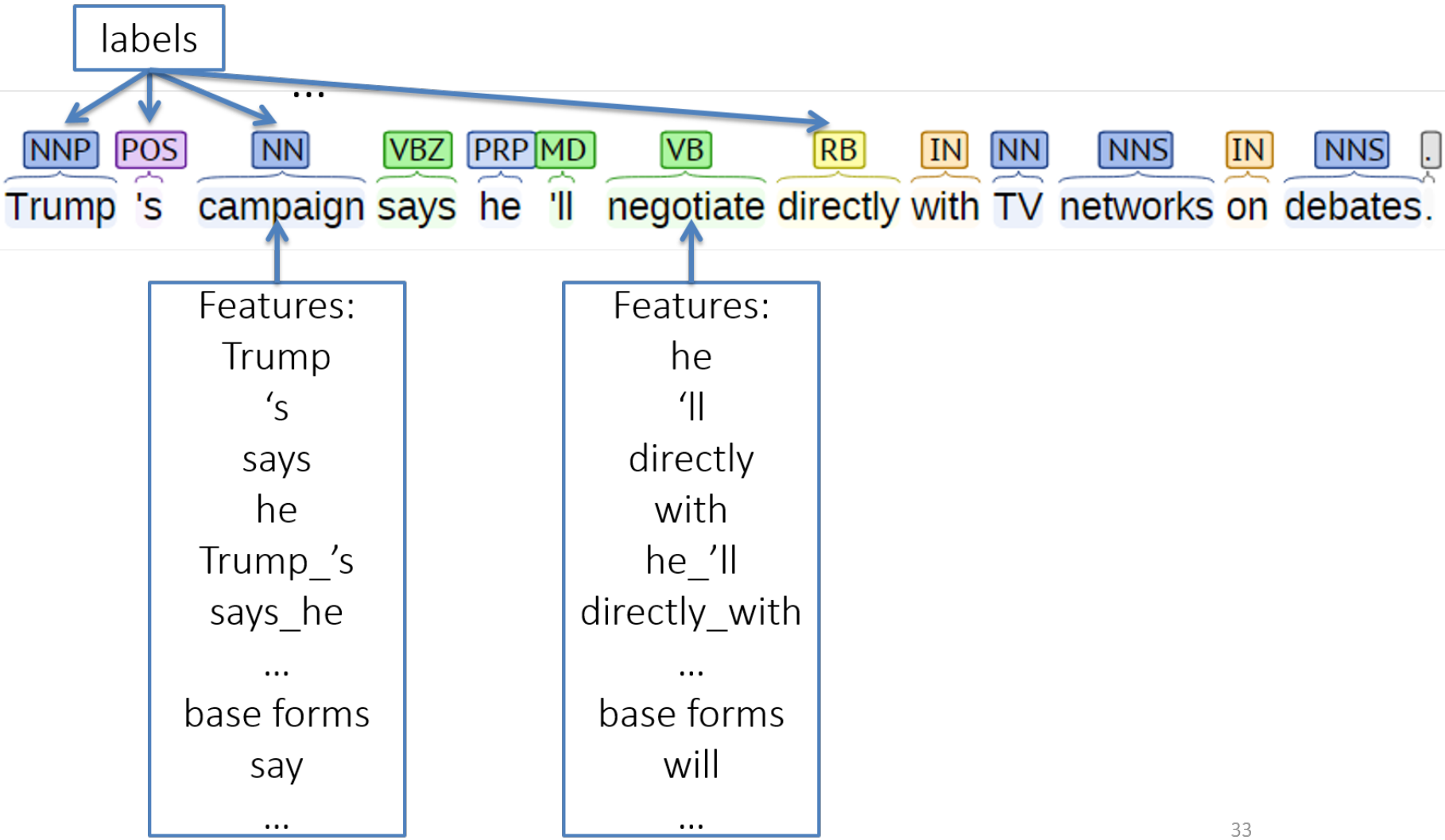



- Not all the parse tree information is needed
- By shallow parsing we mean: identifying non-overlapping, non-embedding phrases
- Shallow Parsing = Text Chunking
- Usually text chunking without any further specification follows the definition from CoNLL-2000 shared task

# Sequence Labeling as Learning



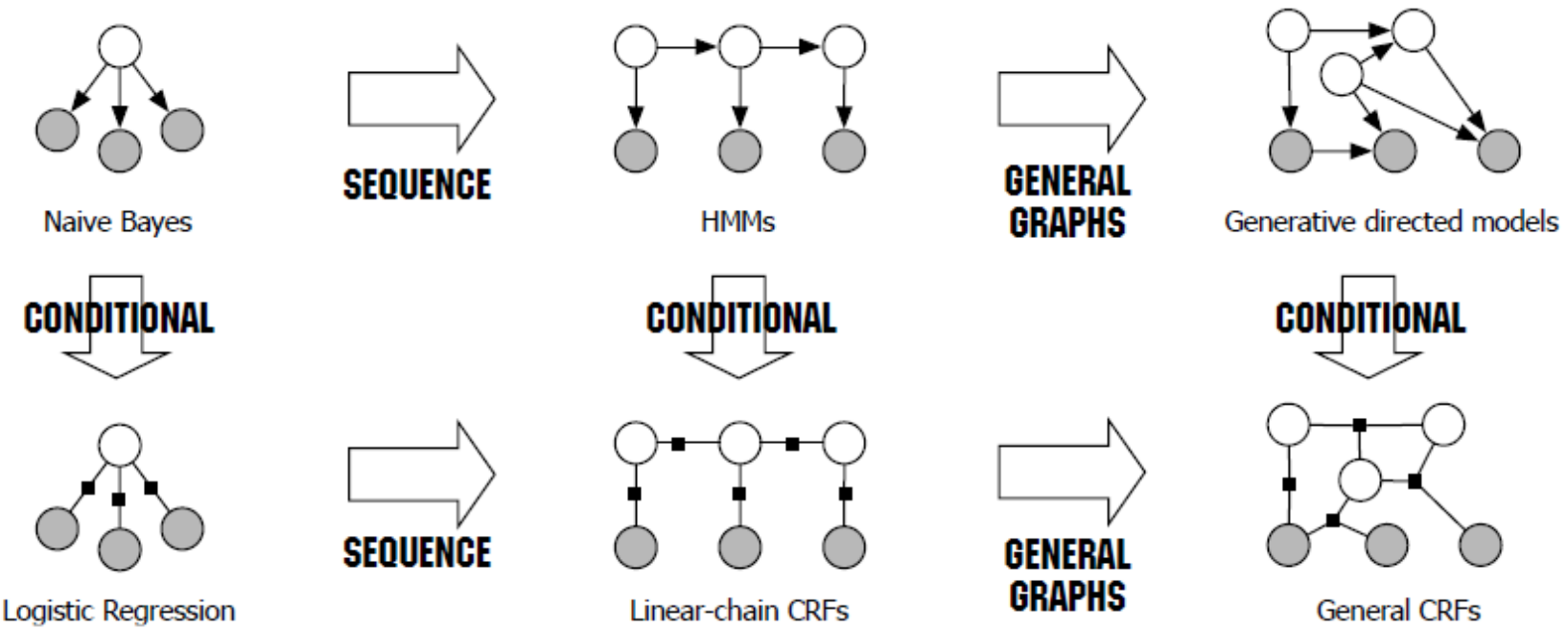
# Classification Problem




33

# Classifiers Feasible for Sequence Labeling

- Generative
  - Naive Bayes
  - Hidden Markov model (HMM)
- Discriminative models
  - Maximum entropy, logistic regression
  - Maximum Entropy Markov Model (MMEM)
  - Conditional random field (CRF)

