Advanced Natural Language Processing

Lluís Màrquez

lluism@lsi.upc.edu

Master on Artificial Intelligence

Software Department (LSI)

Technical University of Catalonia (UPC)

March 7, 2013

Document composed using: **pdflatex**, **ppower4**, **xfig** (with multi meta post format), **mpost**

Overview

- **Sequential Modeling**
  - ⋆ Generative Models: HMM
  - ⋆ Sequential Inference with Classifiers
  - ⋆ Maximum Entropy Markov Models
  - ⋆ Conditional Random Fields
  - ⋆ Structured Perceptron and SVMs

Sequential NLP Tasks

Part–of–Speech Tagging

The San Francisco Examiner issued a special edition around noon yesterday that was filled entirely with earthquake new and information.

Sequential NLP Tasks

Part–of–Speech Tagging

The_DT San_NNP Francisco_NNP Examiner_NNP issued_VBD a_DT special_JJ edition_NN around_IN noon_NN yesterday_NN that_WDT was_VBD filled_VBN entirely_RB with_IN earthquake_NN news_NN and_CC information_NN ._.

Sequential NLP Tasks

Part–of–Speech Tagging

The_DT San_NNP Francisco_NNP Examiner_NNP issued_VBD a_DT special_JJ edition_NN around_IN noon_NN yesterday_NN that_WDT was_VBD filled_VBN entirely_RB with_IN earthquake_NN news_NN and_CC information_NN ._.

POS tagging is a pure sequential labeling problem

(sequential learning paradigm)

Sequential NLP Tasks

Shallow Parsing (Chunking)

He reckons the current account deficit will narrow to only 1.8 billion in September.

Sequential NLP Tasks

Shallow Parsing (Chunking)

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow] [PP to] [NP only 1.8 billion] [PP in] [NP September] .

Sequential NLP Tasks

Shallow Parsing (Chunking)

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow] [PP to] [NP only 1.8 billion] [PP in] [NP September] .

Chunking is a sequential phrase recognition task

It can be seen as a sequential labeling problem (B-I-O encoding)

He_B-NP reckons_B-VP the_B-NP current_I-NP account_I-NP deficit_I-NP will_B-VP narrow_I-VP to_B-PP only_B-NP 1.8_I-NP billion_I-NP in_B-PP September_B-NP ._O

Overview

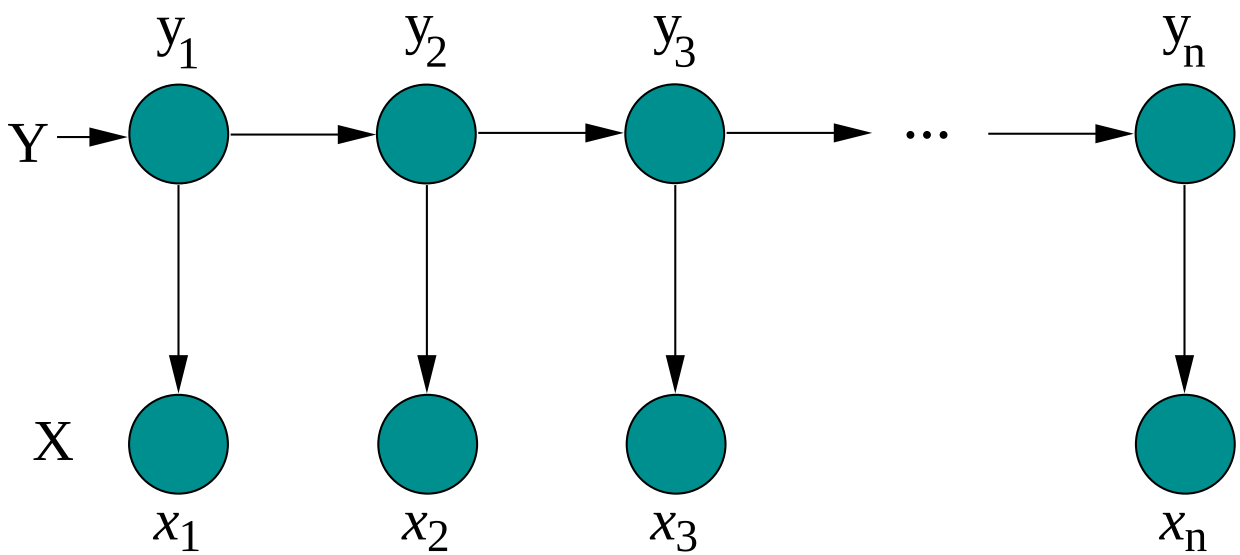
- **Sequential Modeling**
  - ⋆ **Generative Models: HMM**
  - ⋆ Sequential Inference with Classifiers
  - ⋆ Maximum Entropy Markov Models
  - ⋆ Conditional Random Fields
  - ⋆ Structured Perceptron and SVMs

Generative Learning: Models

- Probabilistic models that define a joint probability distribution of the data: $p(\mathcal{X}, \mathcal{Y})$.
- The model is associated to a stochastic generation mechanism of the data, such as an automaton or grammar
- The graphical model underlying the generative mechanism is topologically sorted so as $\mathcal{X}$ variables never preceed $\mathcal{Y}$ variables

Generative Learning: Models

Graphical Model corresponding to a HMM



- Paradigmatic models to recognize structure:
  - ⋆ Hidden Markov Models, e.g. **[Rabiner 89]**
  - ⋆ Probabilistic Context-Free Grammars, e.g. **[Collins 99]**

Generative Learning: Max-Likelihood Estimation

- Based on theory of probability and Bayesian learning:
- Training: via Maximum Likelihood, i.e., simple counts on the training data (very fast; but smoothing is needed)
- Inference Algorithms: efficient algorithms using dynamic programming e.g., Viterbi, CKY, etc.

Generative Models: HMM's

- Generation mechanism: probabilistic automaton with outputs
- Sequences of observations: $\{x_1, \ldots, x_n\}$ and states $\{y_1, \ldots, y_n\}$
- Assumptions: limited horizon (Markov order)
  x_i only depends on y_i

Generative Models: HMM's

- Generation mechanism: probabilistic automaton with outputs
- Sequences of observations: $\{x_1, \ldots, x_n\}$ and states $\{y_1, \ldots, y_n\}$
- Assumptions: limited horizon (Markov order)
  x_i only depends on y_i
- Objective function: $\arg\max_{y_1,\ldots,y_n} P(y_1, \ldots, y_n | x_1, \ldots, x_n) =$

$$\arg\max_{y_1,\ldots,y_n} \frac{P(x_1,\ldots,x_n|y_1,\ldots,y_n) \cdot P(y_1,\ldots,y_n)}{P(x_1,\ldots,x_n)} \approx$$

Generative Models: HMM's

- Generation mechanism: probabilistic automaton with outputs
- Sequences of observations: $\{x_1, \ldots, x_n\}$ and states $\{y_1, \ldots, y_n\}$
- Assumptions: limited horizon (Markov order)

  x_i only depends on y_i
- Objective function: $\arg\max_{y_1,\ldots,y_n} P(y_1, \ldots, y_n | x_1, \ldots, x_n) =$

$$\arg\max_{y_1,\ldots,y_n} \frac{P(x_1,\ldots,x_n|y_1,\ldots,y_n) \cdot P(y_1,\ldots,y_n)}{P(x_1,\ldots,x_n)} \approx$$

$$\boxed{\arg\max_{y_1,\ldots,y_n} \prod_{k=1}^{n} P(y_k|y_{k-2}, y_{k-1}) \cdot P(x_k|y_k)}$$

Generative models: HMM's

- $\boxed{\arg\max_{y_1,\ldots,y_n} \prod_{k=1}^{n} P(y_k|y_{k-2}, y_{k-1}) \cdot P(x_k|y_k)}$

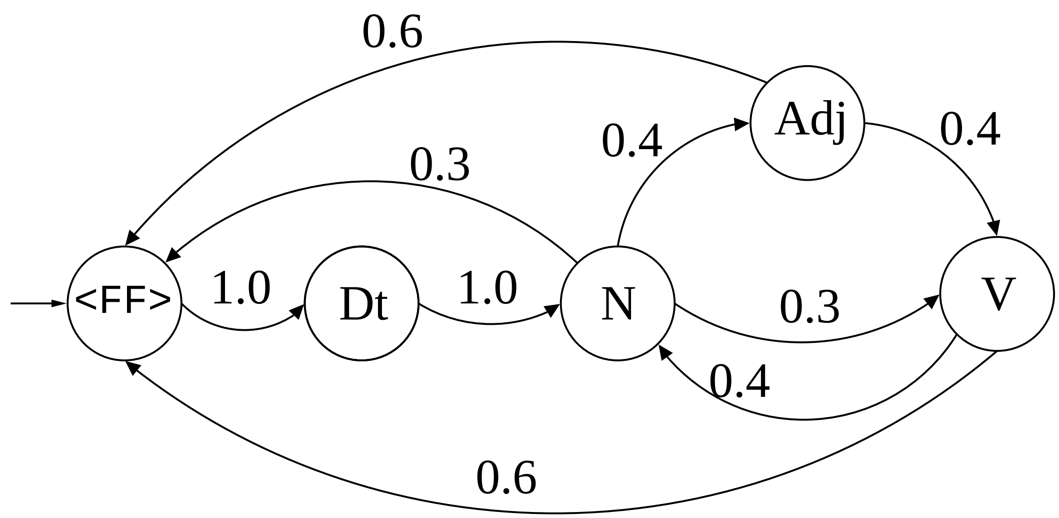
- We need to estimate the following probability distributions:
  - ⋆ emission probabilities: $P(x_k|y_k)$
  - ⋆ transition probabilities: $P(y_k|y_{k-2}, y_{k-1})$ (second order HMM)
  - ⋆ initial state probabilities: $P(y_1)$

- Viterbi algorithm allows to calculate the $\arg\_\max$ in $O(n)$

- But there is a practically important constant factor: $MarkovOrder \times |States|$

Generative Models: HMM example

States and transition probabilities (first order HMM)



Emission probabilities	.	el	la	gato	niña	come	corre	pescado	fresco	pequeña	grande
<FF>	1.0										
Dt		0.6	0.4								
N				0.6	0.1			0.3			
V						0.7	0.3				
Adj									0.3	0.3	0.4

Generative Learning: example on NER

- IdentiFinder$^{\mathrm{TM}}$ **[Bikel, Schwartz and Weischedel 1999]**
- An HMM-based system for Named Entity Recognition, used at MUC conferences
- See complementary slides on IdentiFinder$^{\mathrm{TM}}$ (in PowerPoint)

Pros and Cons

Advantages

- Flexibility to represent complex structures as generative processes
- Under certain simplifying assumptions:
  - ⋆ Simplicity of the training process: fast parameter estimation
  - ⋆ Very efficient decoding algorithms exist

Pros and Cons

Problems

- Training/decoding on complex generative settings is not feasible
- Strong independence assumptions are needed: not necessary in accordance with data

Pros and Cons

Problems

- Training/decoding on complex generative settings is not feasible
- Strong independence assumptions are needed: not necessary in accordance with data
- Difficult to use arbitrary feature representations
  - ⋆ Features are tied to the generation mechanism of the data
  - ⋆ Extending the feature dependendencies imply:
    - ∗ Severe sparsity problems (training is difficult)
    - ∗ Exact decoding may be computationally intractable

Pros and Cons

Problems

- Training/decoding on complex generative settings is not feasible
- Strong independence assumptions are needed: not necessary in accordance with data
- Difficult to use arbitrary feature representations
  - ⋆ Features are tied to the generation mechanism of the data
  - ⋆ Extending the feature dependendencies imply:
    - ∗ Severe sparsity problems (training is difficult)
    - ∗ Exact decoding may be computationally intractable
  - ⋆ Feature specialization is possible but in a limited way

Overview

- **Sequential Modeling**
  - ⋆ Generative Models: HMM
  - ⋆ **Sequential Inference with Classifiers**
  - ⋆ Maximum Entropy Markov Models
  - ⋆ Conditional Random Fields
  - ⋆ Structured Perceptron and SVMs

Learning and Inference: General Approach

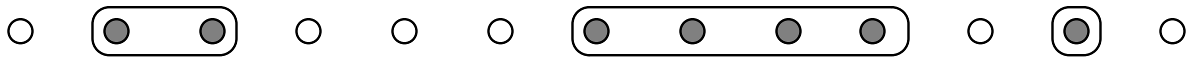
- Transform the recognition problem into a chain of *simple* decisions:
  - ⋆ Segmentation Decisions:
    e.g., Open-Close, Begin-Inside-Outside, Shift-Reduce, etc.
  - ⋆ Labeling Decisions: made during segmentation or afterwards
  - ⋆ Decisions might use the output of earlier steps in the chain

- Set up an inference strategy:
  - ⋆ Decisions are applied in chain to build structure incrementally
  - ⋆ Exploration might be at different levels of amplitude:
    e.g., greedy, dynamic programming, beam search, etc.

