CMSC 473/673
Natural Language Processing

Instructor: Lara J. Martin (she/they)

TA: Duong Ta (he)

Slides modified from Dr. Frank Ferraro & Dr. Jason Eisner

Learning Objectives

Define featurization & other ML terminology

Define some “classification” terminology

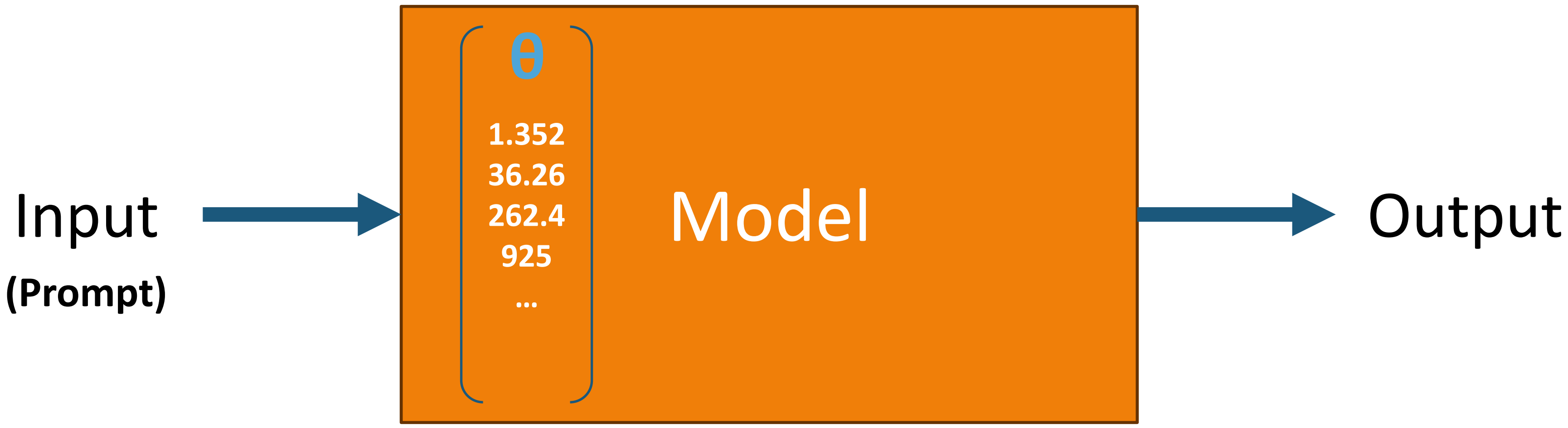
Formalize NLP Tasks at a high-level

- Document classification
- Part of speech tagging
- Syntactic parsing
- Entity id/coreference

Helpful ML Terminology

Model: the (computable) way to go from **features** (input) to labels/scores (output)

Weights/parameters (θ): vectors of numbers that control how the model produces labels/scores from inputs. These are learned through **training**.

