

Lecture 12: Discriminative Sequence Tagging

Nathan Schneider
ENLP | 27 February 2019

HMM + features

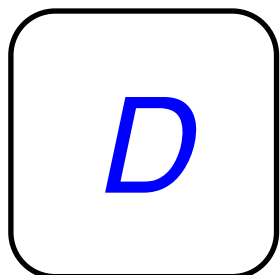
- There are variants of the generative HMM that emit features instead of just words.
- However, these suffer from similar problems as features in naïve Bayes (too strong independence assumptions).
- Can we be **discriminative** instead?
 - Yes! In fact, we can reuse the same machinery for discriminative learning with **linear models**.

Recasting HMM as a Linear Model

- Recall that a linear model is one that scores candidate outputs y with $\mathbf{w}^\top \boldsymbol{\varphi}(\mathbf{x}, y)$. Decoding = $\arg \max_{y'} \mathbf{w}^\top \boldsymbol{\varphi}(\mathbf{x}, y')$.
- Not just classification: we can be predicting a structured output \mathbf{y} . Thus $\arg \max_{\mathbf{y}'} \mathbf{w}^\top \boldsymbol{\varphi}(\mathbf{x}, \mathbf{y}')$.
- How can we express an HMM in this framework?
 - transitions = features over tag n-grams
 - emissions = tag + word features
 - weights = log probabilities
 - $\arg \max_{\mathbf{y}'} =$ Viterbi decoding

Viterbi for Linear Models

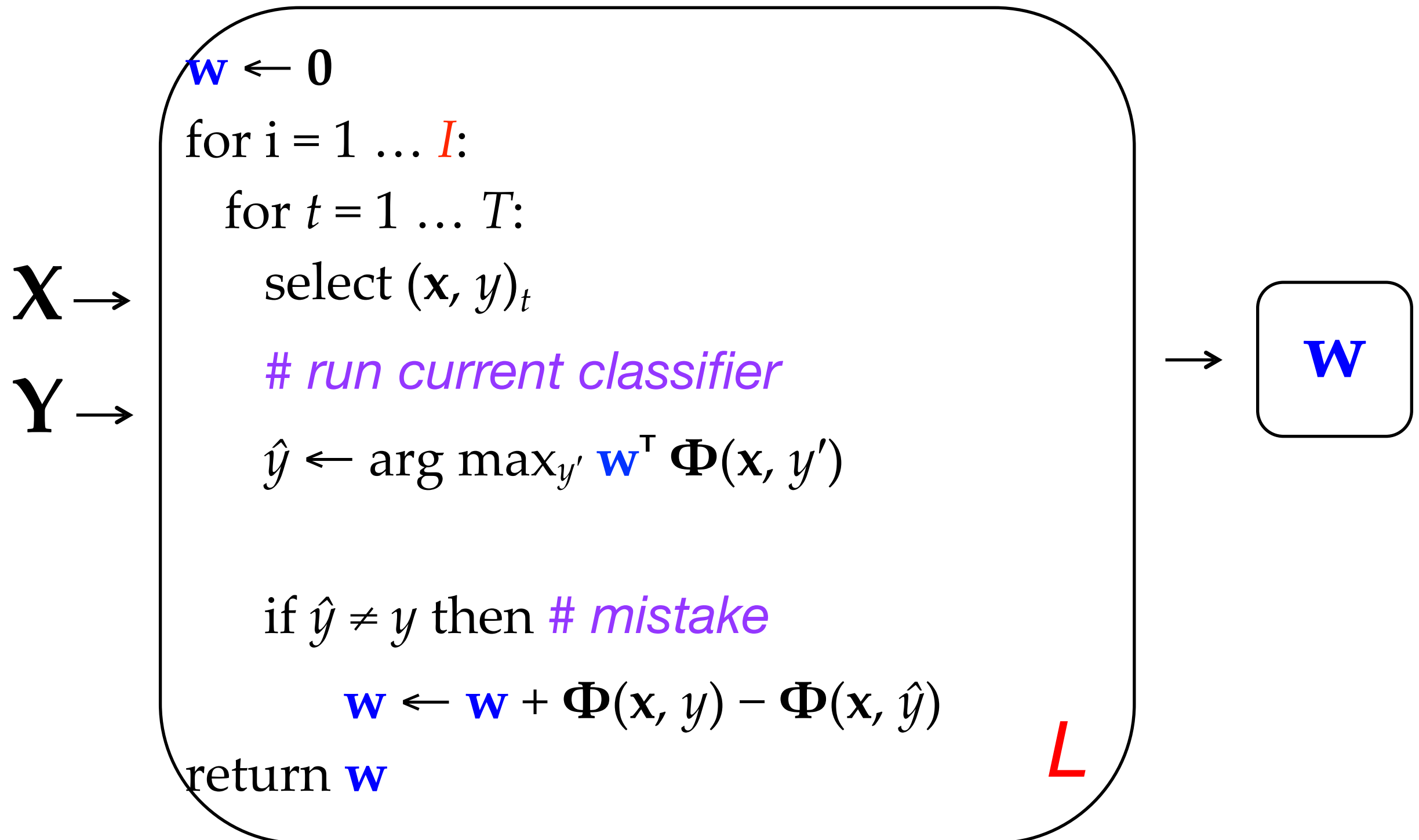
- Essentially, the Viterbi algorithm stays the same:
 - ▶ *transition probabilities* replaced by linear score of **transition (multi-tag) features**
 - ▶ *emission probabilities* replaced by linear score of **non-transition (single-tag) features**