- Learn a prediction function for each decision

Learning and Inference: Simple Examples

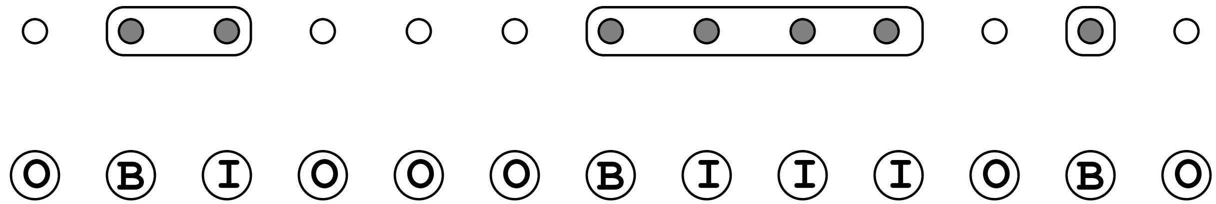
BIO Tagging for Phrase Identification



Learning and Inference: Simple Examples

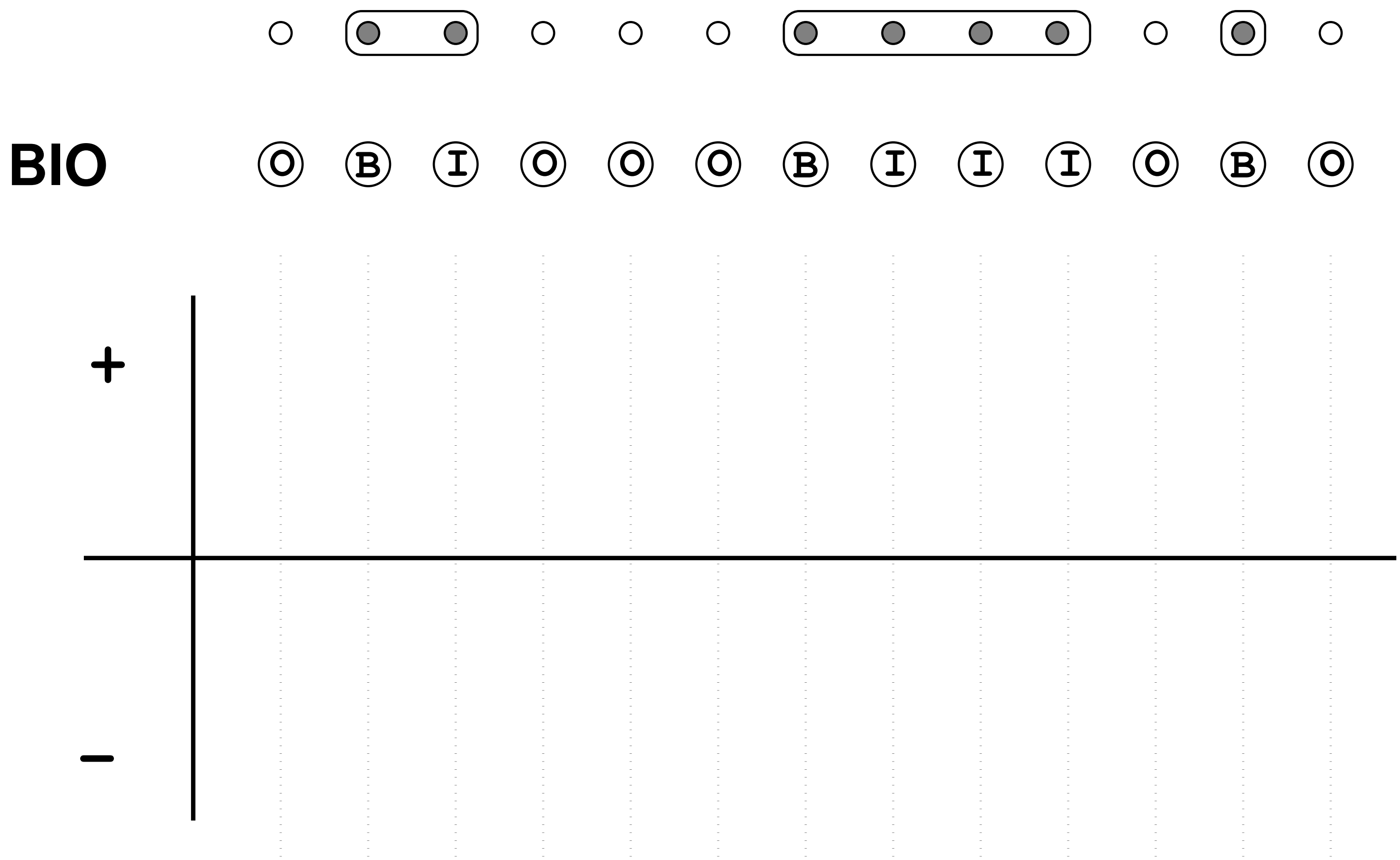
BIO Tagging for Phrase Identification

BIO



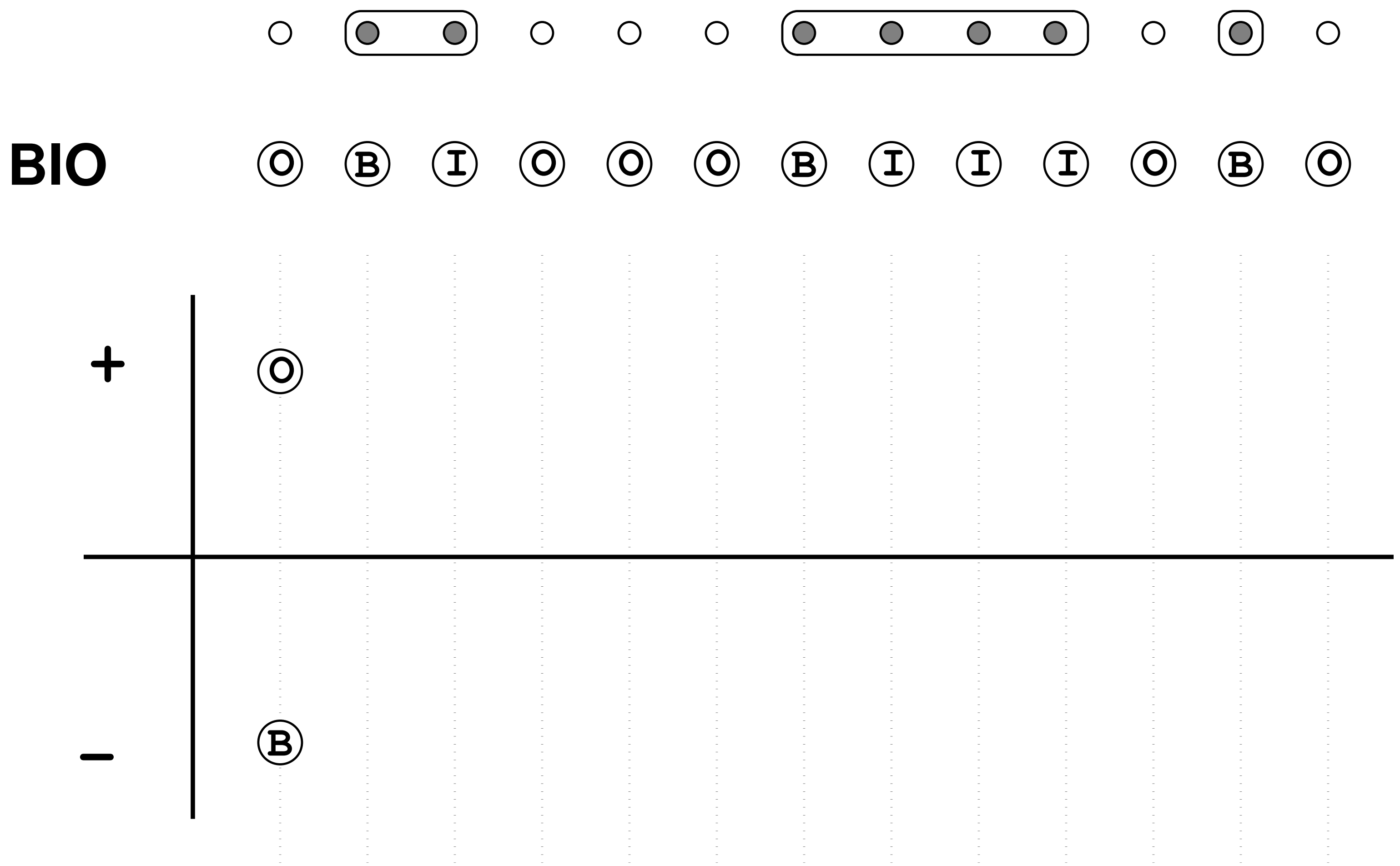
Learning and Inference: Simple Examples

BIO Tagging for Phrase Identification



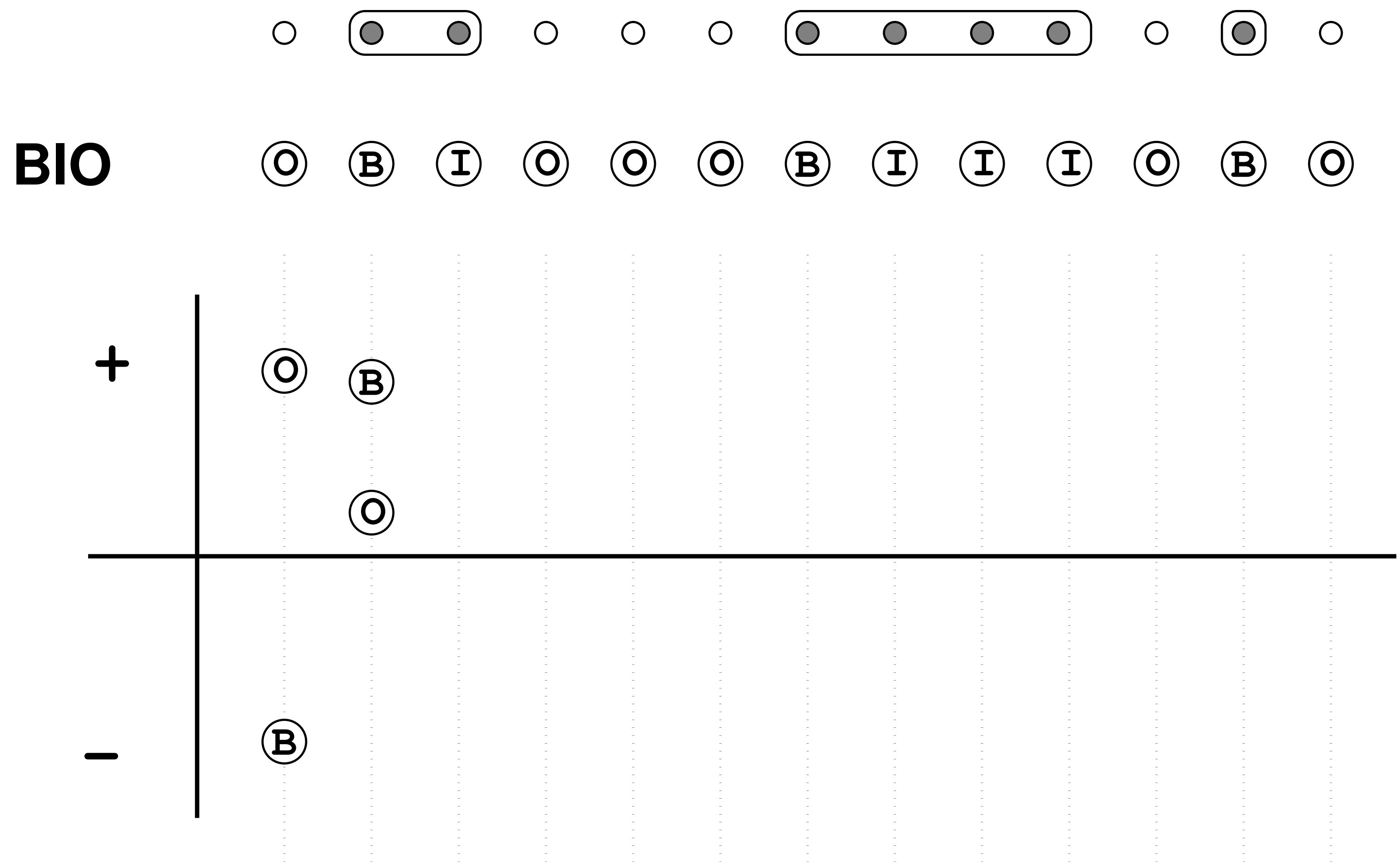
Learning and Inference: Simple Examples

BIO Tagging for Phrase Identification



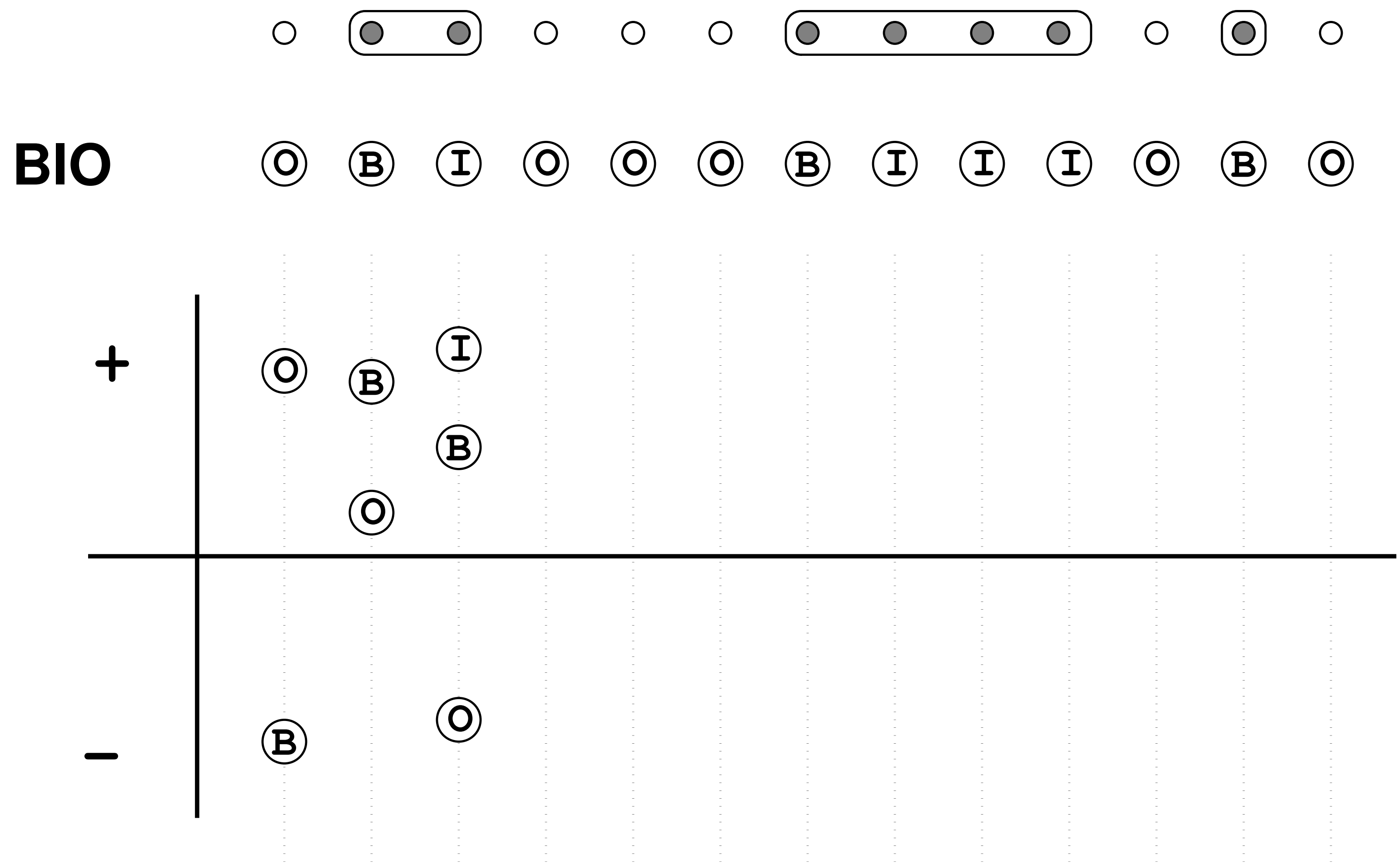
Learning and Inference: Simple Examples

BIO Tagging for Phrase Identification



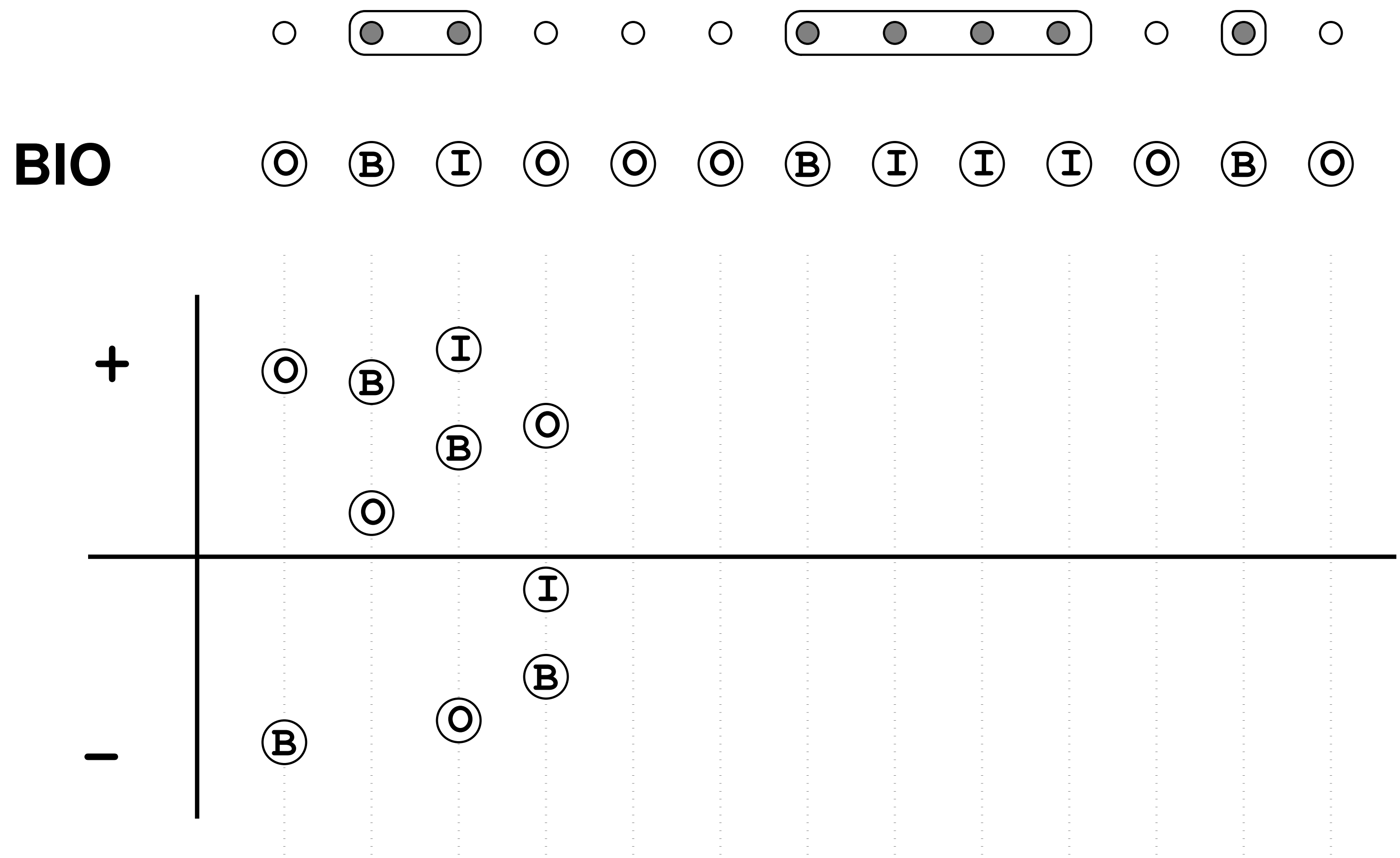
Learning and Inference: Simple Examples

BIO Tagging for Phrase Identification



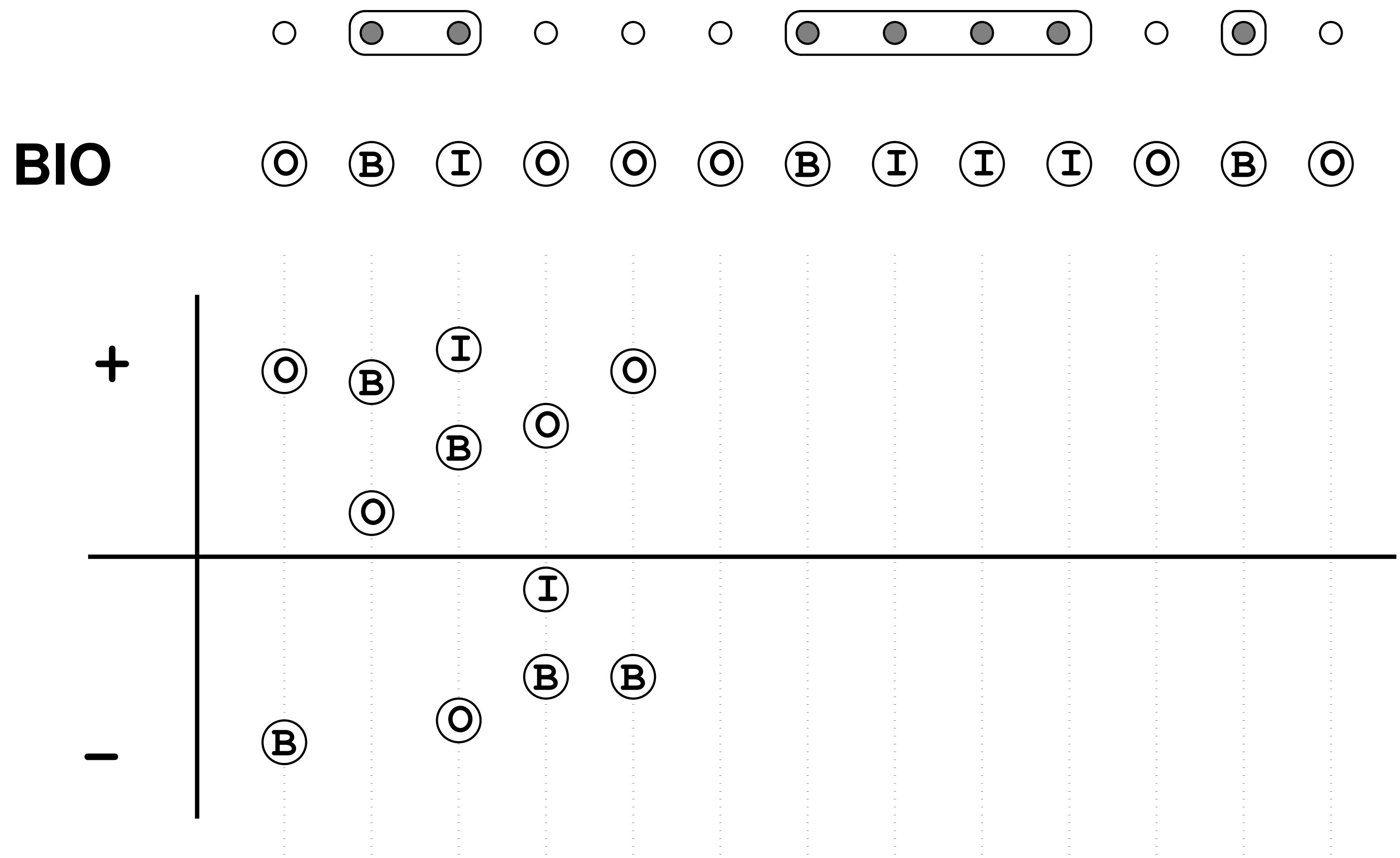
Learning and Inference: Simple Examples

BIO Tagging for Phrase Identification