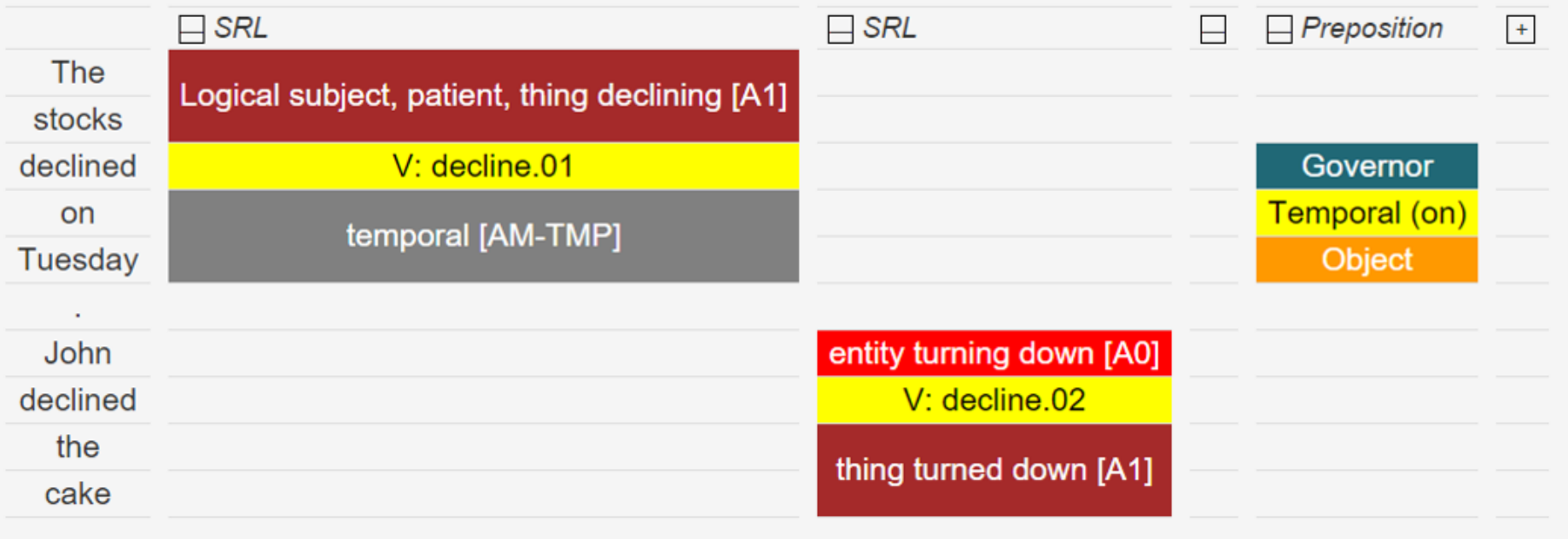


# Overview


1. Application of Sequence Labeling
2. Motivation of Structured Output Learning
3. Multiclass as a Structure: A Very Brief Digression
4. Discussion about Structured Prediction

# Frame based Semantics

## Example (Semantic Role Labeling)



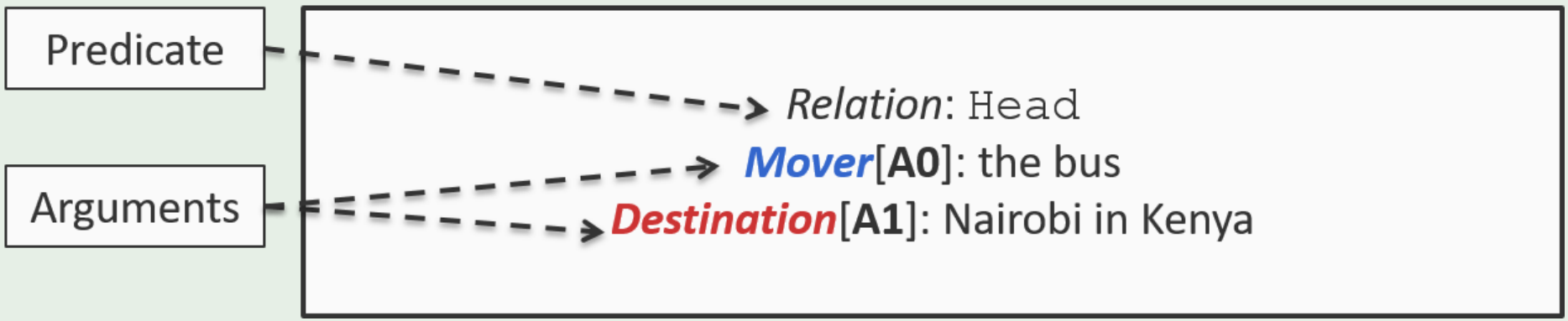

- Predicates
- Arguments
- Senses

# Semantic Role Labeling

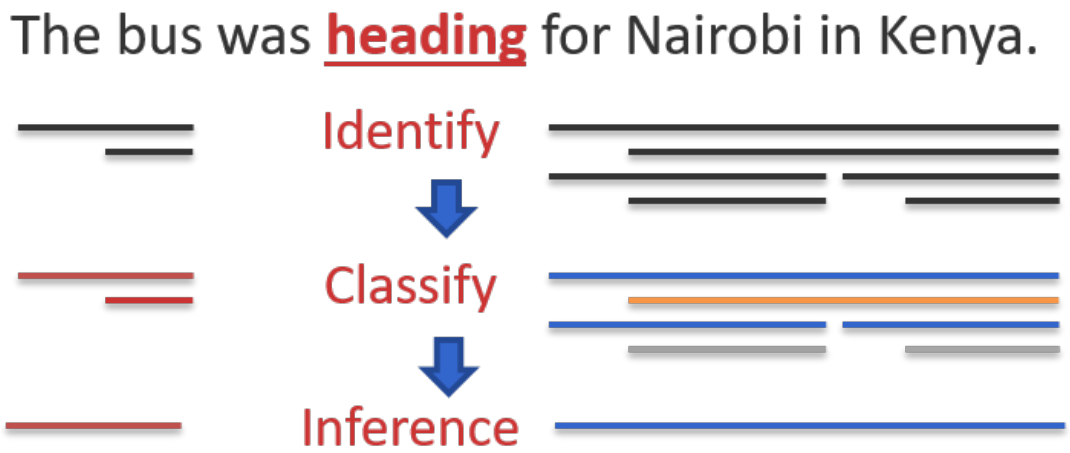
- Based on the dataset PropBank (Palmer et al. (2005))
  - Large human-annotated corpus of verb semantic relations
- The task: To predict arguments of verbs

## Example ("The bus was **heading** for Nairobi in Kenya")

Given the sentence, identifies who does what to whom, where and when.