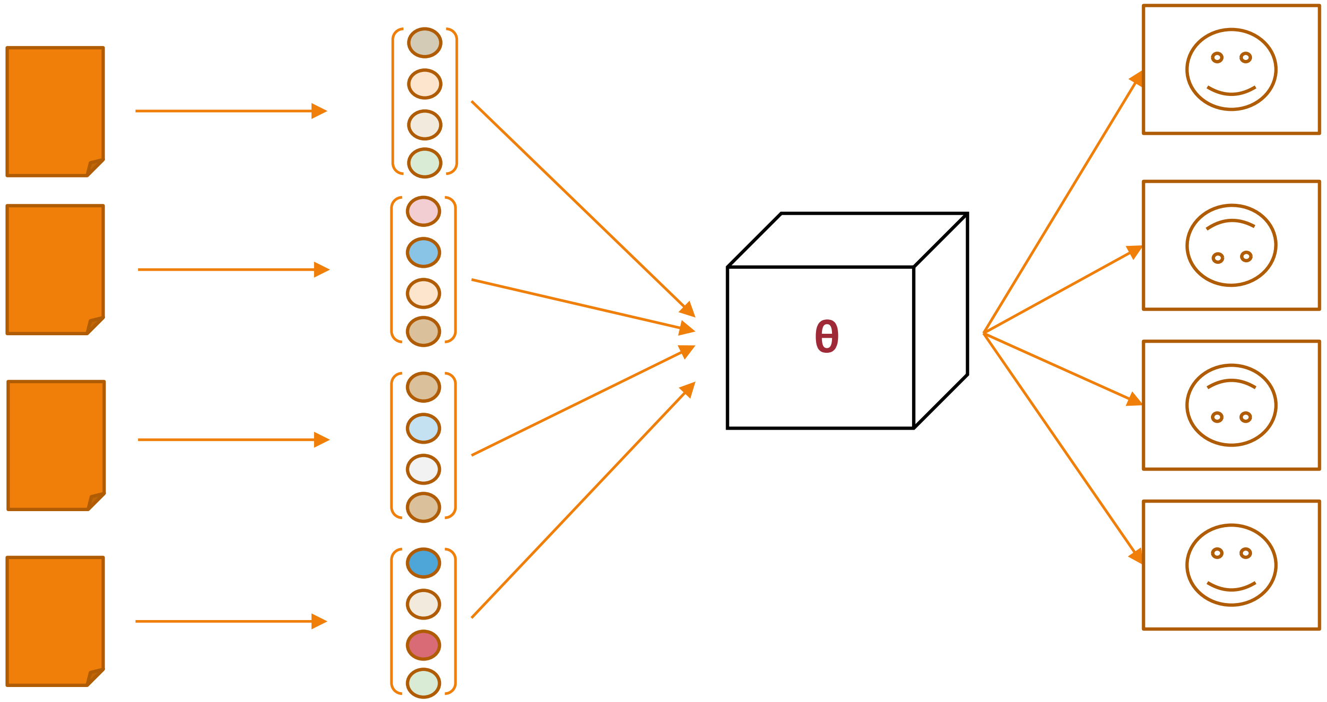


ML/NLP Framework

instances

features:
K-dimensional vector representations (one per instance)

ML model:
- take in featurized input
- output scores/labels
- contains weights θ



Helpful ML Terminology

Model: the (computable) way to go from **features** (input) to labels/scores (output)

Weights/parameters: vectors of numbers that control how the model produces labels/scores from inputs. These are learned through **training**.

Objective function: an algorithm/calculation, whose variables are the **weights** of the **model**, that we numerically optimize in order to learn appropriate weights based on the labels/scores. The **model's** weights are adjusted.

Evaluation function: an algorithm/calculation that scores how "correct" the **model's** predictions are. The **model's** weights are not adjusted.

Note: The evaluation and objective functions are often different!

(More) Helpful ML Terminology

Learning:

- the process of adjusting the model's weights to learn to make good predictions.

Inference / Prediction / Decoding / Classification:

- the process of using a model's existing weights to make (hopefully!) good predictions