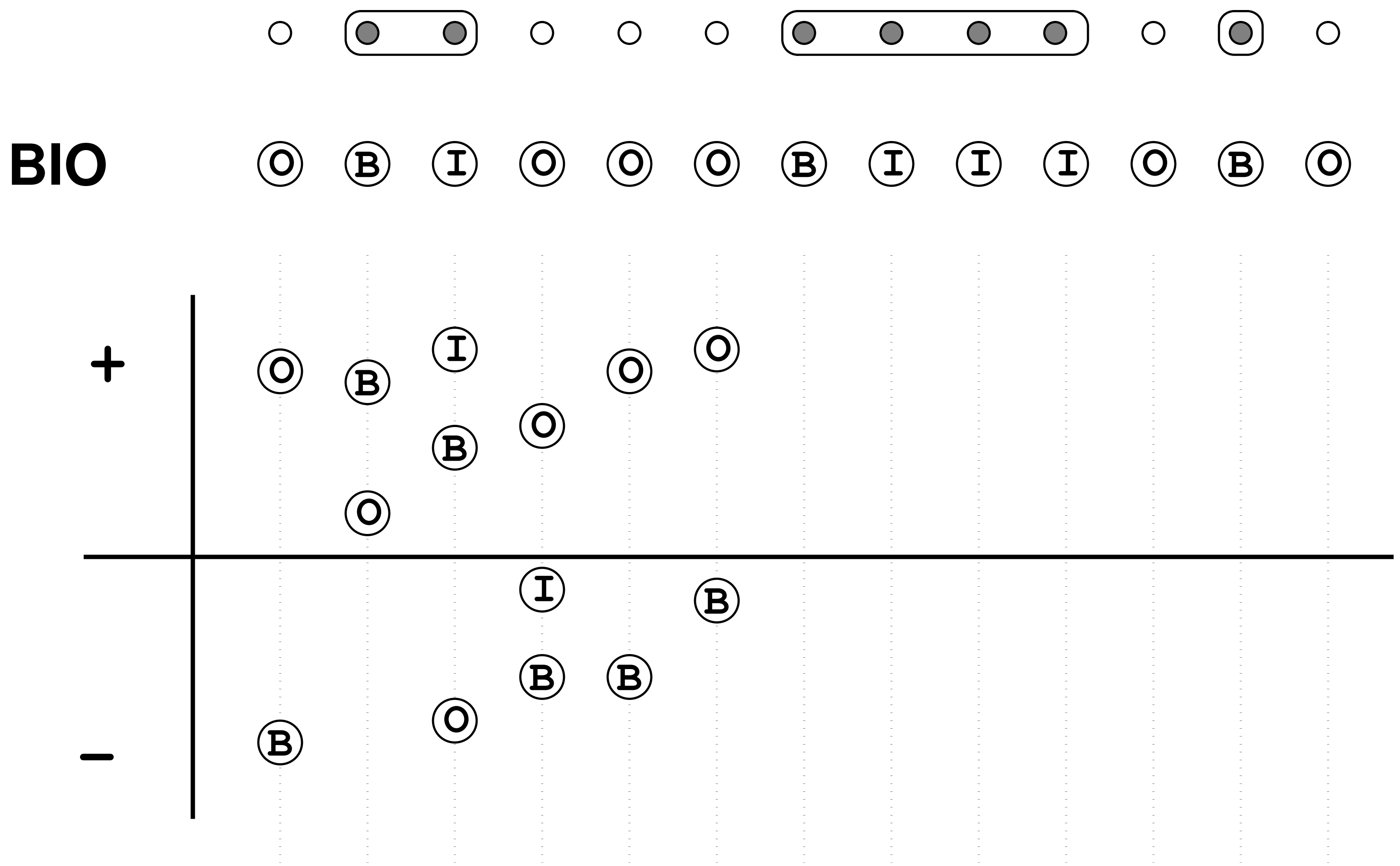
Learning and Inference: Simple Examples

BIO Tagging for Phrase Identification



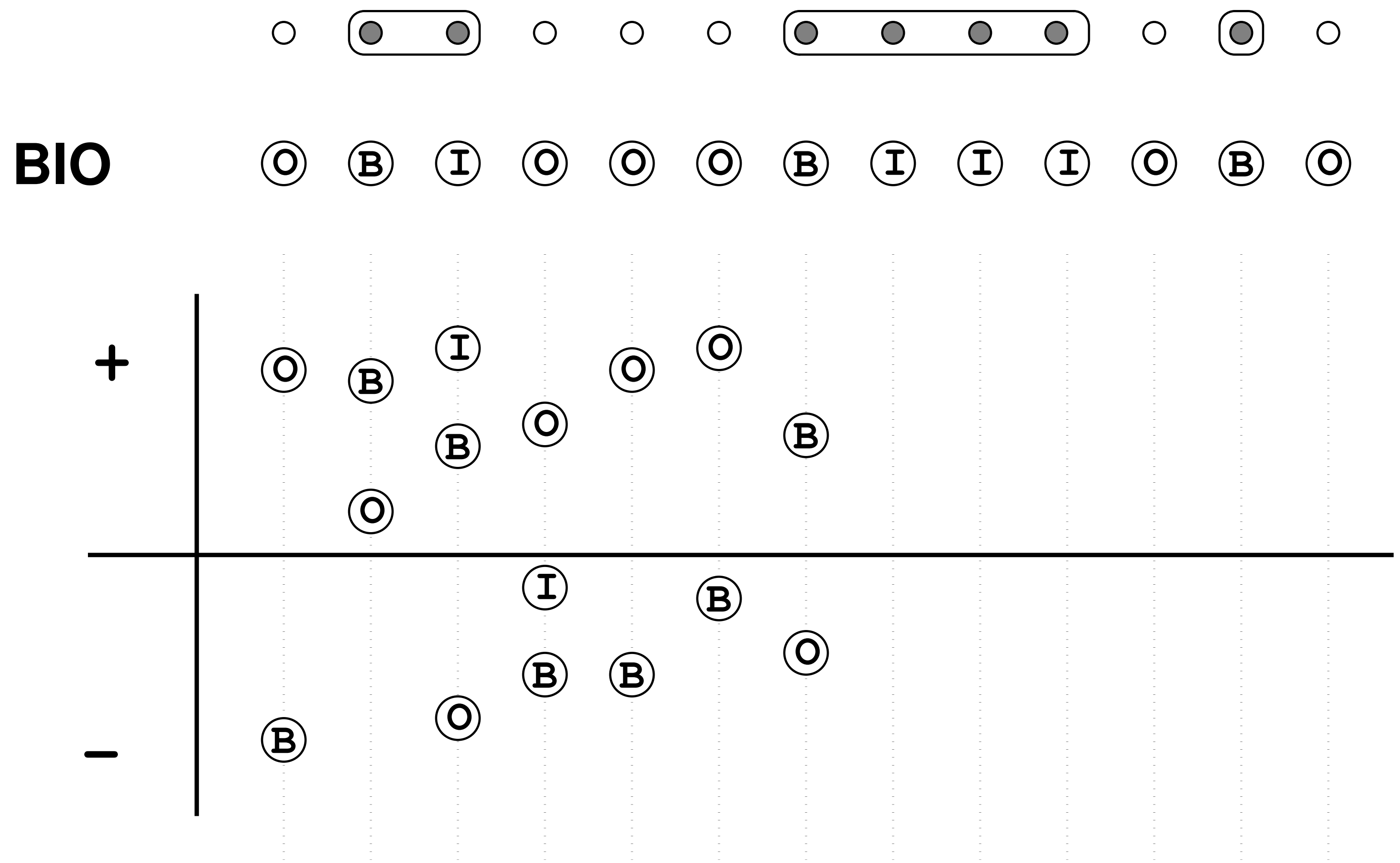
Learning and Inference: Simple Examples

BIO Tagging for Phrase Identification



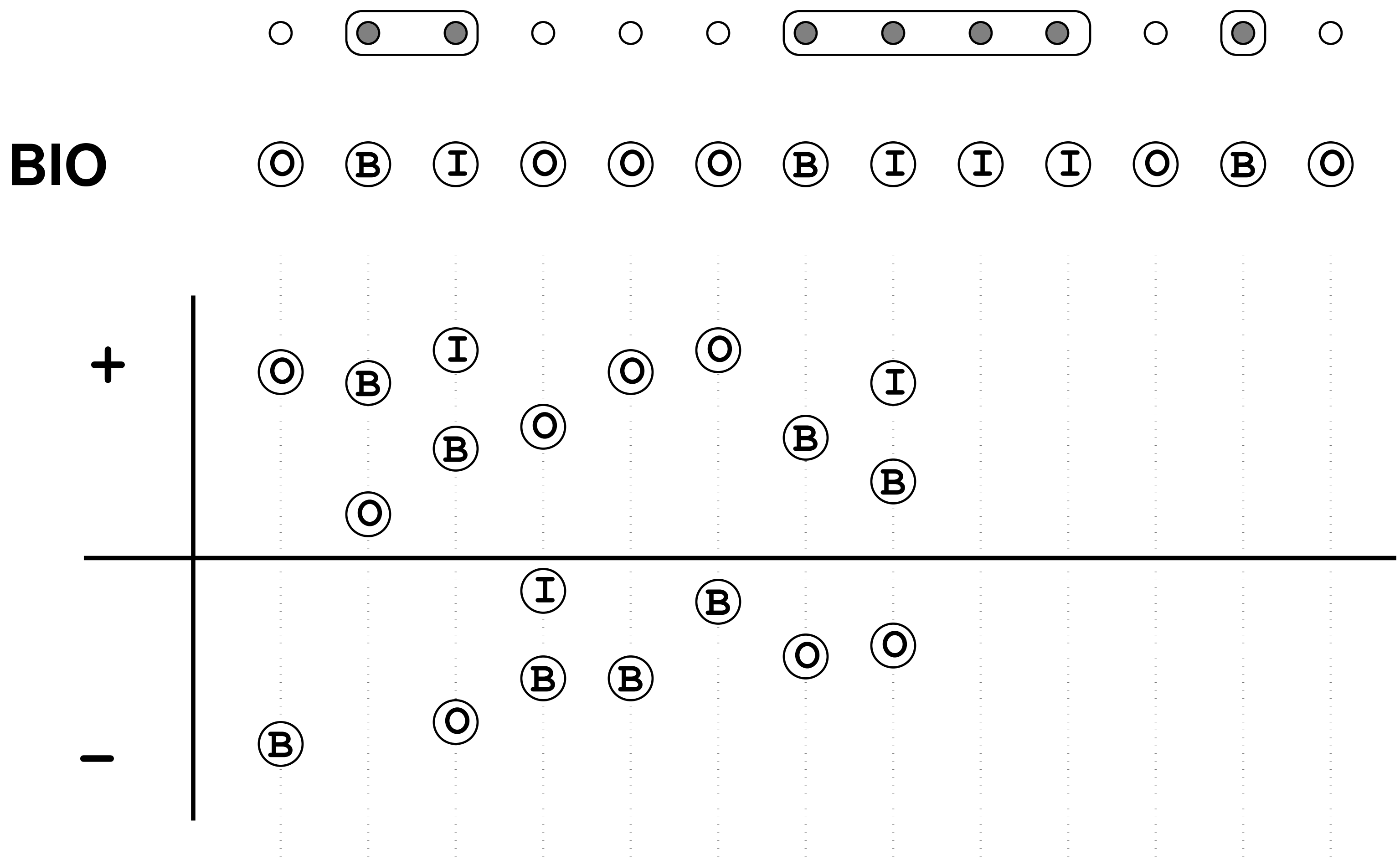
Learning and Inference: Simple Examples

BIO Tagging for Phrase Identification



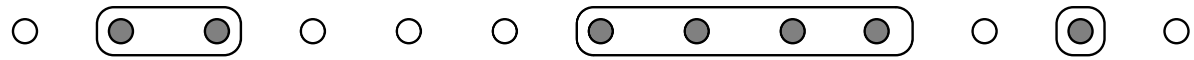
Learning and Inference: Simple Examples

BIO Tagging for Phrase Identification

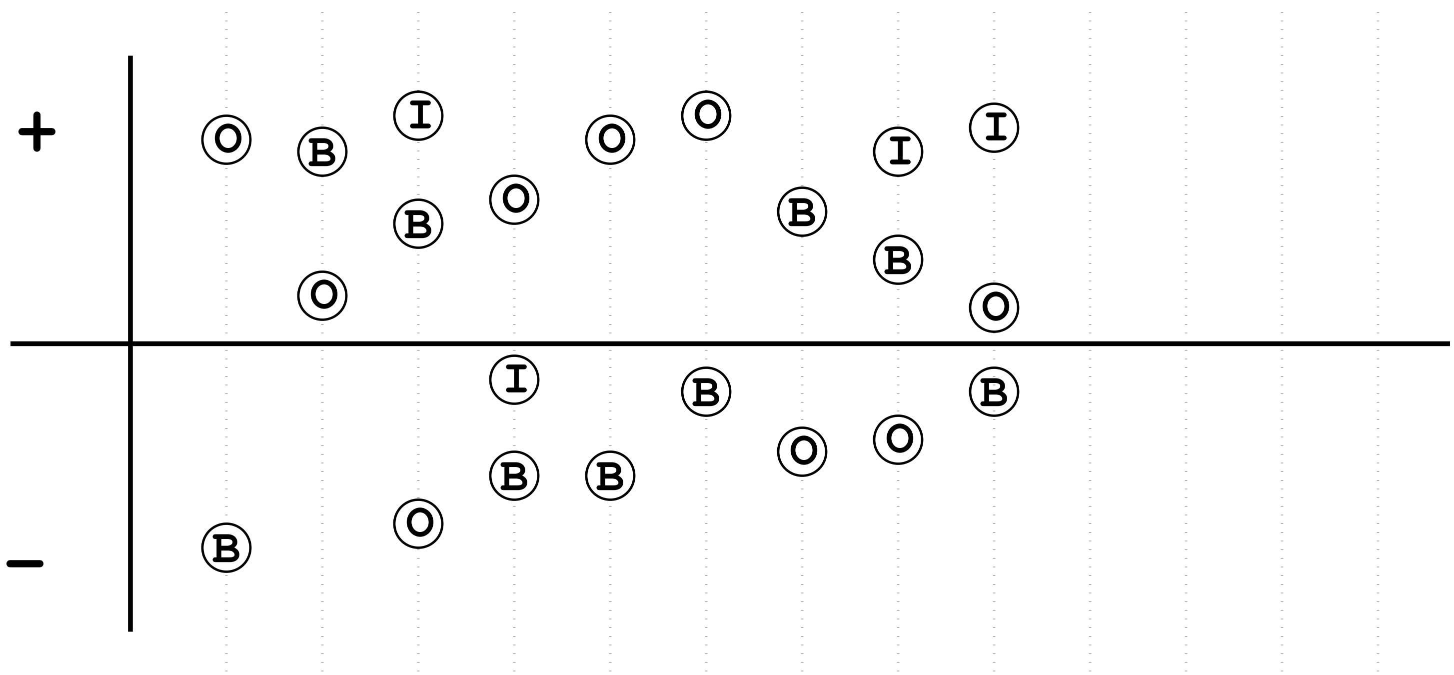


Learning and Inference: Simple Examples

BIO Tagging for Phrase Identification

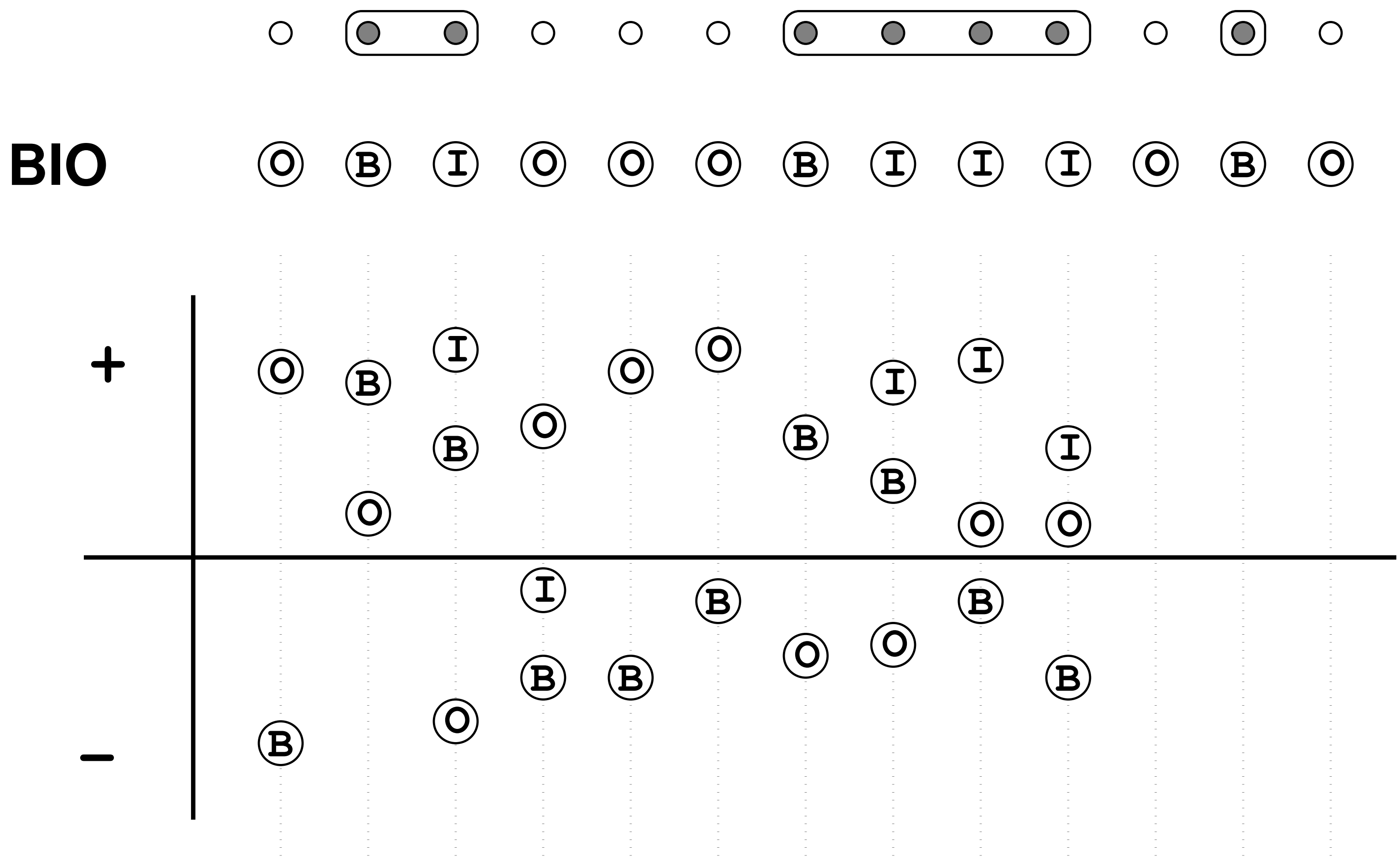


BIO O B I O O O B I I I O B O



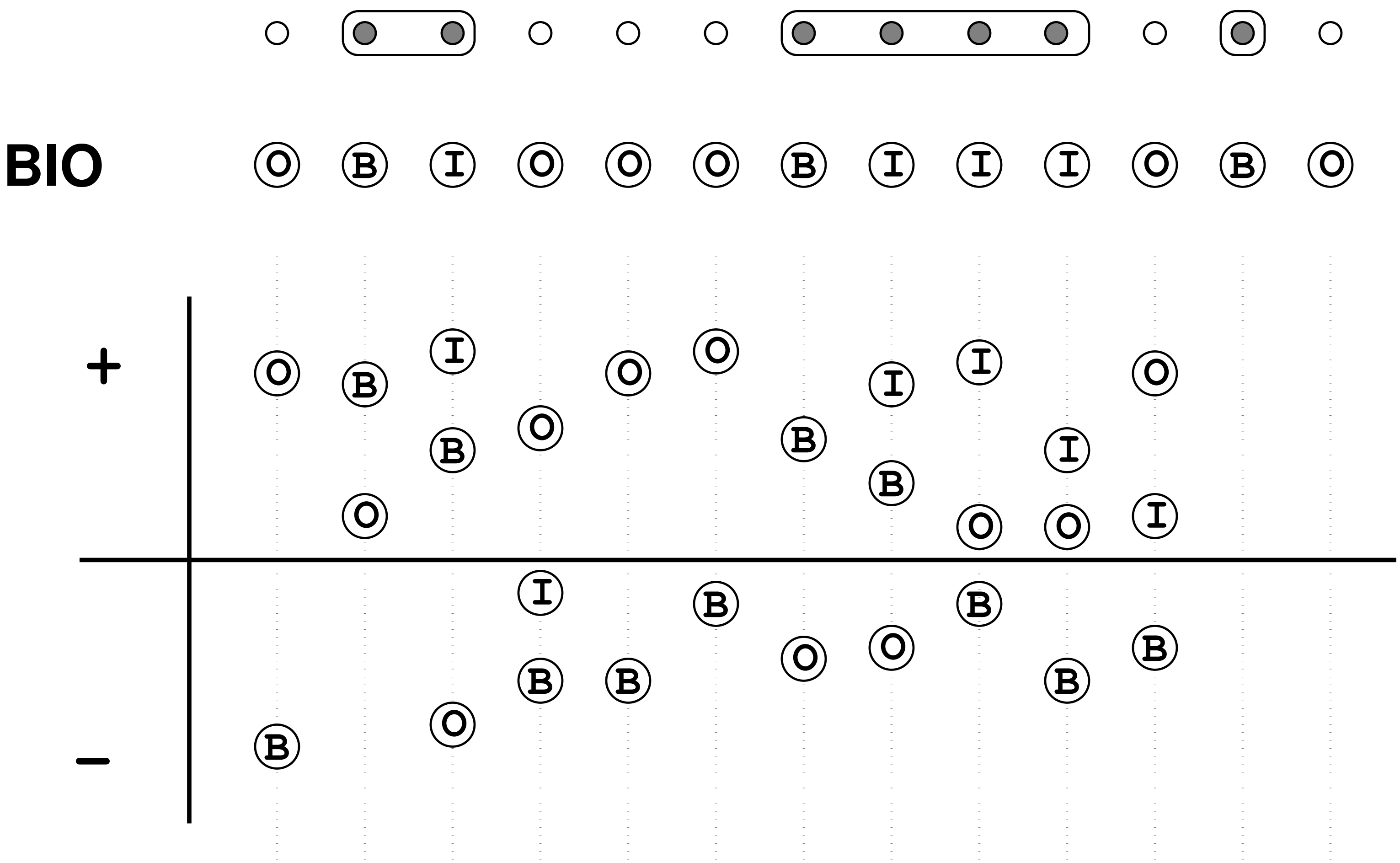
Learning and Inference: Simple Examples

BIO Tagging for Phrase Identification



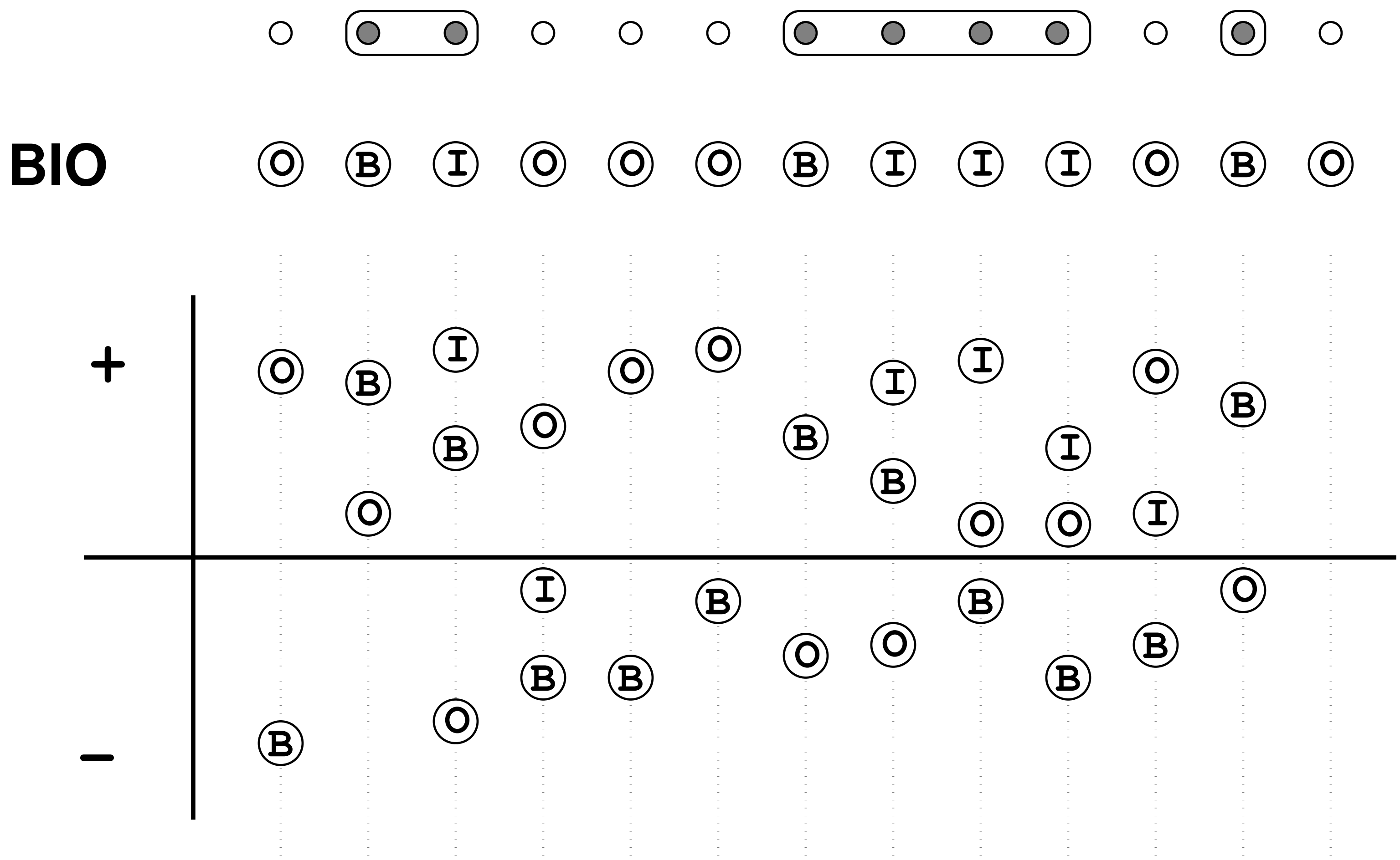
Learning and Inference: Simple Examples

BIO Tagging for Phrase Identification