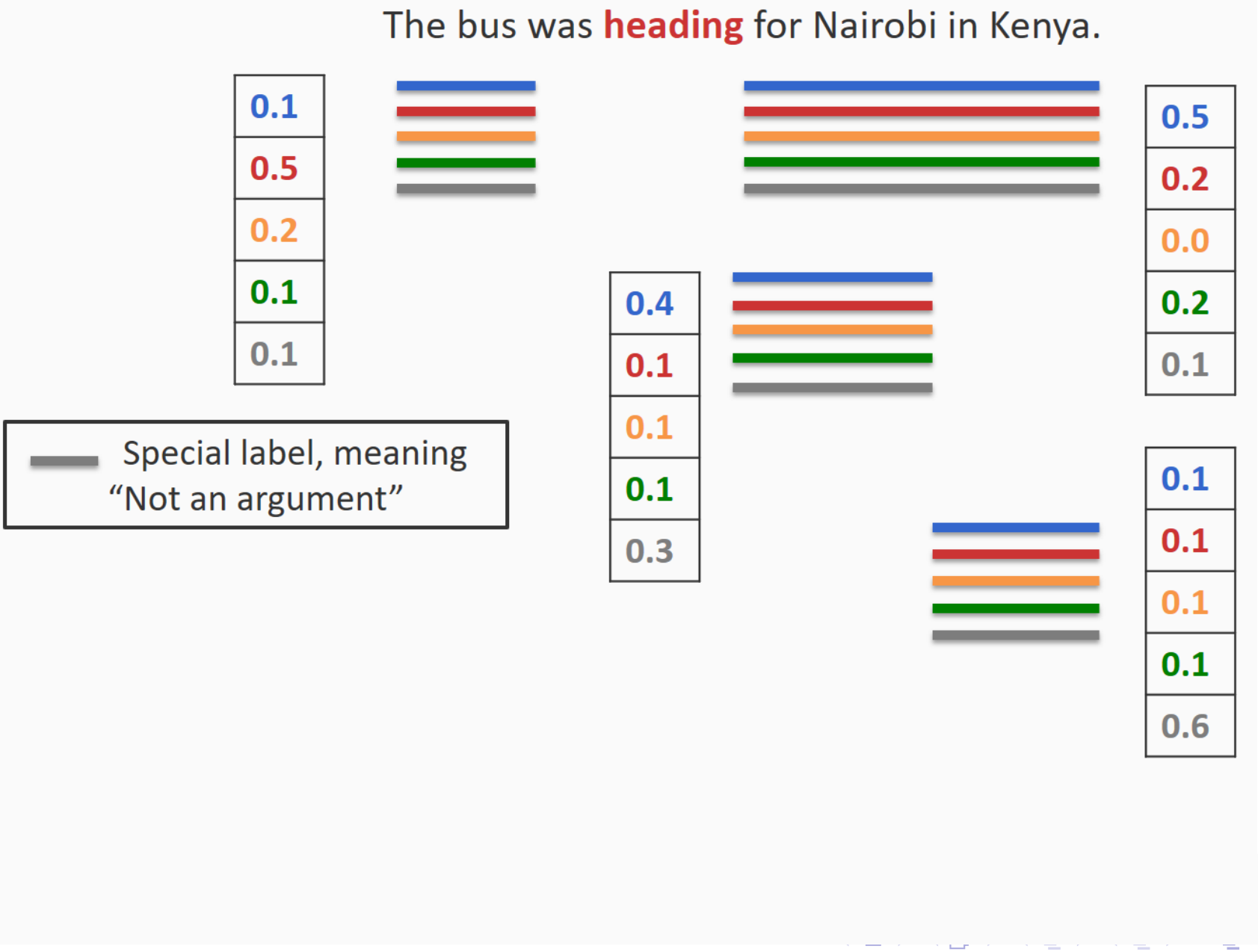


# Predicting Verb Arguments

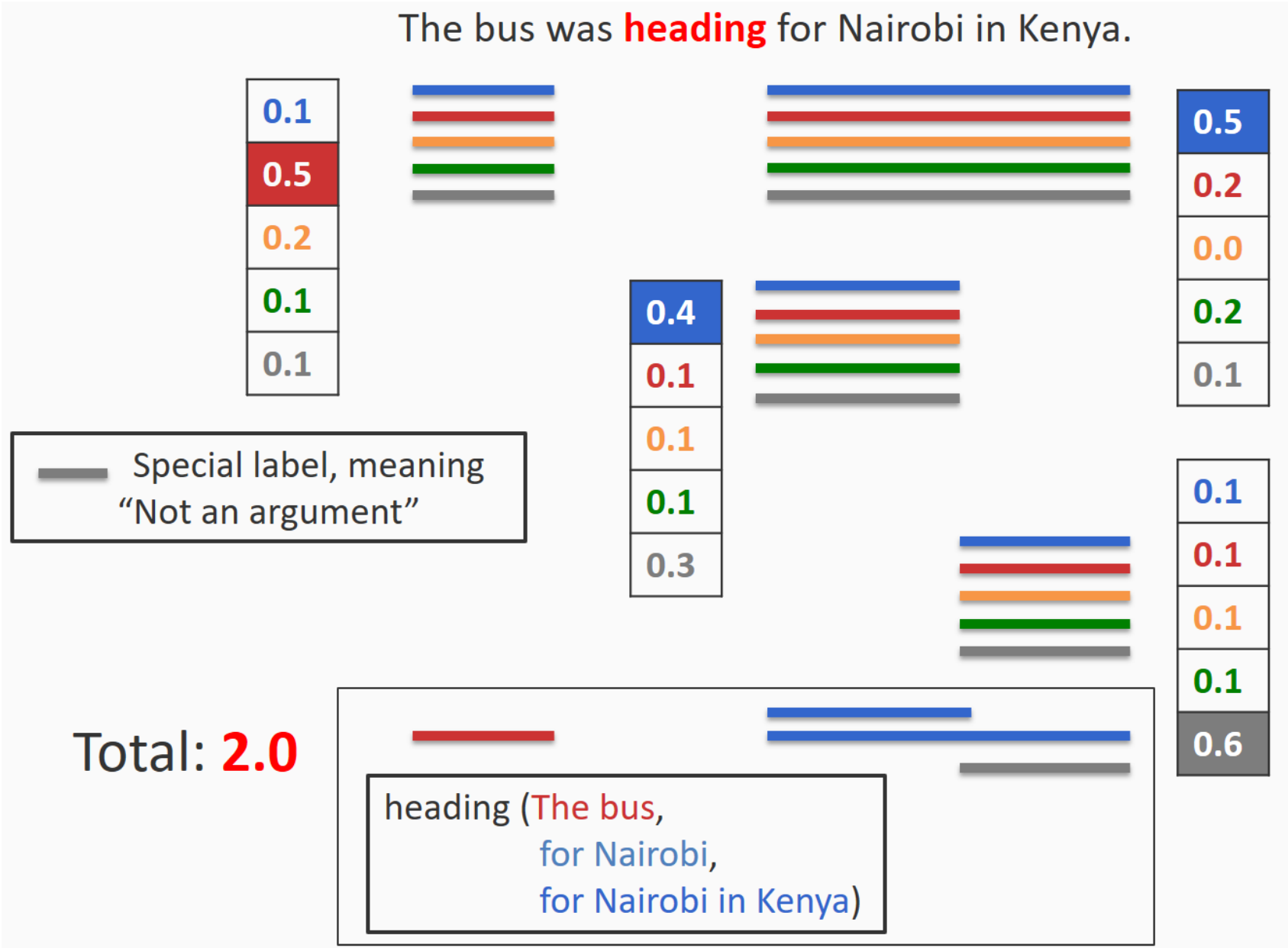



- Identify candidate arguments for verb using parse tree
  - Filtered using a binary classifier
- Classify argument candidates
  - Multi-class classifier (one of multiple labels per candidate)
- Inference
  - Using probability estimates from argument classifier
  - Must respect structural and linguistic constraints, e.g., no overlapping arguments