ML/NLP Framework for Learning

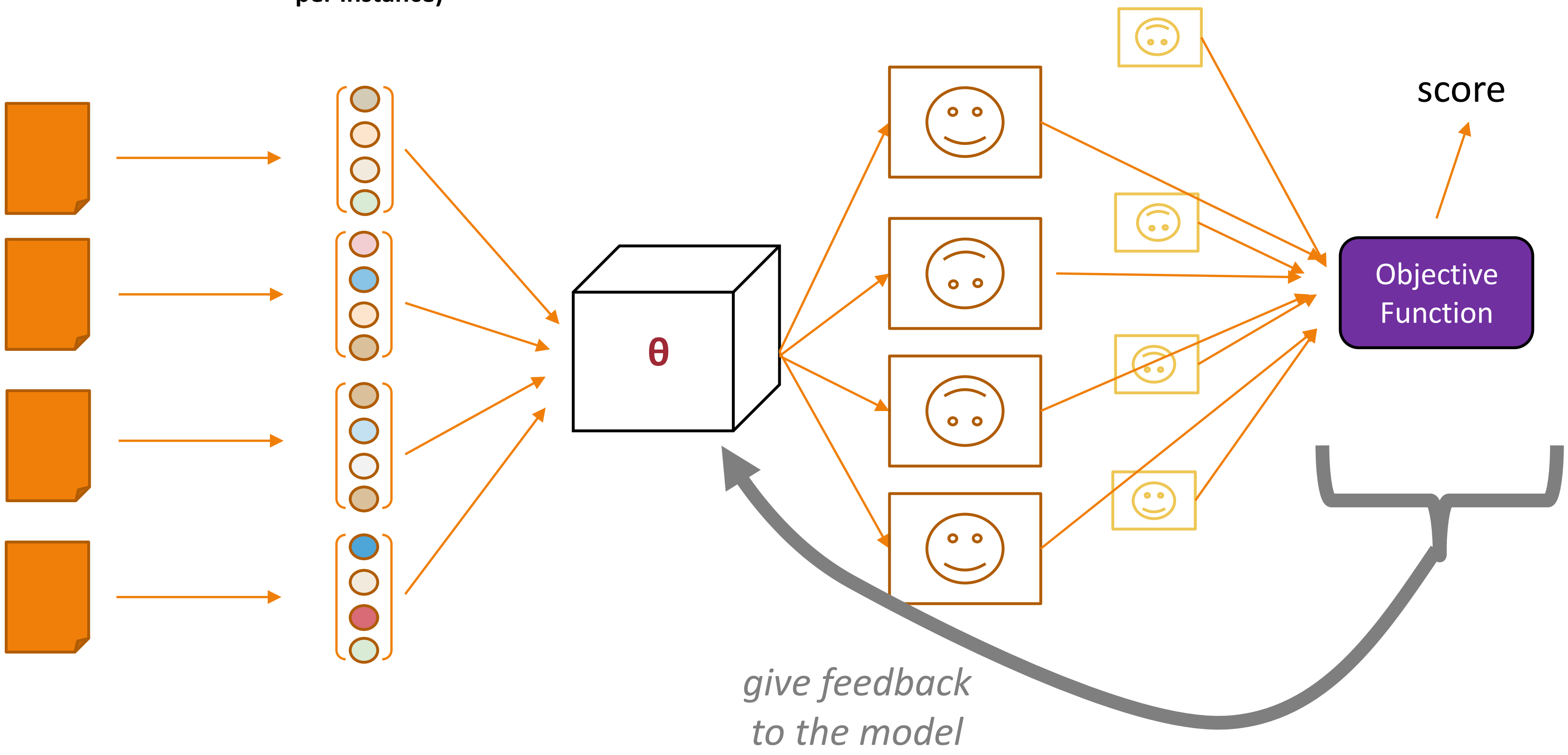
instances

features:
K-dimensional vector representations (one per instance)