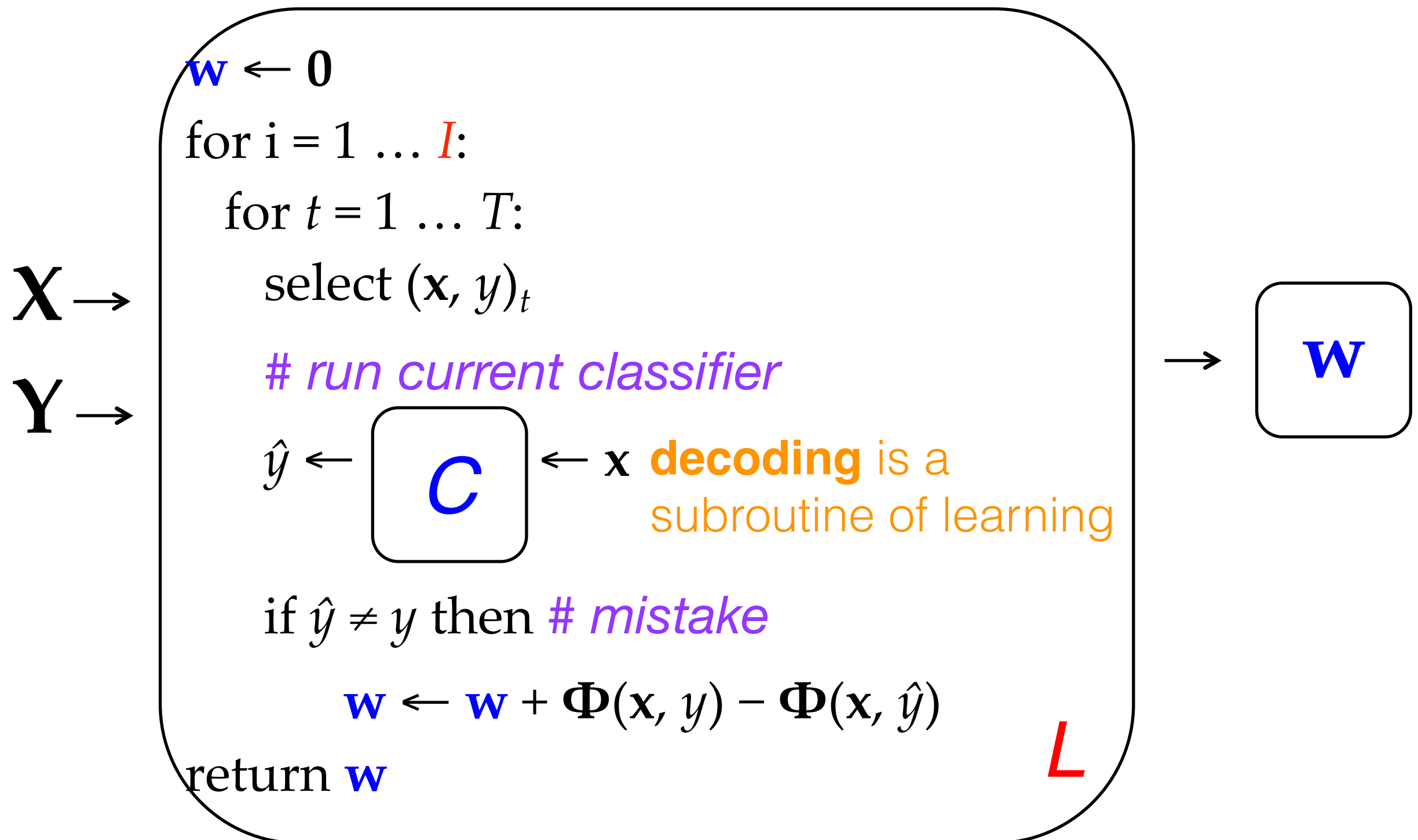
Generative → Discriminative

- If we want to estimate the weights without making independence assumptions about the features...
- ...we can use a discriminative learning algorithm!
- However, the algorithm has to take the **structure** of the output into account. Tag n-gram features mean the prediction of one tag influences what the model thinks about other tags.
- Machine learning with models where the outputs are interrelated is called **structured prediction**.