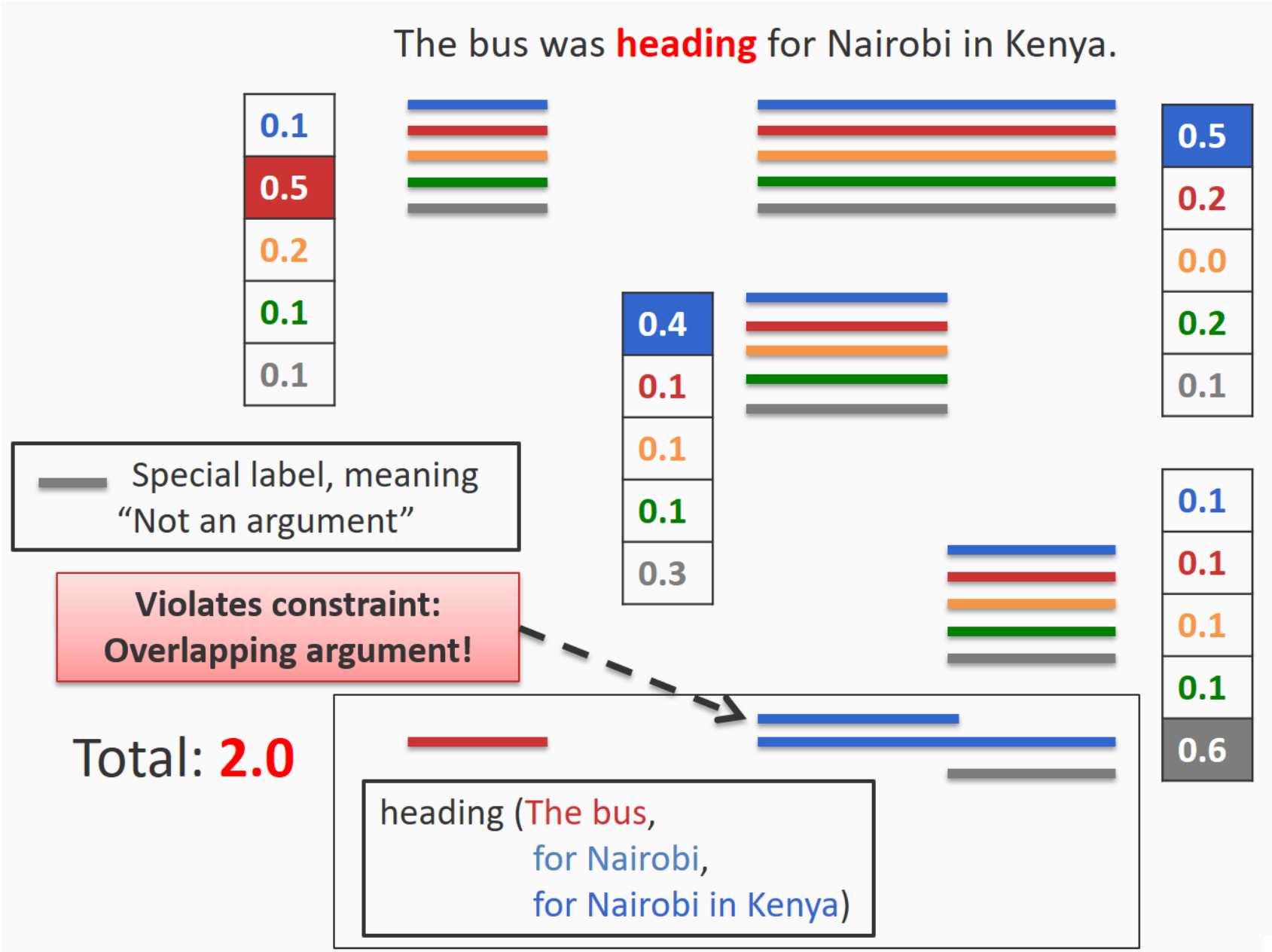
# Inference: Verb Arguments



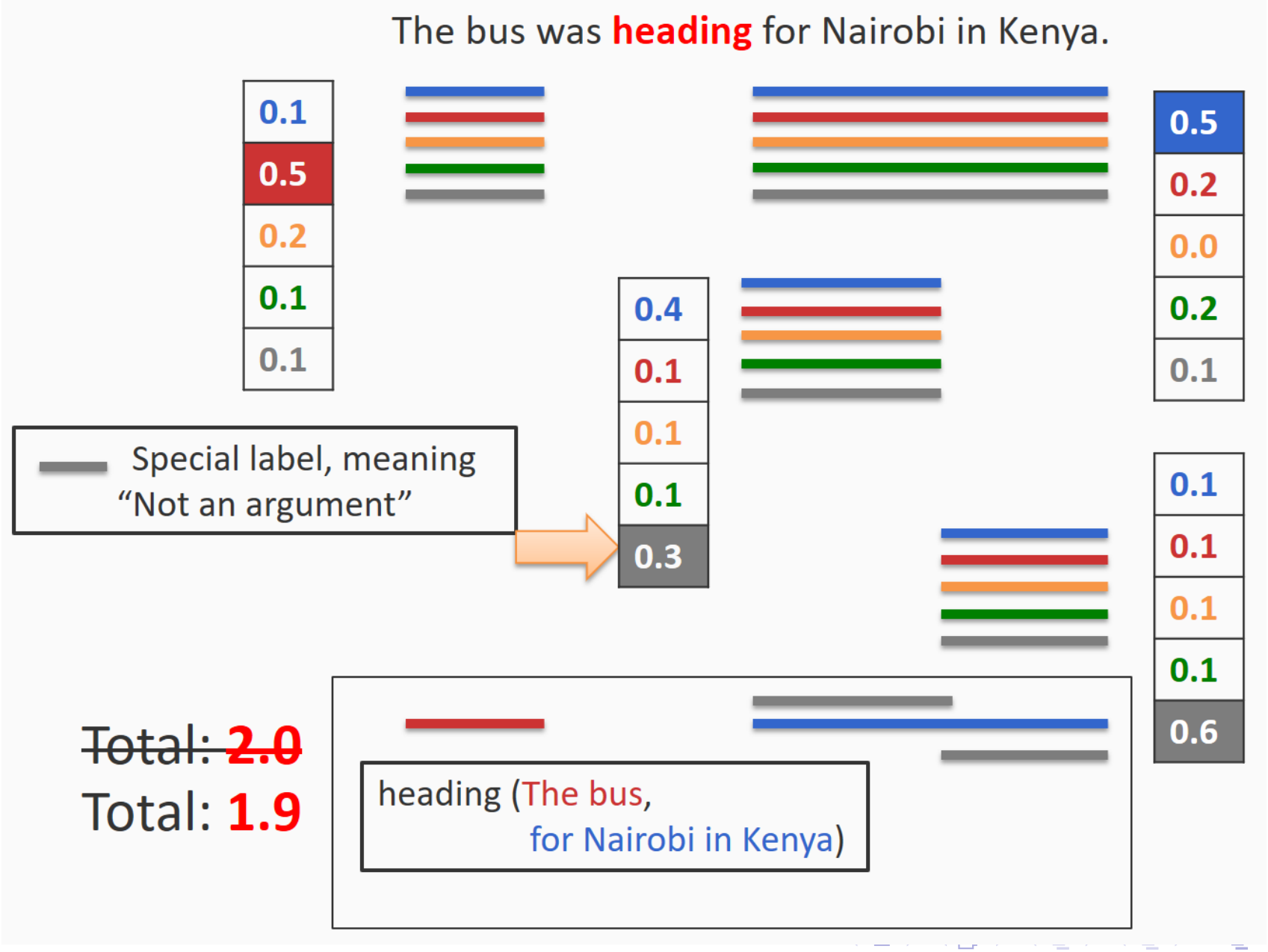
# Inference: Verb Arguments



# Inference: Verb Arguments



# Inference: Verb Arguments

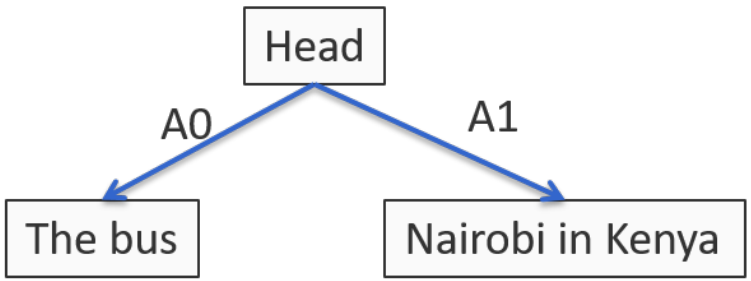


# Structured output is...

- A data structure with a pre-defined schema
  - Eg: SRL converts raw text into a record in a database

| Predicate | A0 | A1 | Location |
|---|---|---|---|
| Head | The bus | Nairobi in Kenya | - |

- Equivalently, a graph
  - Often restricted to be a specific family of graphs: chains, trees, etc



# Object Detection

- How would you design a predictor that labels all the parts using the tools we have seen so far?