ML model:
- take in featurized input
- output scores/labels
- contains weights θ

"Gold" (correct) labels

Objective Function/ Learning



ML/NLP Framework for Prediction

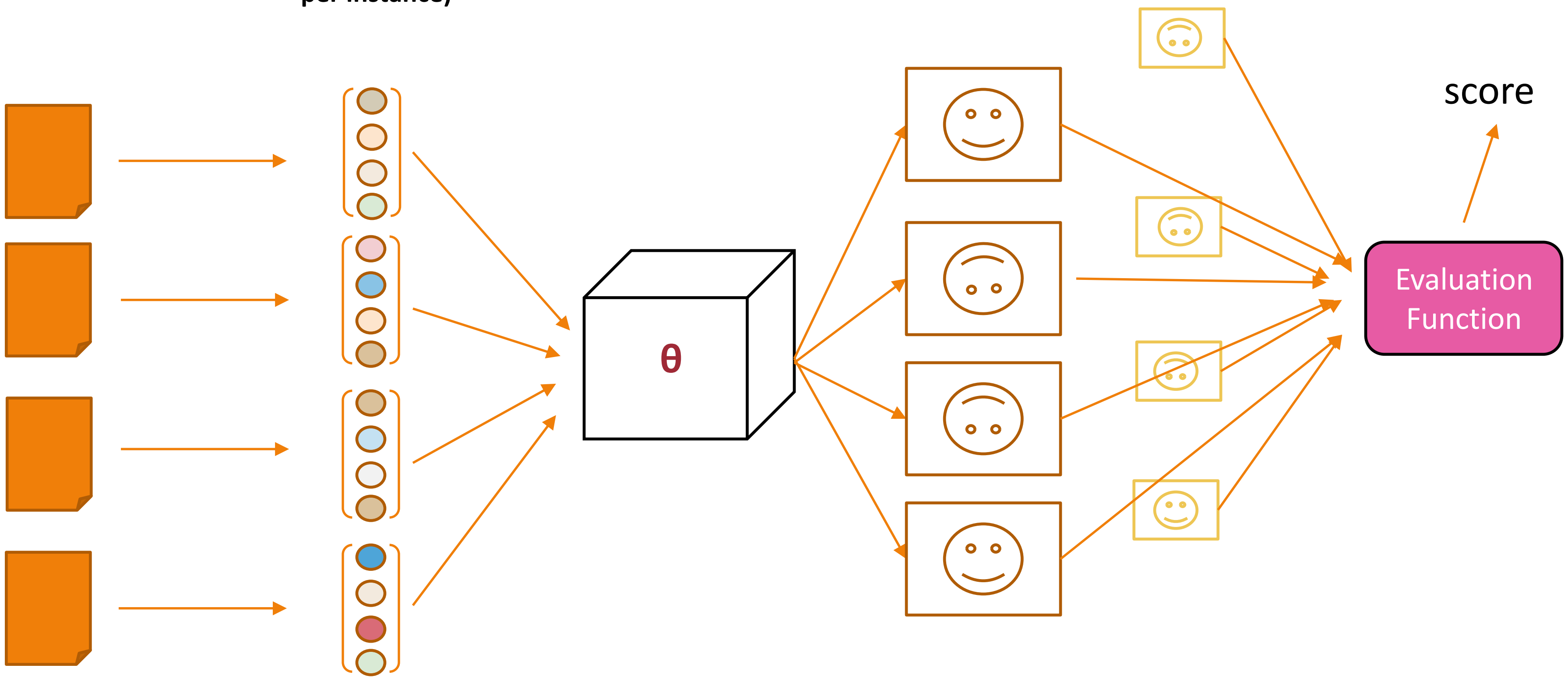
instances

features:
K-dimensional vector representations (one per instance)

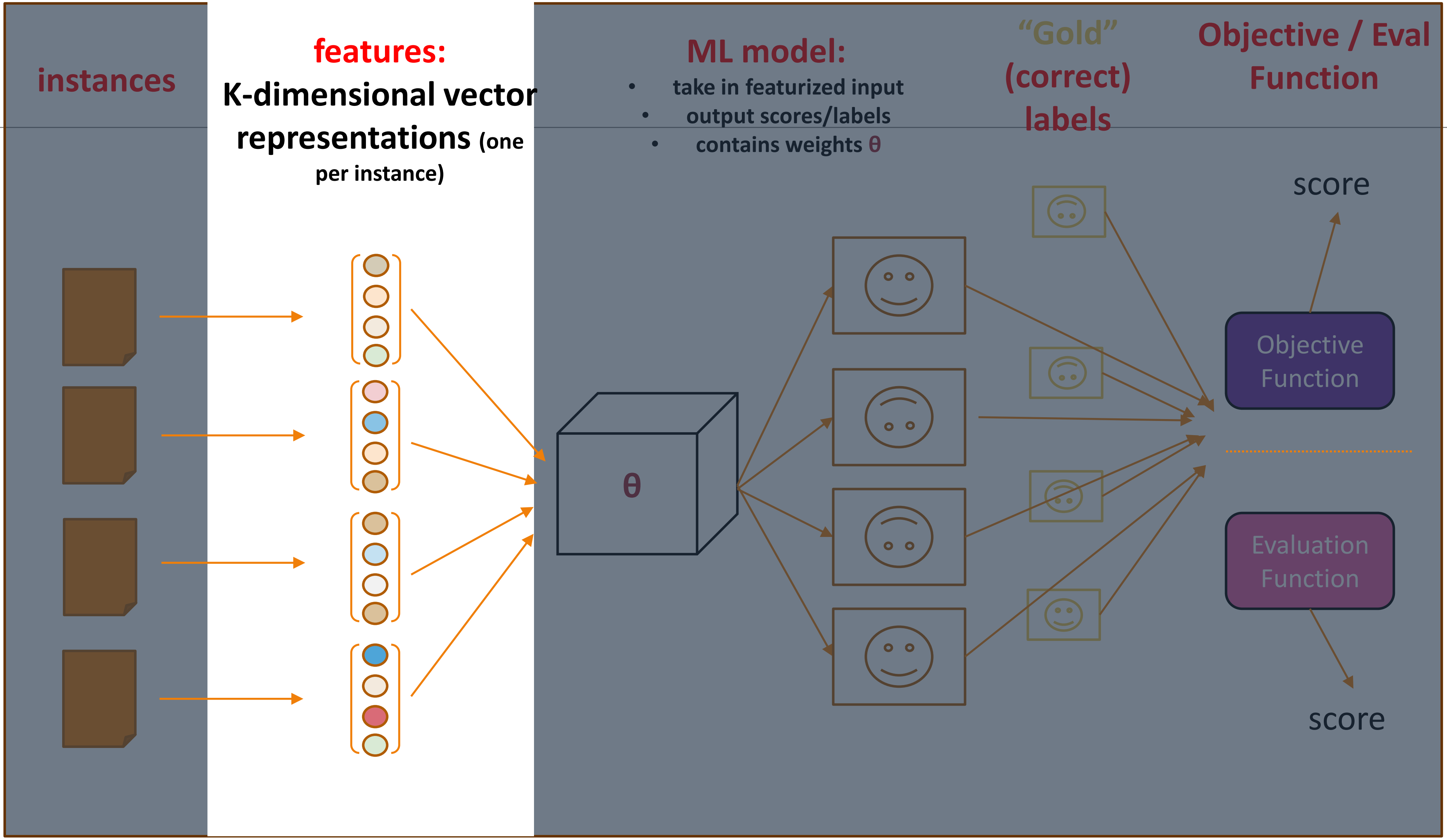
ML model:
- take in featurized input
- output scores/labels
- contains weights θ