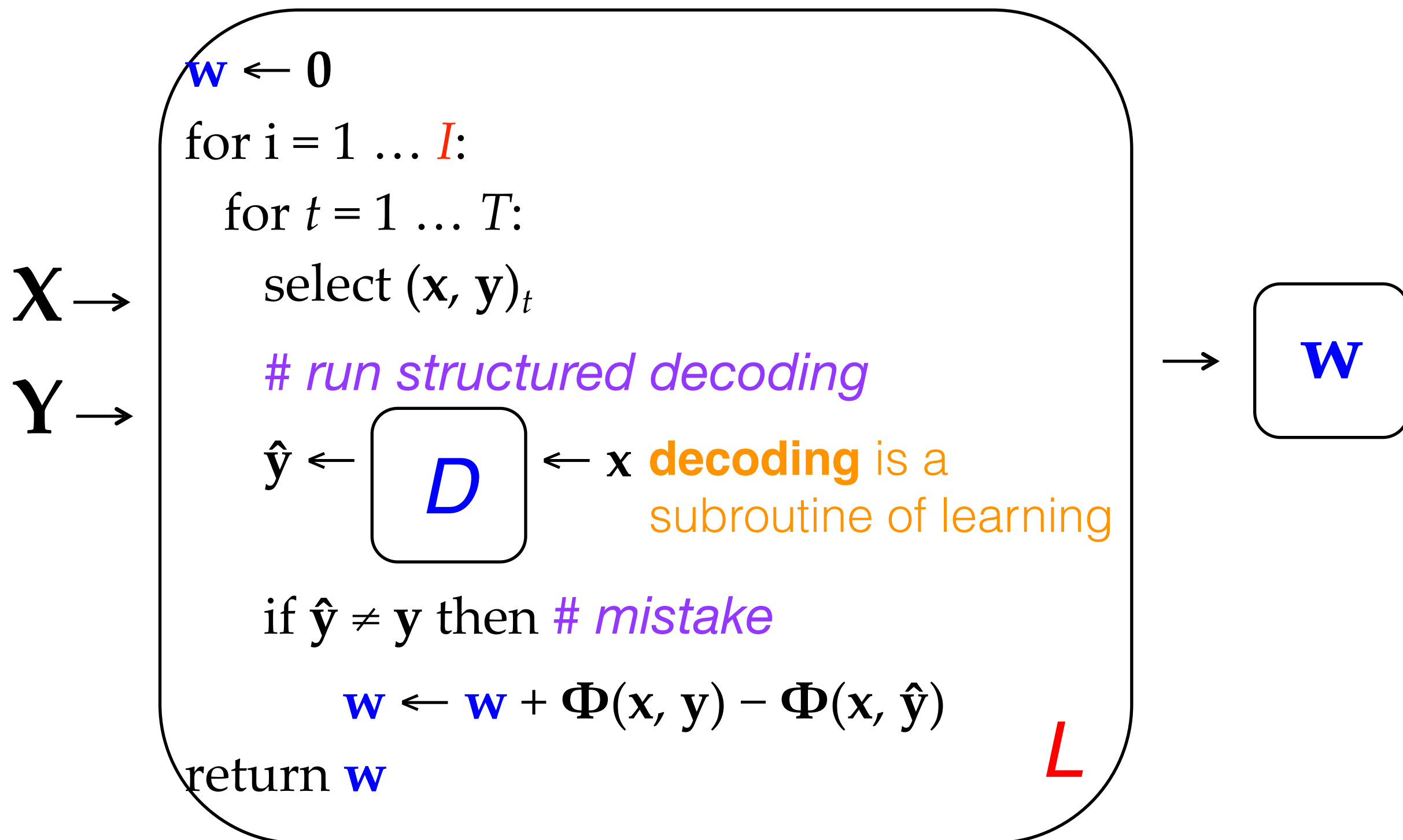
Review: Perceptron Learner



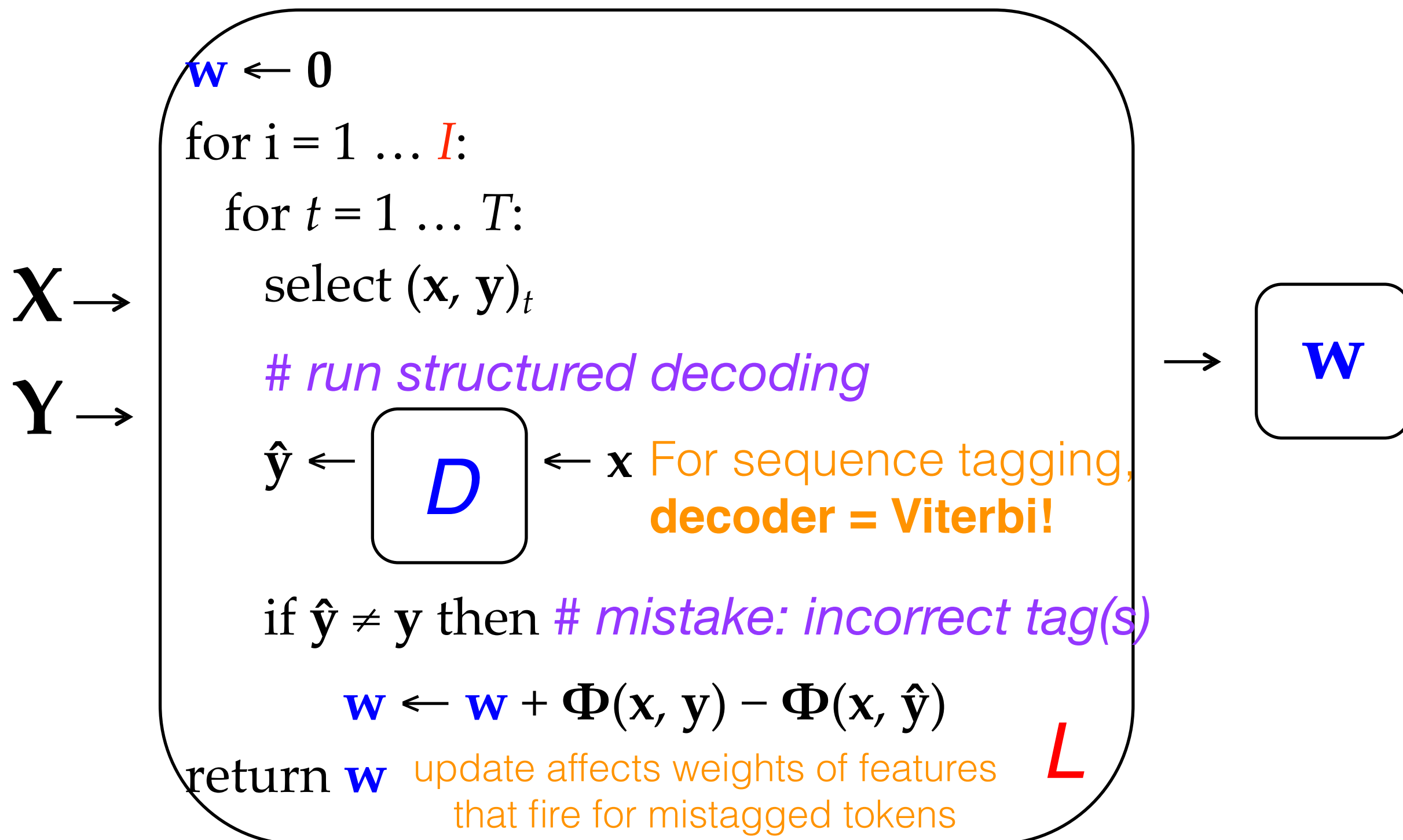
Review: Perceptron Learner



Structured Perceptron Learner



Structured Perceptron Learner



Structured Perceptron

- What are the constraints on the kinds of features we can use? (tag bigrams? trigrams? word bigrams? trigrams?)
 - Remember that discriminative = we don't care about modeling the probability of