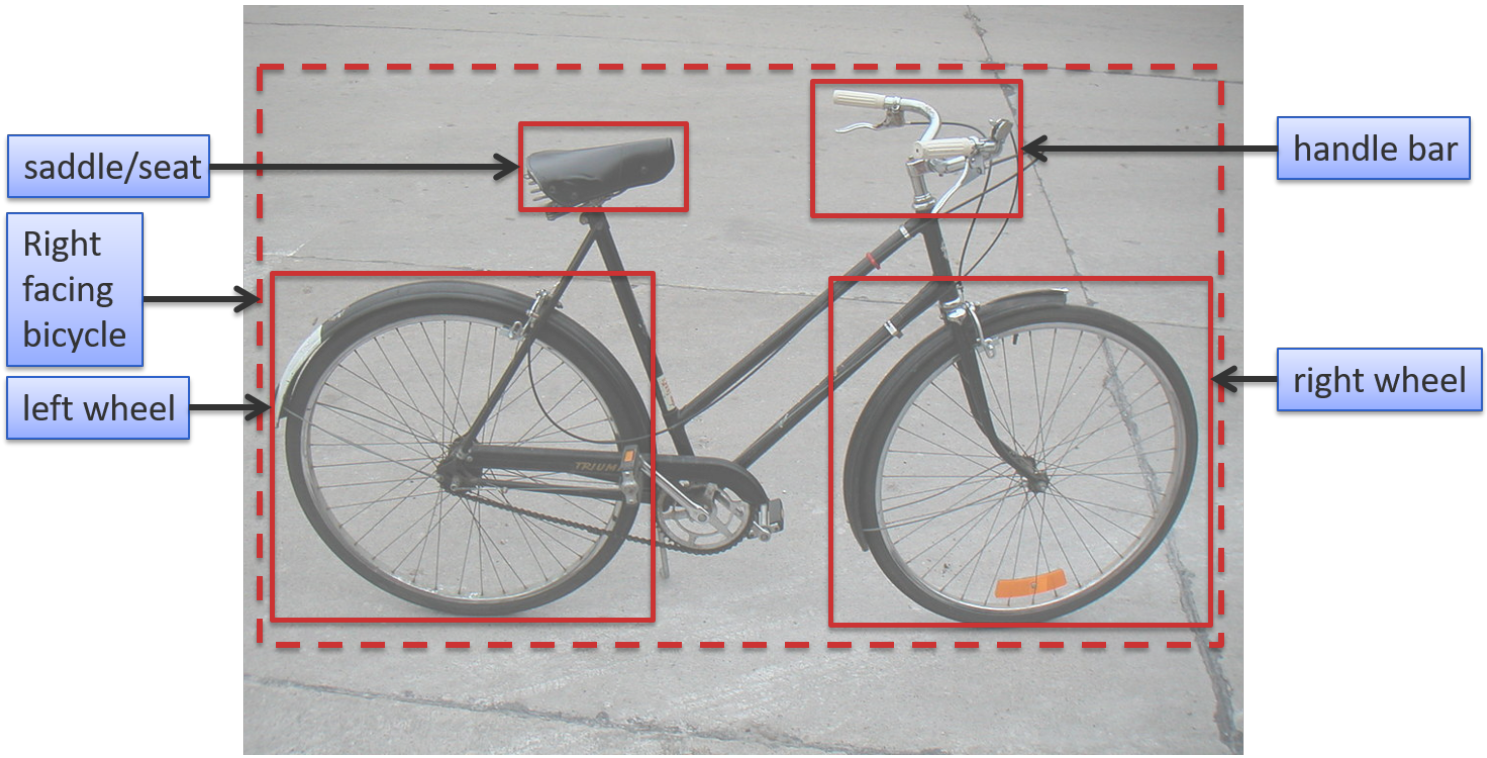



Photo by Andrew Dressel - Own work. Licensed under Creative Commons Attribution-Share Alike 3.0

# Object Detection

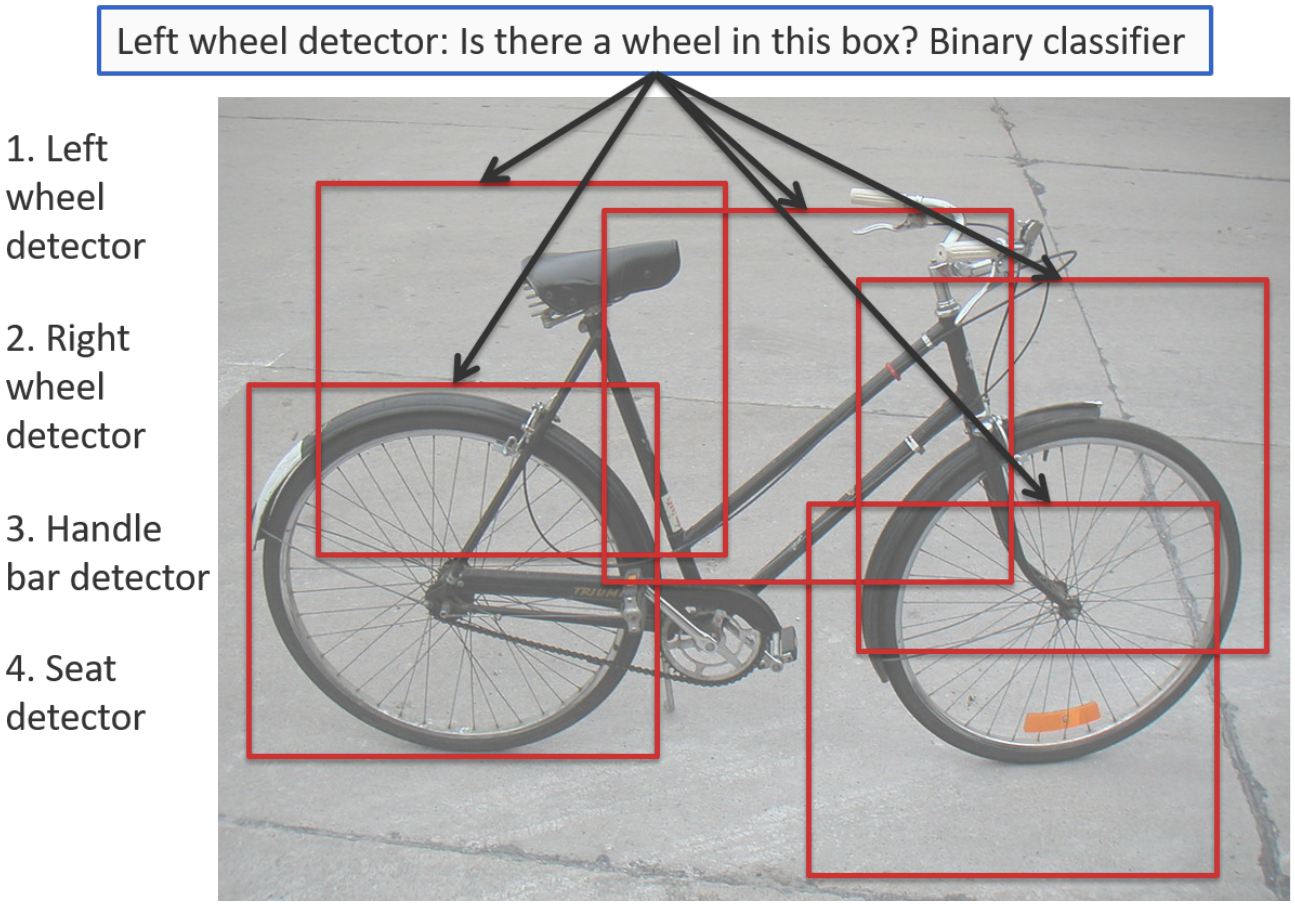
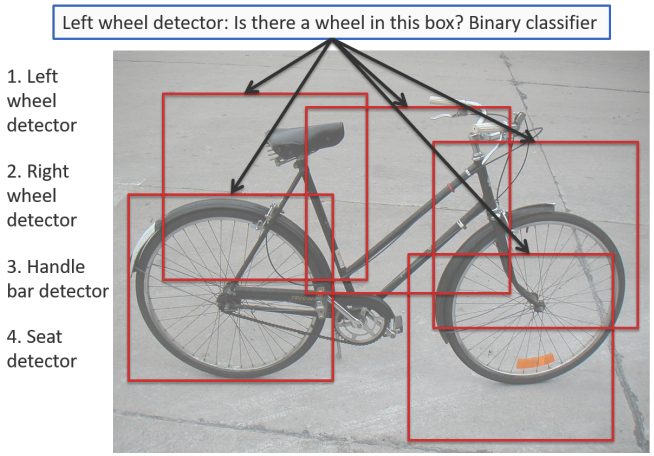