"Gold" (correct) labels

Evaluation Function



First: Featurization / Encoding / Representation



ML Term: "Featurization"

The procedure of extracting **features** for some input

Often viewed as a K-dimensional vector function *f* of the input language *x*

$$f(x) = (\underbrace{f_1(x)}, \dots, \underbrace{f_K(x)})$$

Each of these is a feature
(/feature function)

ML Term: “Featurization”

The procedure of extracting **features** for some input

Often viewed as a K-dimensional vector function f of the input language x

$$f(x) = (f_1(x), \dots, f_K(x))$$

In supervised settings, it can equivalently be viewed as a K-dimensional vector function f of the input language x and a potential label y

- $f(x, y) = (f_1(x, y), \dots, f_K(x, y))$

Features can be thought of as “soft” rules

- E.g., positive sentiments tweets may be *more likely* to have the word “happy”

Defining Appropriate Features

Feature functions help extract useful features (characteristics) of the data

They turn data into numbers

Features that are not 0 are said to have fired

Defining Appropriate Features

Feature functions help extract useful features (characteristics) of the data

They turn data into numbers

Features that are not 0 are said to have fired

You can define classes of features by templating (we'll come back to this!)

Often binary-valued (0 or 1), but can be real-valued

Three Common Types of Featurization in NLP

1. Bag-of-words (or bag-of-characters, bag-of-relations)

2. Linguistically-inspired features

3. Dense features via embeddings

Three Common Types of Featurization in NLP

1. Bag-of-words (or bag-of-characters, bag-of-relations)
    - easy to define / extract
    - sometimes still very useful

2. Linguistically-inspired features

3. Dense features via embeddings

Three Common Types of Featurization in NLP

1. Bag-of-words (or bag-of-characters, bag-of-relations) ←
   - easy to define / extract
   - sometimes still very useful
2. Linguistically-inspired features ←
   - harder to define
   - helpful for interpretation
   - depending on task: conceptually helpful
   - currently, not freq. used
3. Dense features via embeddings

Three Common Types of Featurization in NLP

1. Bag-of-words (or bag-of-characters, bag-of-relations)
   - easy to define / extract
   - sometimes still very useful
2. Linguistically-inspired features
   - harder to define
   - helpful for interpretation
   - depending on task: conceptually helpful
   - currently, not freq. used
3. Dense features via embeddings
   - harder to define
   - harder to extract (unless there's a model to run)
   - currently: freq. used

Three Common Types of Featurization in NLP

1. Bag-of-words (or bag-of-characters, bag-of-relations)
   - Identify ***unique*** sufficient atomic sub-parts (e.g., words in a document)
   - Define simple features over these, e.g.,
     - Binary (0 or 1) ➔ indicating presence
     - Natural numbers ➔ indicating number of times in a context
     - Real-valued ➔ various other score (we'll see examples throughout the semester)
2. Linguistically-inspired features
3. Dense features via embeddings

Example: Document Classification via Bag-of-Words Features

Electronic alerts have been used to assist the authorities in moments of chaos and potential danger: after the Boston bombing in 2013, when the Boston suspects were still at large, and last month in Los Angeles, during an active shooter scare at the airport.

TECH

NOT TECH

Let’s make a core assumption: the **label** can be predicted from **counts of individual word types**

Example: Document Classification via Bag-of-Words Features

Electronic alerts have been used to assist the authorities in moments of chaos and potential danger: after the Boston bombing in 2013, when the Boston suspects were still at large, and last month in Los Angeles, during an active shooter scare at the airport.

TECH

NOT TECH

feature extraction

With V word types, define V feature functions $f_i(x)$ as

$f_i(x)$ =# of times word type *i* appears in document x

Core assumption: the label can be predicted from counts of individual word types

Example: Document Classification via Bag-of-Words Features

Electronic alerts have been used to assist the authorities in moments of chaos and potential danger: after the Boston bombing in 2013, when the Boston suspects were still at large, and last month in Los Angeles, during an active shooter scare at the airport.

TECH

NOT TECH

feature extraction

With V word types, define V feature functions $f_i(x)$ as

$f_i(x)$ =# of times word type *i* appears in document x

$$f(x) = \left(f_i(x)\right)_i^V$$

Core assumption: the label can be predicted from counts of individual word types

Example: Document Classification via Bag-of-Words Features

Electronic alerts have been used to assist the authorities in moments of chaos and potential danger: after the Boston bombing in 2013, when the Boston suspects were still at large, and last month in Los Angeles, during an active shooter scare at the airport.