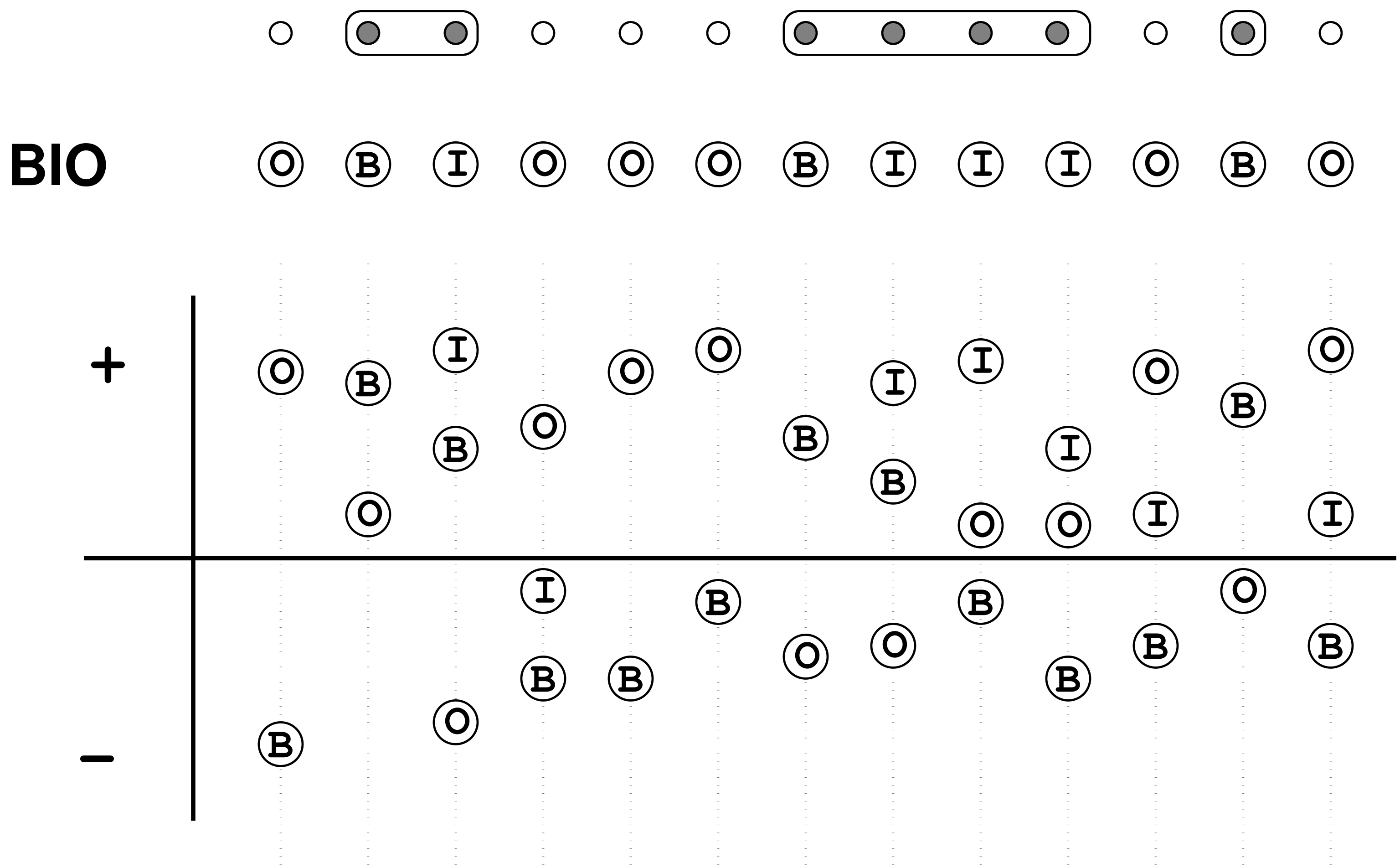
Learning and Inference: Simple Examples

BIO Tagging for Phrase Identification



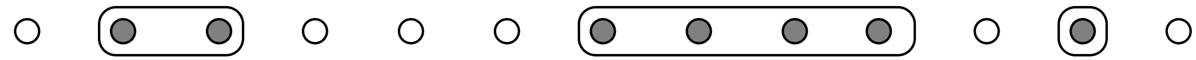
Learning and Inference: Simple Examples

BIO Tagging for Phrase Identification



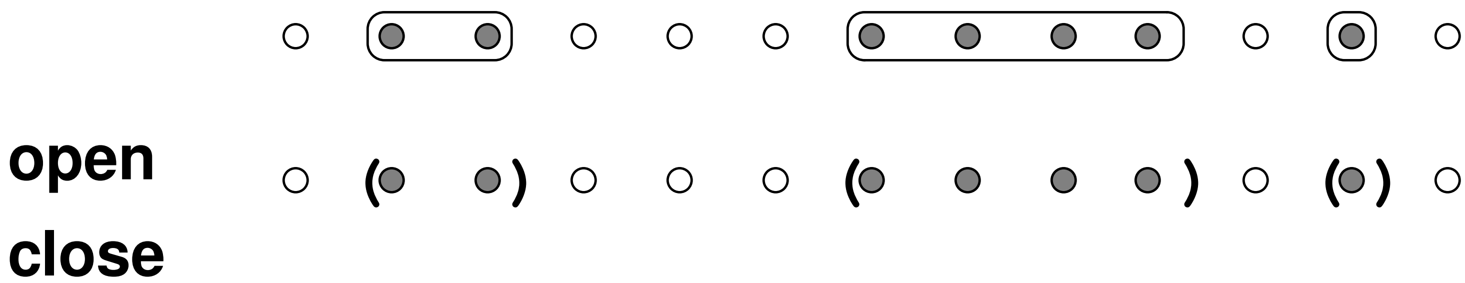
Learning and Inference: Simple Examples

Open-Close for Phrase Identification



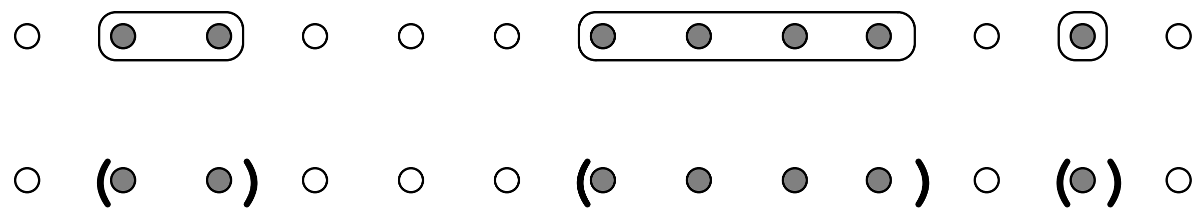
Learning and Inference: Simple Examples

Open-Close for Phrase Identification



Learning and Inference: Simple Examples

Open-Close for Phrase Identification



open

close

`open`

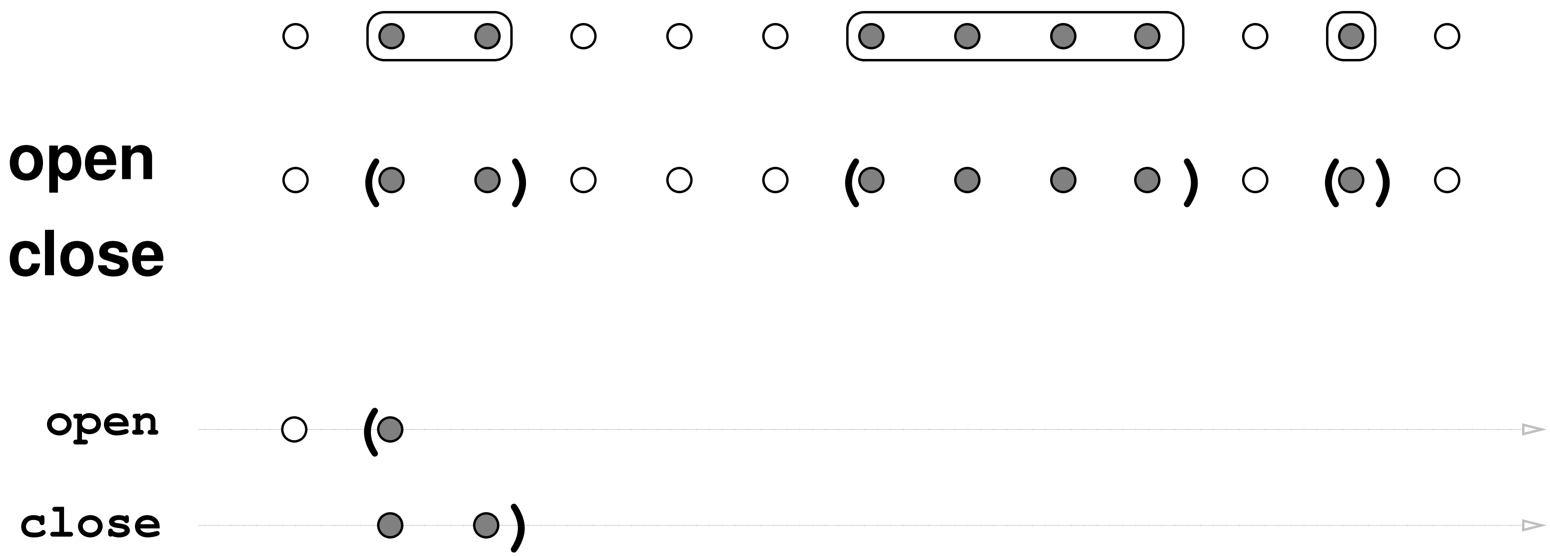
`close`

Learning and Inference: Simple Examples

Open-Close for Phrase Identification

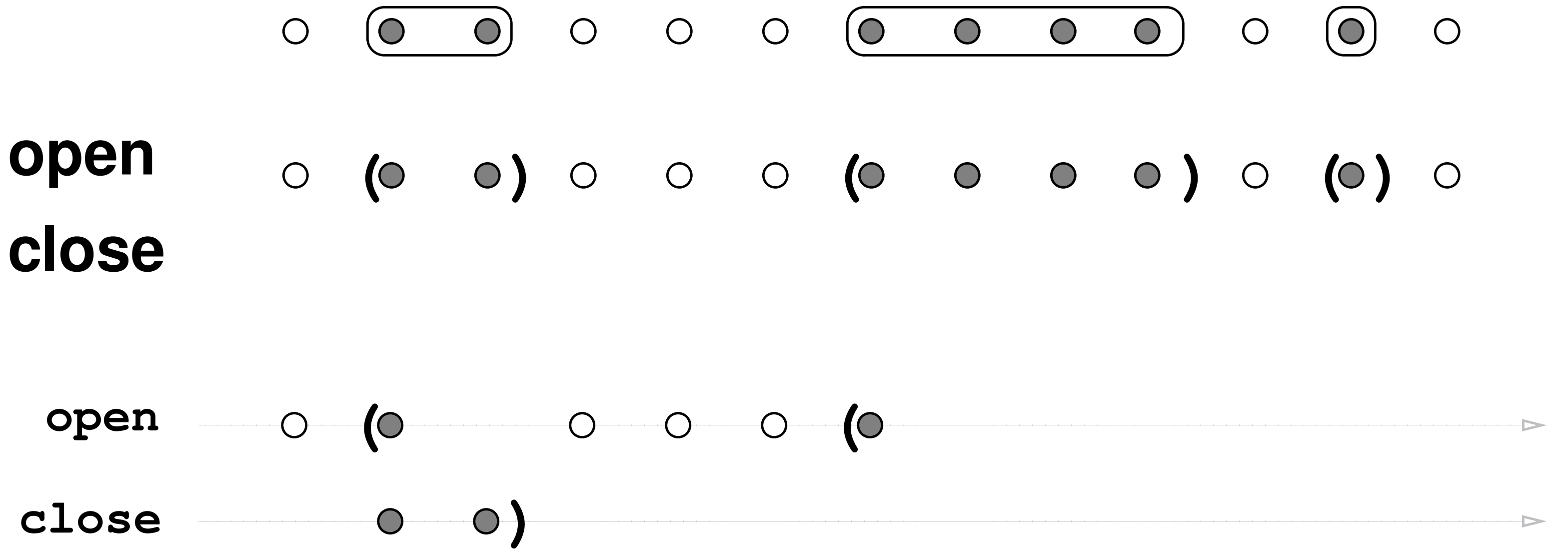
Learning and Inference: Simple Examples

Open-Close for Phrase Identification



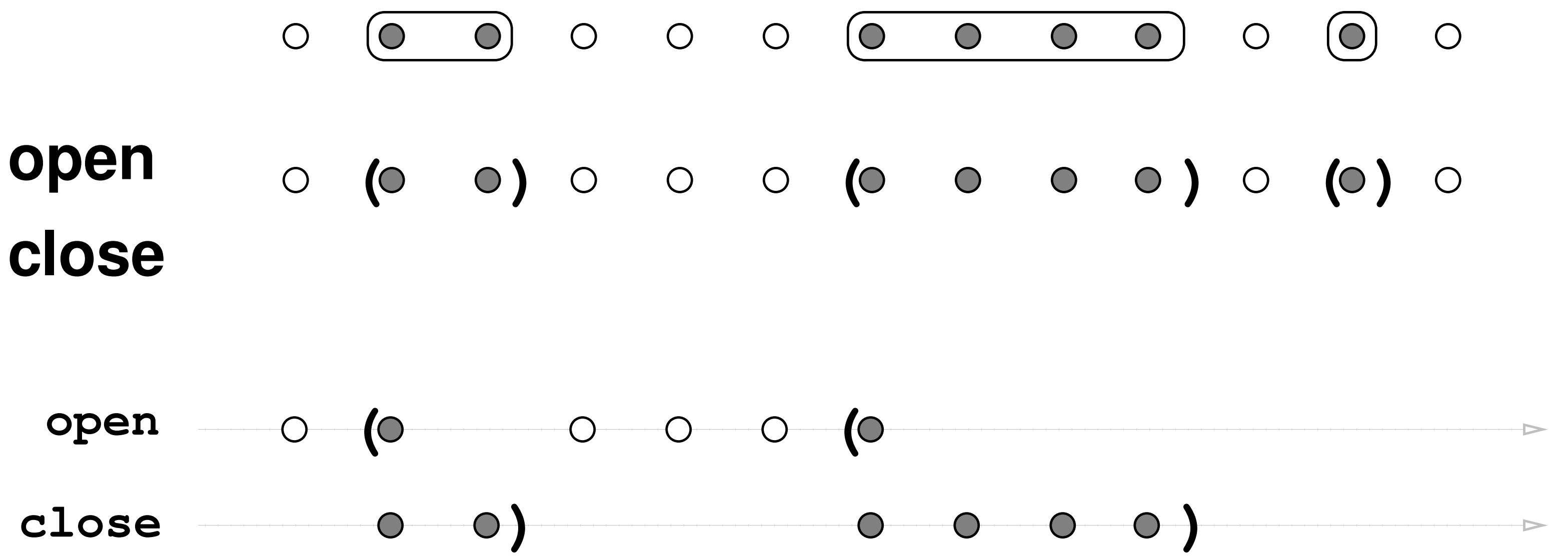
Learning and Inference: Simple Examples