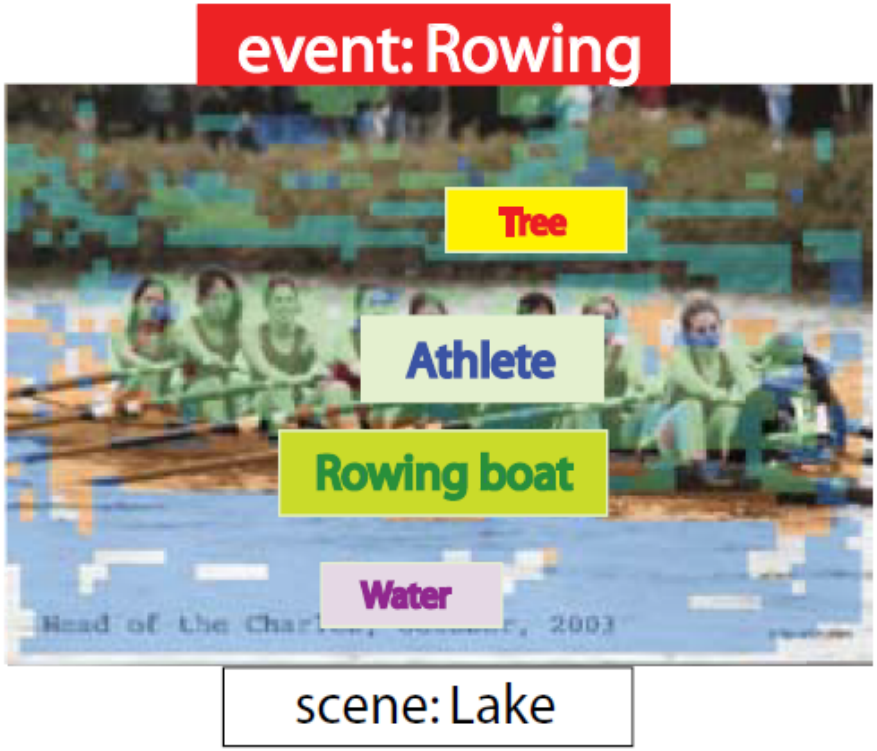

Photo by Andrew Dressel - Own work. Licensed under Creative Commons Attribution-Share Alike 3.0

# Object Detection




- Final output: Combine the predictions of these individual classifiers (local classifiers)
- The predictions interact with each other
- Eg: The same box can not be both a left wheel and a right wheel, handle bar does not overlap with seat, etc.
- Need inference to compose the output

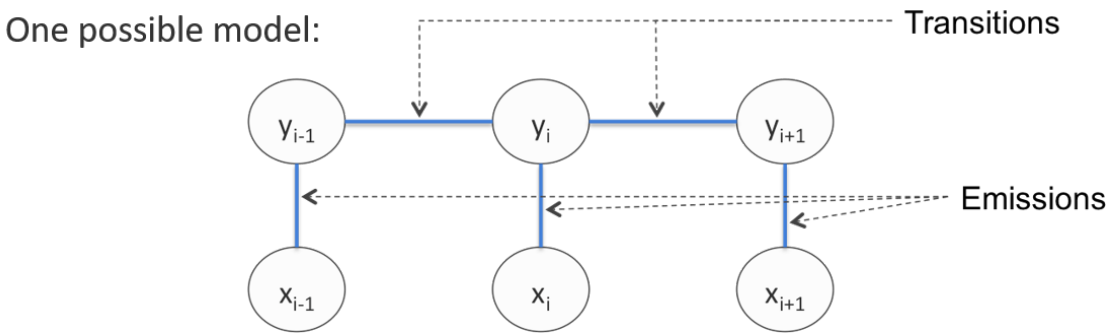
# How about this?


Li Fei-Fei and Li-Jia Li. What, Where and Who? Telling the Story of an Image by Activity Classification, Scene Recognition and Object Categorization

# Sequence Labeling as a Special Structured Output Learning Problem

- Input: A sequence of tokens (like words)
- Output: A sequence of labels of same length as input
- Given a word, its label depends on :
  - The identity and characteristics of the word
    - E.g., Raises is a Verb because it ends in es (among other reasons)
  - Its grammatical context
    - Fed in "The Fed" is a Noun because it follows a Determiner
    - Fed in "I fed the.." is a Verb because it follows a Pronoun



# Structured output is...

- A graph, possibly labeled and/or directed (representation)
  - Possibly from a restricted family, such as chains, trees, etc.
  - A discrete representation of input
    - E.g., A table, the SRL frame output, a sequence of labels etc.
- A collection of inter-dependent decisions
  - E.g., The sequence of decisions used to construct the output
- The result of a combinatorial optimization problem
  $\arg\max_{\mathbf{y}} \text{score}(\mathbf{x}, \mathbf{y})$
  - We have seen something similar before in the context of multiclass
  - Question: Why can't we treat each output as a label and train/predict as multiclass?

# Challenges with Structured Output

- Two challenges
  - We cannot train a separate weight vector for each possible inference outcome
    - For multiclass, we could train one weight vector for each label
  - We cannot enumerate all possible structures for inference
    - Inference for multiclass was easy
- Solution
  - Decompose the output into parts that are labeled
  - Define
    - how the parts interact with each other
    - how labels are scored for each part
    - an inference algorithm to assign labels to all the parts

# Overview


1. Application of Sequence Labeling
2. Motivation of Structured Output Learning
3. Multiclass as a Structure: A Very Brief Digression
4. Discussion about Structured Prediction

# Multiclass as a Structured Output

- A graph (in general, hypergraph), possibly labeled and/or directed
- A collection of inter-dependent decisions
- The output of a combinatorial optimization problem $\arg\max_{\mathbf{y}} \text{score}(\mathbf{x}, \mathbf{y})$

- A graph with one node and no edges: Node label is the output
- Can be composed via multiple decisions
- Winner-take-all $\textit{label} = \arg\max_i \mathbf{w}_i^\top \mathbf{x}$

# Multiclass is a Structure: Implications

- A lot of the ideas from multiclass may be generalized to structures
  - Not always trivial, but useful to keep in mind
- Broad statements about structured learning must apply to multiclass classification
  - Useful for sanity check, also for understanding
- Binary classification is the most trivial form of structured classification
  - Multiclass with two classes

# Overview


1. Application of Sequence Labeling
2. Motivation of Structured Output Learning
3. Multiclass as a Structure: A Very Brief Digression
4. Discussion about Structured Prediction