the language. Thus, every model feature should involve at least one tag.
 - As a sequence model, **Markov order** is still relevant: if we want to use the *bigram* Viterbi algorithm, which is $O(T^2N)$, we can have features over *tag bigrams*, but not trigrams.
 - **local feature** = feature which respects the independence assumptions of the decoding algorithm (e.g., tag bigram Viterbi). Using nonlocal features would require fancier algorithms.
 - Unlike the generative HMM, **no constraint on which words** can be in a feature. E.g., there could be a feature that relates the first tag to the last token! (In POS tagging, perhaps ending with “?” correlates with certain kinds of initial words.)

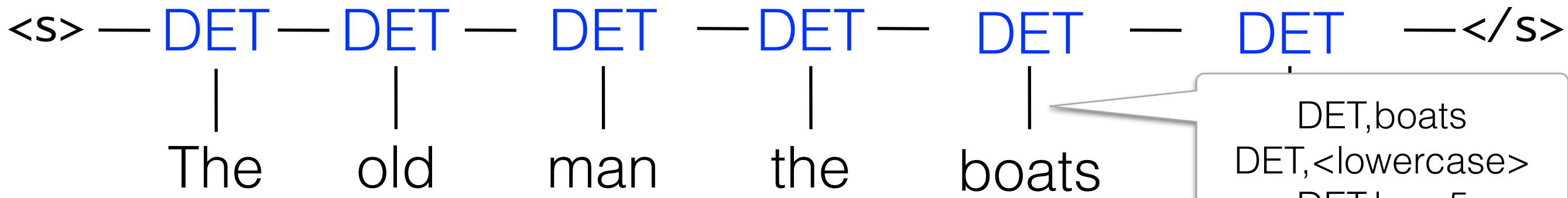
Predicted (iteration 1):

<s> — DET — DET — DET — DET — DET — DET — </s>
| | | | | | |
The old man the boats .