Open-Close for Phrase Identification



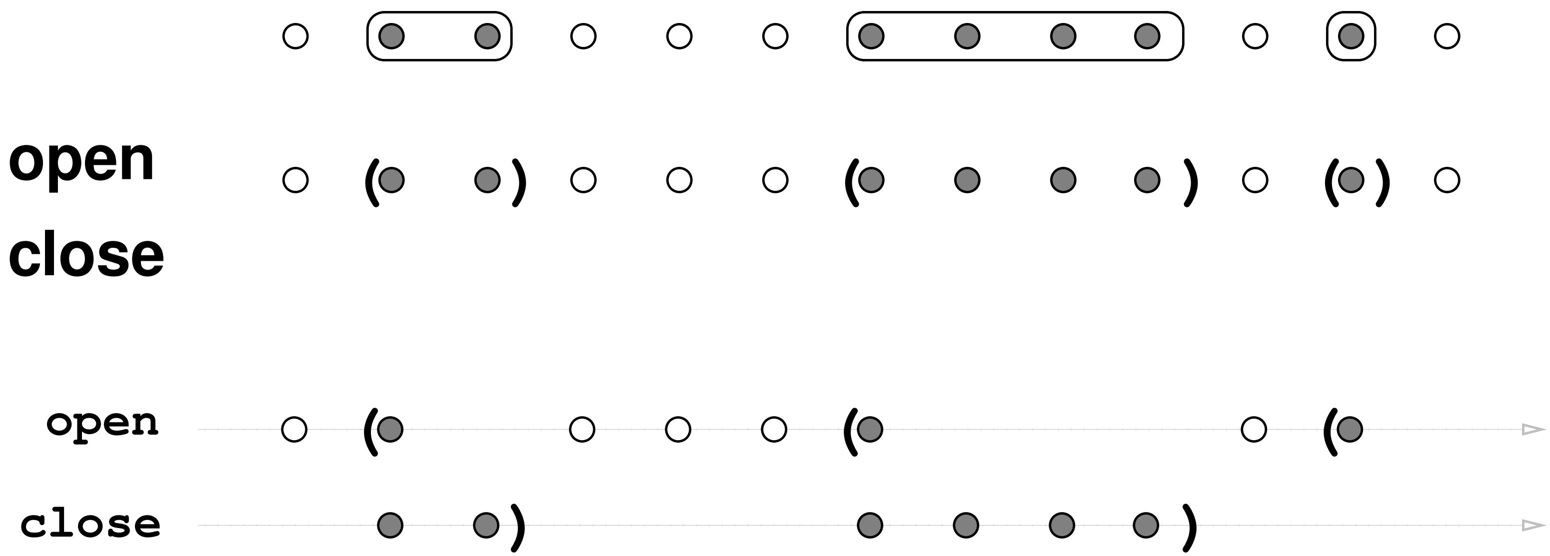
Learning and Inference: Simple Examples

Open-Close for Phrase Identification



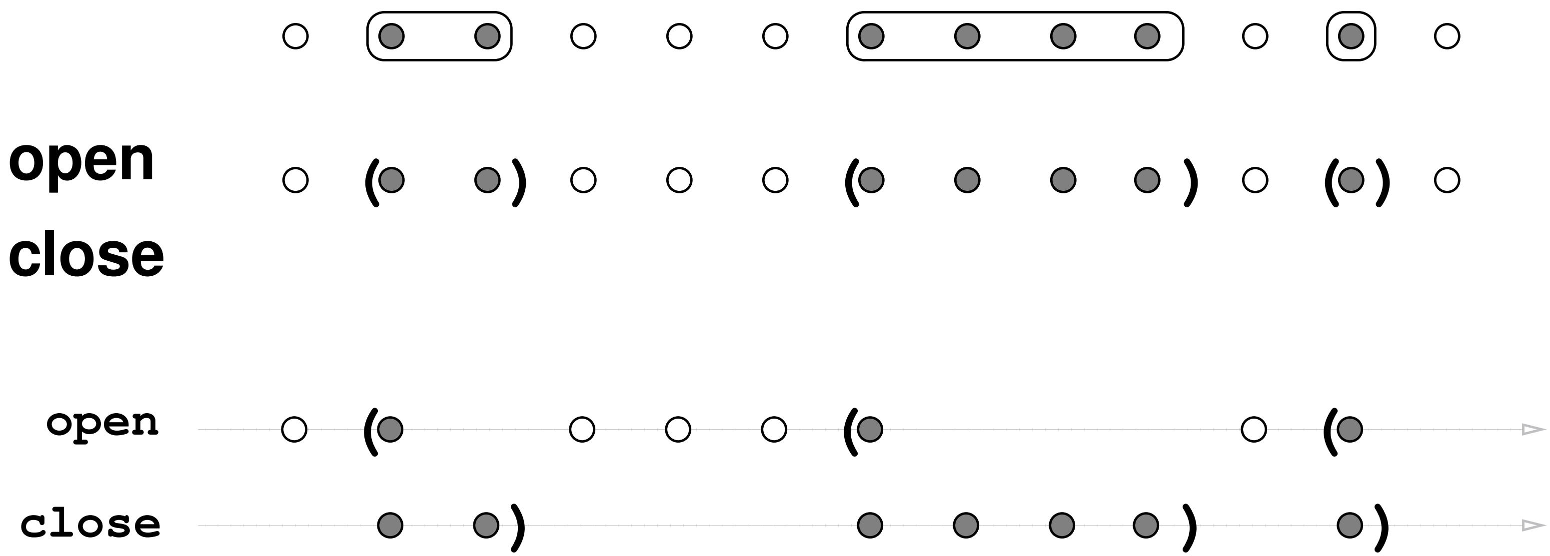
Learning and Inference: Simple Examples

Open-Close for Phrase Identification



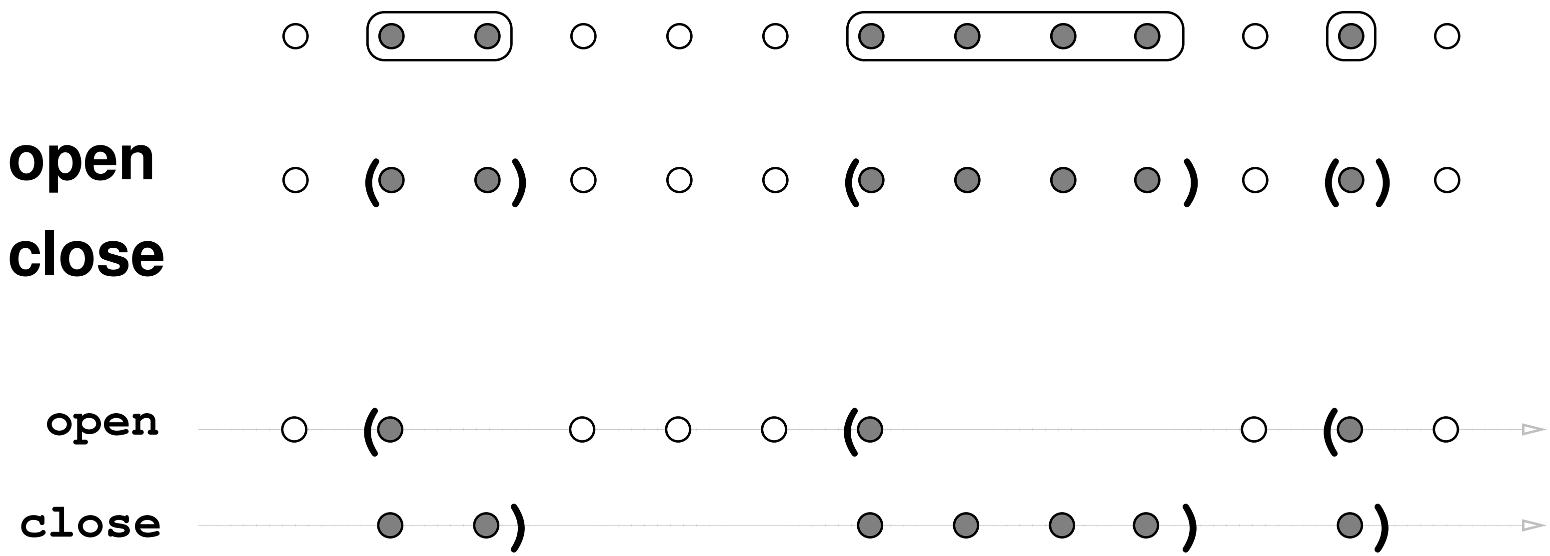
Learning and Inference: Simple Examples

Open-Close for Phrase Identification



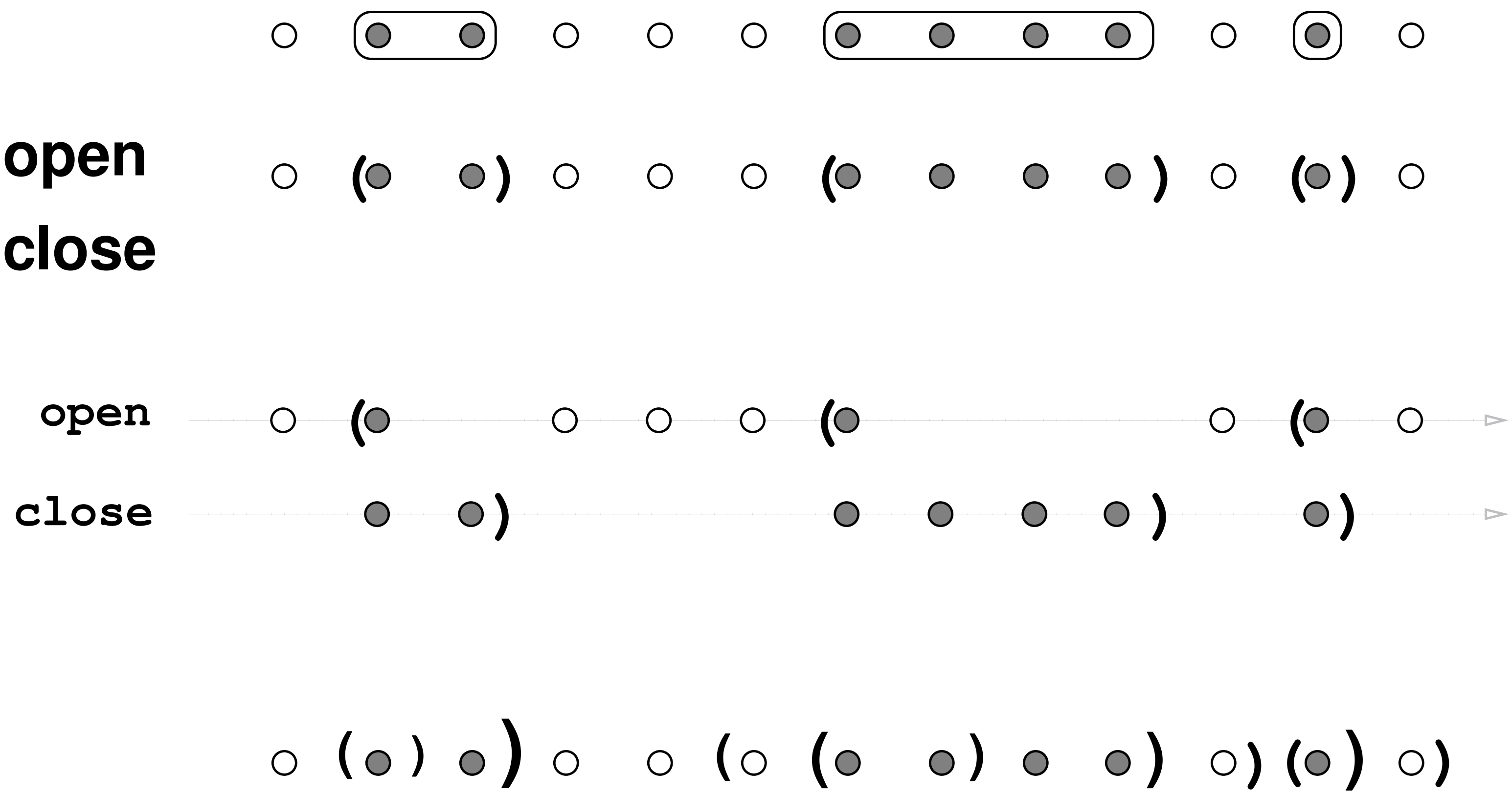
Learning and Inference: Simple Examples

Open-Close for Phrase Identification



Learning and Inference: Simple Examples

Open-Close for Phrase Identification



Learning and Inference (Roth et al.)

- **Divide and Conquer** strategy:
  - ⋆ Decomposition into a number of local decisions to learn (you can use any classifier that output confidence scores)
  - ⋆ Inference scheme to construct the solution on top of classifiers' predictions; possibly including constraints given by the problem **[Punyakanok and Roth, 2001; 2004; Yih and Roth, 2004]**