# Decomposing the Output

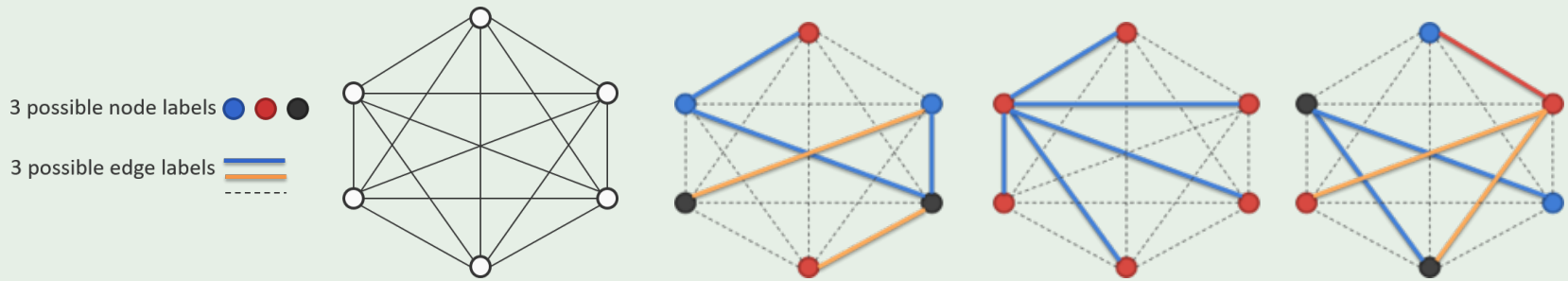
- We need to produce a graph
  - We cannot enumerate all possible graphs for the argmax
- Solution: Think of the graph as combination of many smaller parts
  - The parts should agree with each other in the final output
  - Each part has a score
  - The total score for the graph is the sum of scores of each part
- Decomposition of the output into parts also helps generalization
  - Why?

# Decomposing the Output: Example

- The scoring function (via the weight vector) scores outputs
- For generalization and ease of inference, break the output into parts and score each part
- The score for the structure is the sum of the part scores
- What is the best way to do this decomposition? Depends...

## Example (Decomposing the Output)

- Nodes and edges are labeled and the blue and orange edges form a tree
- Goal: Find the highest scoring tree




- The output $\mathbf{y}$ is a labeled assignment of the nodes and edges
- The input $x$ not shown here

# Decomposing the Output: Example

- Goal:

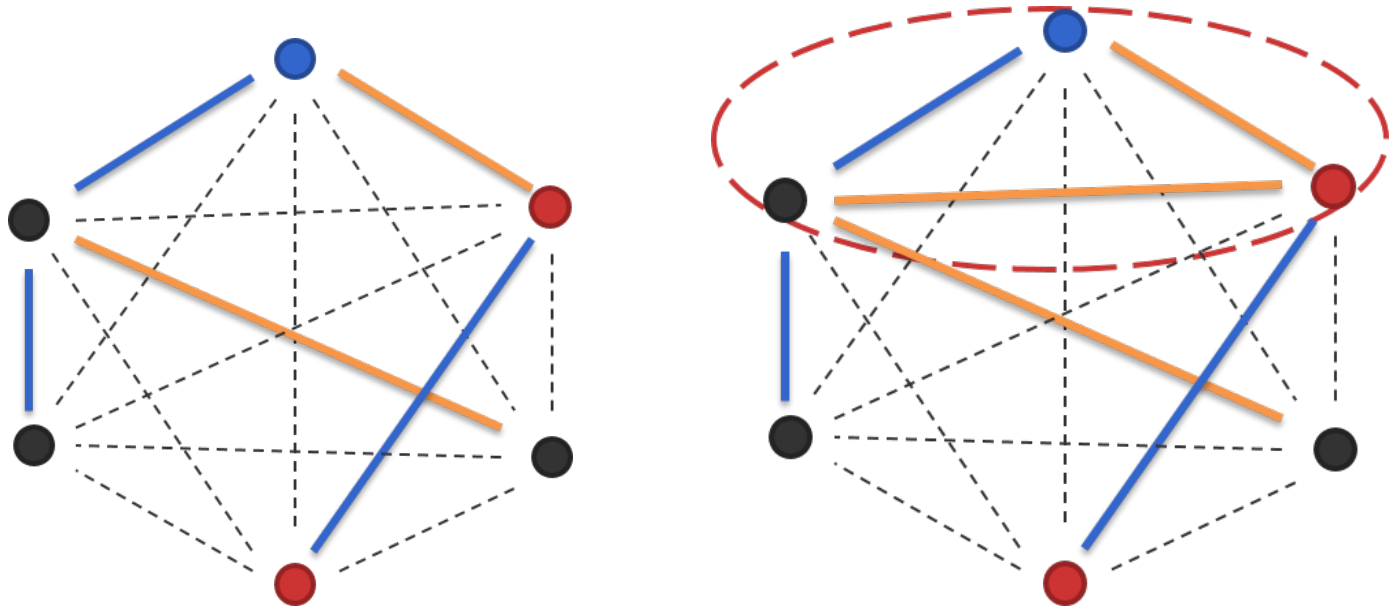
$$\text{score}(\mathbf{x}, \mathbf{y}) = \sum_{n \in \text{nodes}(\mathbf{x},\mathbf{y})} \text{score}(n) + \sum_{e \in \text{edges}(\mathbf{x},\mathbf{y})} \text{score}(e)$$

- Option 1: Decompose fully. All nodes and edges are independently scored
  - Still need to ensure that the colored edges form a valid output (i.e. a tree)
  - Prediction:

$$\begin{array}{ll} \arg\max_y & \sum_{n \in \text{nodes}(\mathbf{x},\mathbf{y})} \text{score}(\mathbf{x}, \mathbf{y}) \\ s.t. & \mathbf{y} \text{ forms a tree} \end{array}$$

# Decomposing the Output: Example

- Still need to ensure that the colored edges form a valid output (i.e. a tree)

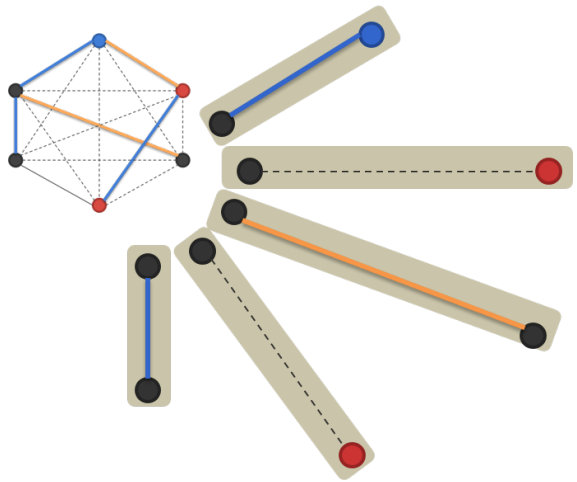


- Right: This is invalid output!
- Even this simple decomposition requires inference to ensure validity

# Decomposing the Output: Example

- Option 2: Score each edge and its nodes together

$$\text{score}(\mathbf{x}, \mathbf{y}) = \sum_{\substack{n_1, n_2 \in \text{nodes}(\mathbf{x},\mathbf{y}) \\ \forall e \in \text{edges}(\mathbf{x},\mathbf{y})}} \text{score}(n_1, n_2, e)$$



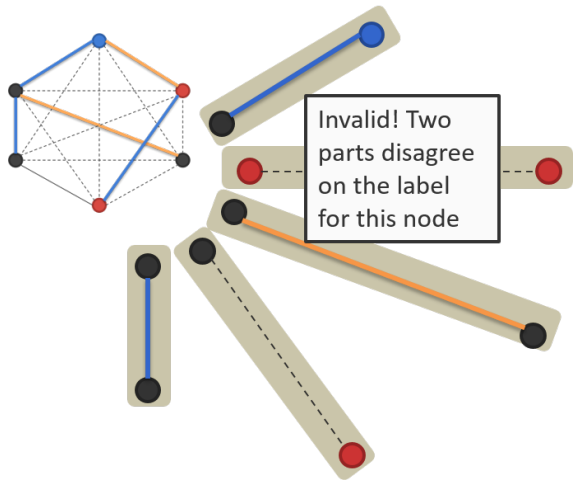
Inference should ensure that

- The output is a tree, and
- Shared nodes have the same label in all the parts

# Decomposing the Output: Example

- Option 2: Score each edge and its nodes together

$$\text{score}(\mathbf{x}, \mathbf{y}) = \sum_{\substack{n_1, n_2 \in \text{nodes}(\mathbf{x},\mathbf{y}) \\ \forall e \in \text{edges}(\mathbf{x},\mathbf{y})}} \text{score}(n_1, n_2, e)$$