Gold:

<s> — DET — ADJ — VERB — DET — NOUN — PUNCT — </s>
| | | | | | |
The old man the boats .

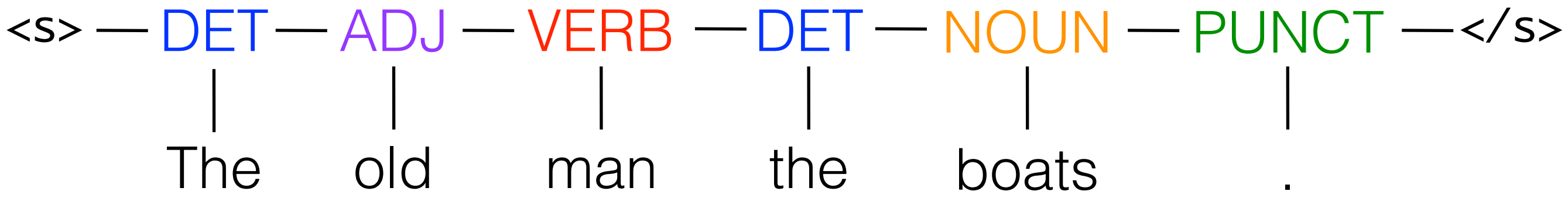
Predicted (iteration 1):



DET,DET

DET,boats
DET,<lowercase>
DET,len=5
DET,suffix=s

Gold:



Unlike the generative HMM, each connection can involve multiple weighted features.

Predicted (iteration 1):

<s> — DET — **DET** — **DET** — DET — **DET** — **DET** — </s>
| | | | | | |
The old man the boats .

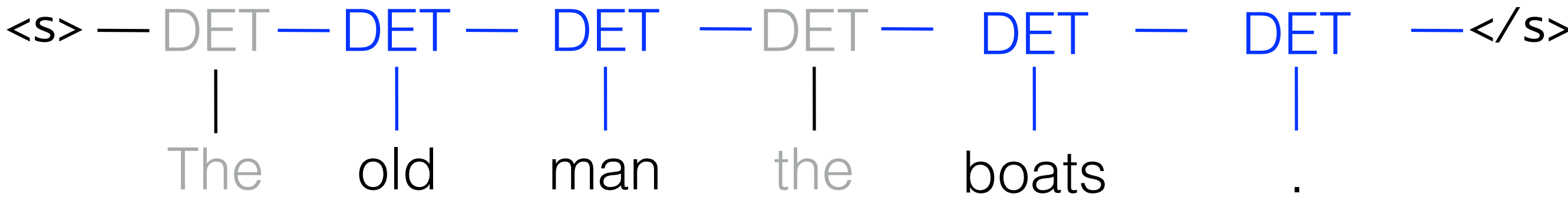
Gold:

<s> — DET — **ADJ** — **VERB** — DET — **NOUN** — **PUNCT** — </s>
| | | | | | |
The old man the boats .

Update parameters!

correct tags: no change to weights

Predicted (iteration 1):



Gold:



Update parameters!

weights for incorrectly predicted tags get **more negative**,
weights for gold tags get **more positive**

Discriminative Classifiers: Non-probabilistic

- The structured counterpart of the perceptron classifier is called...the structured perceptron.
 - Also: structural SVM (max-margin).

Discriminative Classifiers: Probabilistic

- The structured counterpart of the logistic regression classifier: **conditional random field (CRF)**.
 - Most common: **linear-chain** structure, i.e., sequence
 - Probabilistic—linear score is exponentiated & normalized
 - Training requires forward-backward algorithm (expensive!)
 - Downloadable implementations include CRF++
 - If you want the gory details: Sutton & McCallum, <http://homepages.inf.ed.ac.uk/csutton/publications/crftut-fnt.pdf>
- There is also the **Maximum Entropy Markov Model (MEMM)**, which makes simplifying assumptions to reduce computation and is nearly as accurate in practice.