TECH

NOT TECH

feature extraction

feature $f_i(x)$	value
alerts	1
assist	1
bombing	1
Boston	2
…	
sniffle	0
…	

Core assumption: the label can be predicted from counts of individual word types

Example: Document Classification via Bag-of-Words Features

Electronic alerts have been used to assist the authorities in moments of chaos and potential danger: after the Boston bombing in 2013, when the Boston suspects were still at large, and last month in Los Angeles, during an active shooter scare at the airport.

TECH

NOT TECH

f(**x**): "bag of words"

feature $f_i(x)$	value
alerts	1
assist	1
bombing	1
Boston	2
…	
sniffle	0
…	

w: weights

feature	weight
alerts	.043
assist	-0.25
bombing	0.8
Boston	-0.00001
…	

Three Common Types of Featurization in NLP

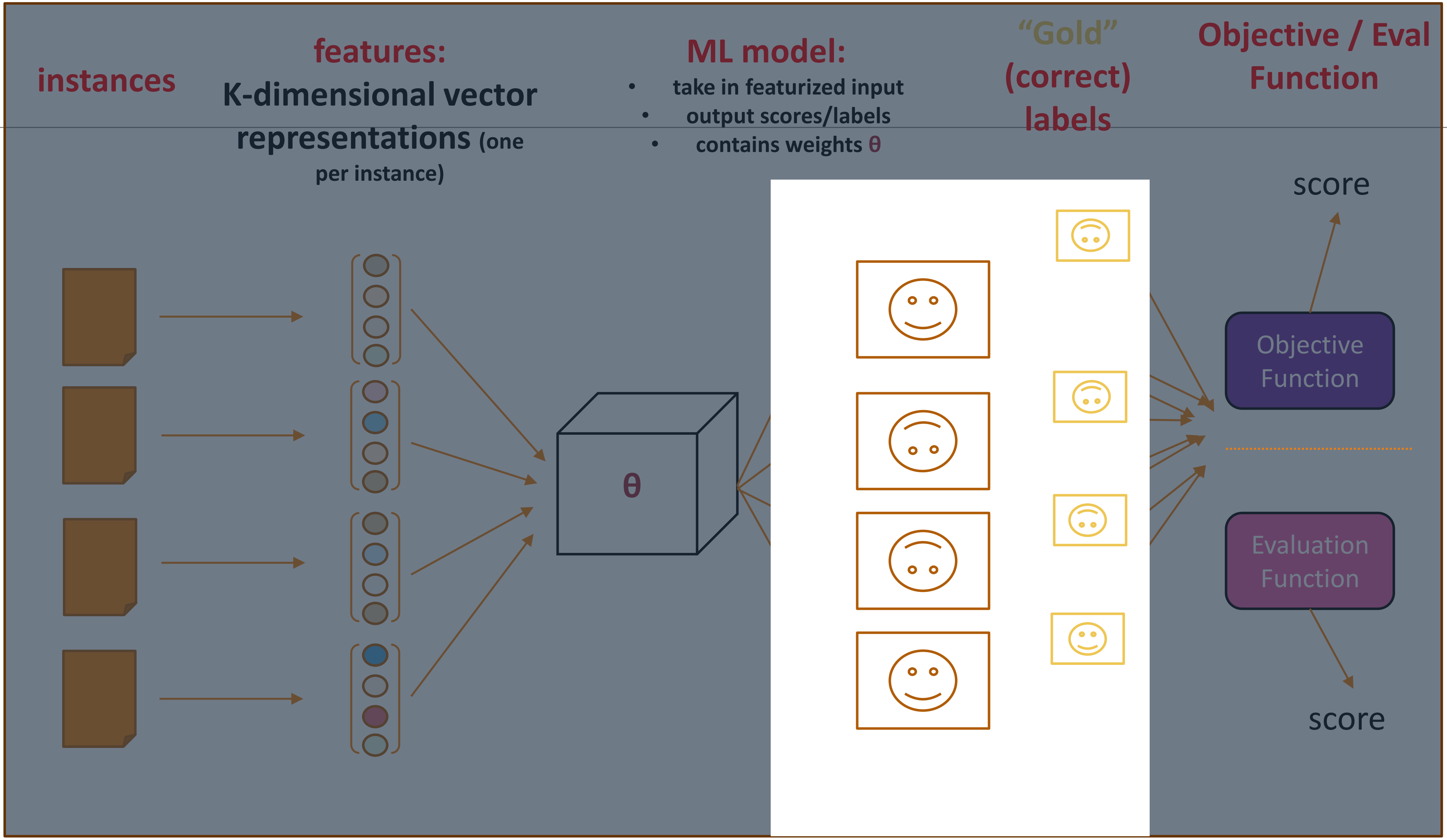
1. Bag-of-words (or bag-of-characters, bag-of-relations)
   - Identify ***unique*** sufficient atomic sub-parts (e.g., words in a document)
   - Define simple features over these, e.g.,
     - Binary (0 or 1) ➔ indicating presence
     - Natural numbers ➔ indicating number of times in a context
     - Real-valued ➔ various other score (we'll see examples throughout the semester)
2. Linguistically-inspired features
   - Define features from words, word spans, or linguistic-based annotations extracted from the document
3. Dense features via embeddings