Inference should ensure that

- The output is a tree, and
- Shared nodes have the same label in all the parts

# Decomposing the Output: Example

- Many other decompositions possible
- In general, we need a learning phase (learn the parameters of a model) and a inference phase (to decide the labels for an example)

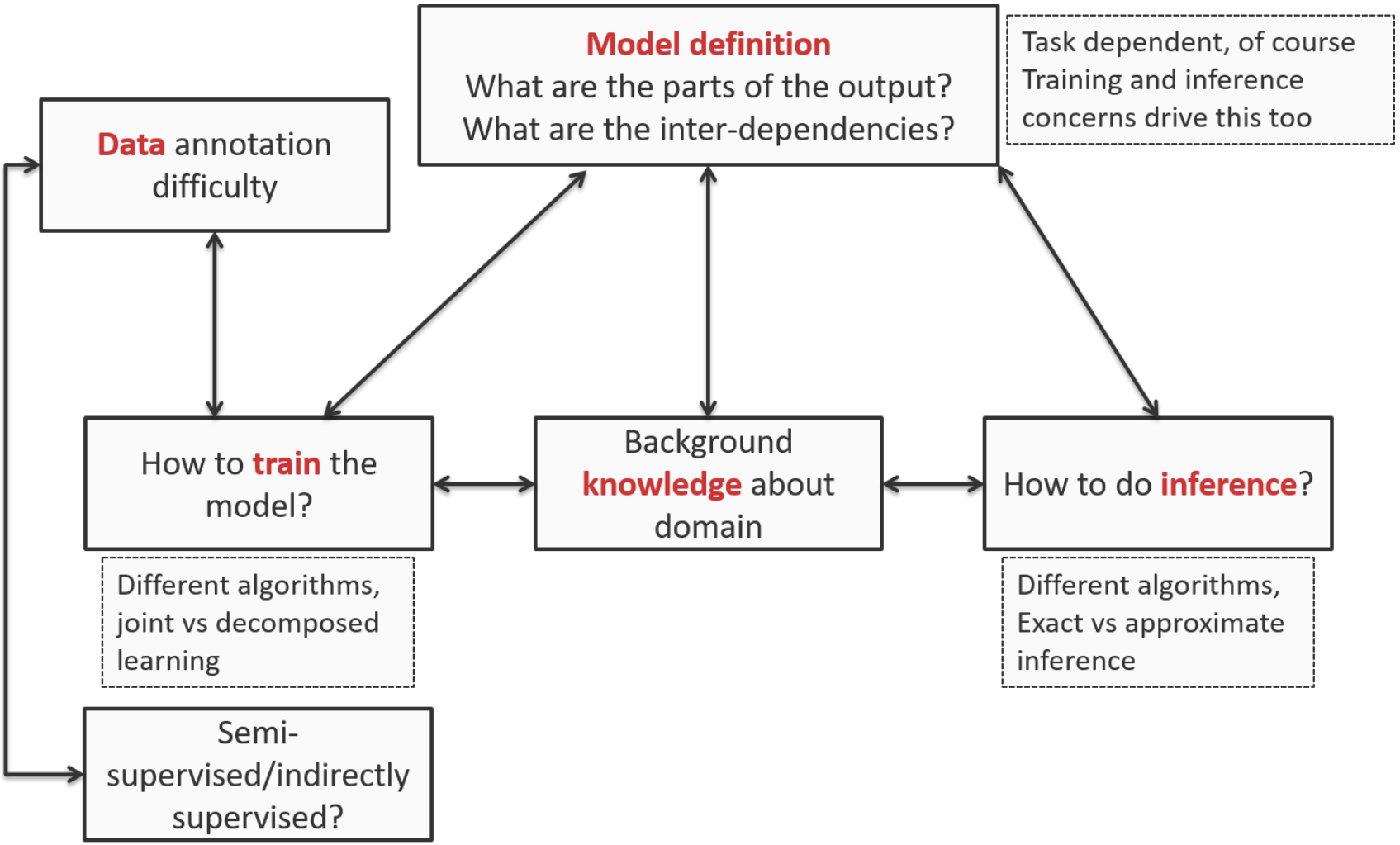
# Inference

- Each part is scored independently
  - Key observation: Number of possible inference outcomes for each part may not be large
  - Even if the number of possible structures might be large
- Inference: How to glue together the pieces to build a valid output?
  - Depends on the "shape" of the output
- Computational complexity of inference is important
  - Worst case: intractable
  - With assumptions about the output, polynomial algorithms exist
    - Predicting sequence chains: Viterbi algorithm
    - To parse a sentence into a tree: Cocke-Younger-Kasami algorithm
    - In general, might have to either live with intractability or approximate

# Training Regimes

- Decomposition of outputs gives two approaches for training
  - Decomposed training/learning without inference
    - Learning algorithm does not use the prediction procedure during training
  - Global training/Joint training/Inference-based training
    - Learning algorithm uses the final prediction procedure during training
- Similar to the two strategies we had before with multiclass
  - One-vs.-all
  - Kesler construction
- Inference complexity often an important consideration in choice of modeling and training
  - Especially so if full inference plays a part during training
  - Ease of training smaller/less complex models could give intermediate training strategies between fully decomposed and fully joint

# Computational Issues: Reprise



# Summary

- We saw several examples of structured output (structures are graphs)
  - Sometimes useful to think of them as a sequence of decisions
  - Also useful to think of them as data structures
- Multiclass is the simplest type of structure
  - Lessons from multiclass are useful
- Modeling outputs as structures
  - Decomposition of the output, inference, training

# Further Reading

- Kartik Goyal, Graham Neubig, Chris Dyer, Taylor Berg-Kirkpatrick: A Continuous Relaxation of Beam Search for End-to-End Training of Neural Sequence Models. AAAI 2018: 3045-3052 (Goyal et al. (2018))
- Kartik Goyal, Chris Dyer, Taylor Berg-Kirkpatrick: An Empirical Investigation of Global and Local Normalization for Recurrent Neural Sequence Models Using a Continuous Relaxation to Beam Search. NAACL-HLT (1) 2019: 1724-1733 (Goyal et al. (2019))
- Lifu Tu, Kevin Gimpel: Learning Approximate Inference Networks for Structured Prediction. ICLR (Poster) 2018 (Tu and Gimpel (2018))
- Lifu Tu, Kevin Gimpel: Benchmarking Approximate Inference Methods for Neural Structured Prediction. NAACL-HLT (1) 2019: 3313-3324 (Tu and Gimpel (2019))

# References


Goyal, K., Dyer, C., and Berg-Kirkpatrick, T. (2019). An empirical investigation of global and local normalization for recurrent neural sequence models using a continuous relaxation to beam search. In *NAACL-HLT (1)*, pages 1724–1733. Association for Computational Linguistics.

Goyal, K., Neubig, G., Dyer, C., and Berg-Kirkpatrick, T. (2018). A continuous relaxation of beam search for end-to-end training of neural sequence models. In *AAAI*, pages 3045–3052. AAAI Press.

Palmer, M., Kingsbury, P., and Gildea, D. (2005). The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Tu, L. and Gimpel, K. (2018). Learning approximate inference networks for structured prediction. In *ICLR (Poster)*. OpenReview.net.

Tu, L. and Gimpel, K. (2019). Benchmarking approximate inference methods for neural structured prediction. In *NAACL-HLT (1)*, pages 3313–3324. Association for Computational Linguistics.