Sequential Phrase Identification

- Formalization and proposal of three decompositions and exact inference procedures **[Punyakanok & Roth, 2001; 2004]**

- **HMM with classifiers**:
  - ⋆ HMM: $P(y_1)$, $P(y_t|y_{t-1})$, $P(x_t|y_t)$
  - ⋆ $P(x_t|y_t) = \frac{P(y_t|x_t)P(x_t)}{P(y_t)}$
  - ⋆ Classifiers provide $P(y_t|x_t)$
  - ⋆ Actually, it is extended to $P(y_t|\hat{x}_t)$
  - ⋆ The objective function is exactly the same than in regular HMM's. Inference is done by using the Viterbi decoder

Sequential Phrase Identification

[Punyakanok & Roth, 2001; 2004]

- **Projective Markov Models (PMM)**:
  - ⋆ Classifiers directly estimate $P(y_t|y_{t-1}, \hat{x}_t)$
  - ⋆ Optionally, train:
    - ∗ a binary classifier for each pair (y_t, y_{t-1})
    - ∗ a binary classifier for each y including features on y_{t-1}
    - ∗ a single multiclass classifier including features on y_{t-1}
  - ⋆ Convert output scores in true probabilities (e.g., using $\mathrm{softmax}$)
  - ⋆ The objective functions is: $\arg\max_{y_1,\ldots,y_n} \prod_{k=1}^{n} P(y_t|y_{t-1}, \hat{x}_t)$
  - ⋆ The inference is again the Viterbi decoder

Sequential Phrase Identification

[Punyakanok & Roth, 2001; 2004]

- **Constraint Satisfaction with classifiers**:
  - ⋆ CSP problem casted as a DAG based on open-close
  - ⋆ Classifiers provide confidence on open and close decisions
  - ⋆ The inference is the *shortest path* algorithm
- **Empirical Results on the chunking task:**

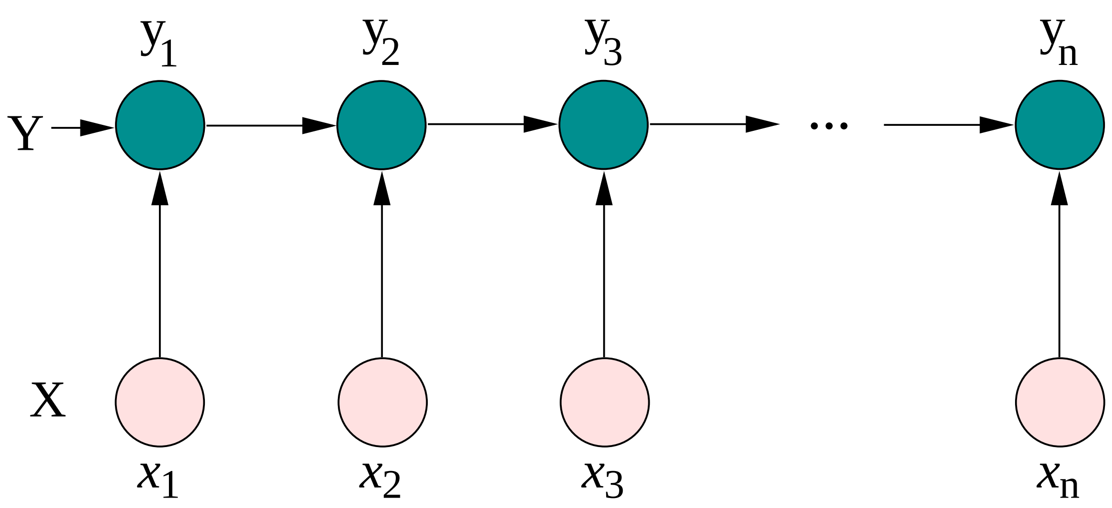
$$\textbf{HMM} < \textbf{HMM+class} < \textbf{PMM} \approx \textbf{CS+class}$$

Sequential Phrase Identification

[Punyakanok & Roth, 2001; 2004]

- **Note$_1$**
  - ⋆ A PMM is also called Conditional Markov Model. Other examples using Maximum Entropy (MEMM) **[Ratnaparkhi 1996; 1999; McCallum et al.,2000]**