Three Common Types of Featurization in NLP

1. Bag-of-words (or bag-of-characters, bag-of-relations)
   - Identify ***unique*** sufficient atomic sub-parts (e.g., words in a document)
   - Define simple features over these, e.g.,
     - Binary (0 or 1) ➔ indicating presence
     - Natural numbers ➔ indicating number of times in a context
     - Real-valued ➔ various other score (we'll see examples throughout the semester)
2. Linguistically-inspired features
   - Define features from words, word spans, or linguistic-based annotations extracted from the document
3. Dense features via embeddings
   - Compute/extract a real-valued vector, e.g., from word2vec, ELMO, BERT, …

Will be discussed in a future lecture

Second: Classification Terminology



Classification Types (Terminology)

Name	Number of Tasks (Domains) Labels are Associated with	# Label Types	Example
(Binary) Classification			
Multi-class Classification			
Multi-label Classification			
Multi-task Classification			

Classification Types (Terminology)

Name	Number of Tasks (Domains) Labels are Associated with	# Label Types	Example
(Binary) Classification	1	2	Sentiment: Choose one of {positive or negative}
Multi-class Classification			
Multi-label Classification			
Multi-task Classification			

Classification Types (Terminology)

Name	Number of Tasks (Domains) Labels are Associated with	# Label Types	Example
(Binary) Classification	1	2	Sentiment: Choose one of {positive or negative}
Multi-class Classification	1	> 2	Part-of-speech: Choose one of {Noun, Verb, Det, Prep, …}
Multi-label Classification			
Multi-task Classification			

Classification Types (Terminology)

Name	Number of Tasks (Domains) Labels are Associated with	# Label Types	Example
(Binary) Classification	1	2	Sentiment: Choose one of {positive or negative}
Multi-class Classification	1	> 2	Part-of-speech: Choose one of {Noun, Verb, Det, Prep, …}
Multi-label Classification	1	> 2	Sentiment: Choose multiple of {positive, angry, sad, excited, …}
Multi-task Classification			

Classification Types (Terminology)

Name	Number of Tasks (Domains) Labels are Associated with	# Label Types	Example
(Binary) Classification	1	2	Sentiment: Choose one of {positive or negative}
Multi-class Classification	1	> 2	Part-of-speech: Choose one of {Noun, Verb, Det, Prep, …}
Multi-label Classification	1	> 2	Sentiment: Choose multiple of {positive, angry, sad, excited, …}
Multi-task Classification	> 1	Per task: 2 or > 2 (can apply to binary or multi-class)	Task 1: part-of-speech Task 2: named entity tagging … ---------------------- Task 1: document labeling Task 2: sentiment

Text Annotation Tasks

1. Classify the entire document ("text categorization")
2. Classify word tokens individually
3. Classify word tokens in a sequence
4. Identify phrases ("chunking")
5. Syntactic annotation (parsing)
6. Semantic annotation
7. Text generation

Slide courtesy Jason Eisner, with mild edits

Text Annotation Tasks

1. Classify the entire document (“text categorization”)
2. Classify word tokens individually
3. Classify word tokens in a sequence
4. Identify phrases (“chunking”)
5. Syntactic annotation (parsing)
6. Semantic annotation