Graphical Model corresponding to a MEMM

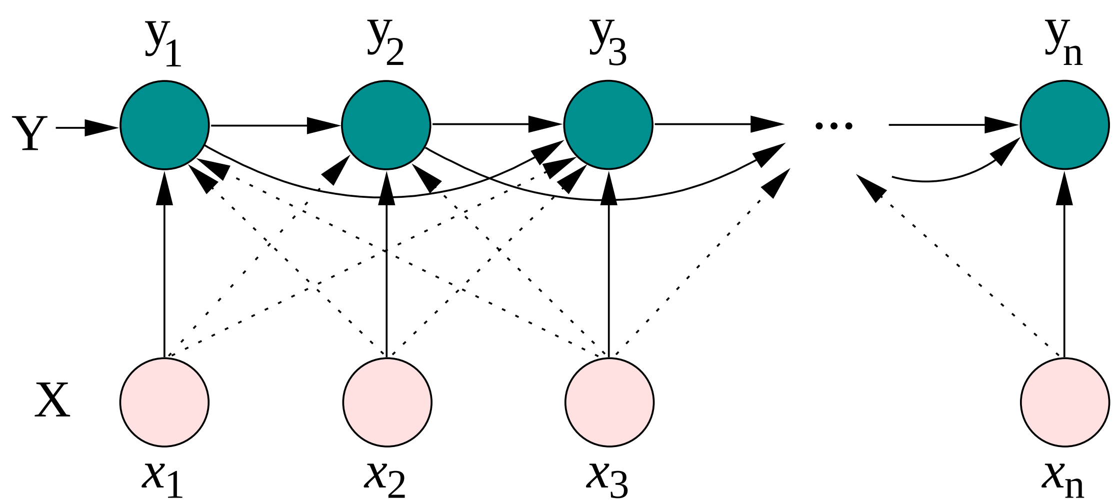


Sequential Phrase Identification

[Punyakanok & Roth, 2001; 2004]

- **Note$_1$**
  - ⋆ A PMM is also called Conditional Markov Model. Other examples using Maximum Entropy (MEMM) **[Ratnaparkhi 1996; 1999; McCallum et al.,2000]**

Graphical Model corresponding to a MEMM



Generalized Inference with Classifiers

Extension of the previous work

- Work with general constraints, not only structural
  - ⋆ Joint recognition of Named Entities and Relations **[Yih and Roth, 2004]**. See slides on that paper
  - ⋆ Application to Semantic Role Labeling **[Punyakanok et al., 2005]**. See the survey on SRL
- Modeled as optimization with integer linear constraints
  - ⋆ Flexible to model many NLP processes (e.g., parsing)
  - ⋆ Solved using Integer Linear Programming
  - ⋆ Exact inference is feasible in practice