Slide courtesy Jason Eisner, with mild edits

Text Classification

Assigning subject categories, topics, or genres

Spam detection

Authorship identification

Age/gender identification

Language Identification

Sentiment analysis

…

Text Classification

Assigning subject categories, topics, or genres

Spam detection

Authorship identification

Age/gender identification

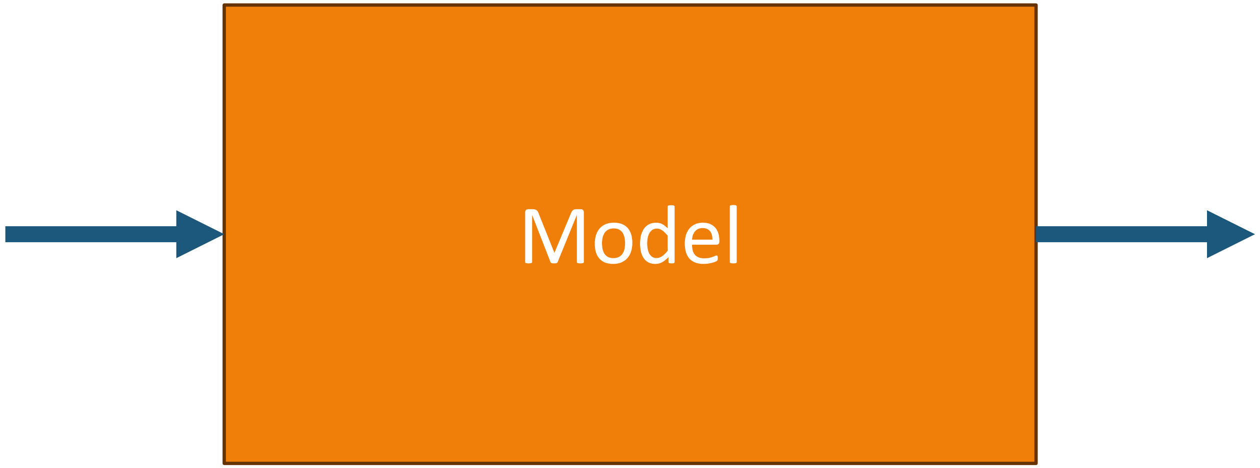
Language Identification

Sentiment analysis

…

a document

a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$



a predicted class c from C

Text Classification: Hand-coded Rules?

Assigning subject categories, topics, or genres

Spam detection

Authorship identification

Age/gender identification

Language Identification

Sentiment analysis

…

Rules based on combinations of words or other features

spam: black-list-address OR ("dollars" AND "have been selected")

Accuracy can be high

If rules carefully refined by expert

Building and maintaining these rules is expensive

Can humans faithfully assign uncertainty?

Text Classification: Supervised Machine Learning

Assigning subject categories, topics, or genres

Spam detection

Authorship identification

Age/gender identification

Language Identification

Sentiment analysis

…

a document d

a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

a training set of m hand-labeled documents $(d_1, y_1), \ldots, (d_m, y_m)$, $y \in C$

Model

a learned classifier γ that maps documents to classes

Text Classification: Supervised Machine Learning

Assigning subject categories, topics, or genres

Spam detection

Authorship identification

Age/gender identification

Language Identification

Sentiment analysis

…

a document d

a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

a training set of m hand-labeled documents $(d_1,y_1),\ldots.,(d_m,y_m)$, y \in C

Naïve Bayes
Logistic regression
Neural network
Support-vector machines
k-Nearest Neighbors
…

a learned classifier γ that maps documents to classes

Text Annotation Tasks

1. Classify the entire document (“text categorization”)
2. Classify word tokens individually
3. Classify word tokens in a sequence
4. Identify phrases (“chunking”)
5. Syntactic annotation (parsing)
6. Semantic annotation
7. Text generation

Slide courtesy Jason Eisner, with mild edits

p(class | token in context)

Word Sense Disambiguation (WSD)

Problem:

The company said the *plant* is still operating ...

⇒ (A) Manufacturing plant or

⇒ (B) Living plant

Training Data: Build a special classifier just for tokens of “plant”

Sense	Context		
(1) Manufacturing	... union responses to	*plant*	closures . ...
” ”	... computer disk drive	*plant*	located in ...
” ”	company manufacturing	*plant*	is in Orlando ...
(2) Living	... animal rather than	*plant*	tissues can be ...
” ”	... to strain microscopic	*plant*	life from the ...
” ”	and Golgi apparatus of	*plant*	and animal cells

Test Data:

Sense	Context		
???	... vinyl chloride monomer	*plant*	, which is ...
???	... molecules found in	*plant*	tissue from the ...

slide courtesy of D. Yarowsky (modified)

p(class | token in context)

WSD for Machine Translation
(English → Spanish)

Problem:

... He wrote the last **sentence** two years later ...

⇒ *sentencia* (legal sentence) or

⇒ *frase* (grammatical sentence)

Training Data: Build a special classifier just for tokens of "sentence"

Translation	Context
(1) sentencia	... for a maximum *sentence* for a young offender ...
" "	... of the minimum *sentence* of seven years in jail ...
" "	... were under the *sentence* of death at that time ...
(2) frase	... read the second *sentence* because it is just as ...
" "	... The next *sentence* is a very important ...
" "	... It is the second *sentence* which I think is at ...

Test Data:

Translation	Context
???	... cannot criticize a *sentence* handed down by ...
???	... listen to this *sentence* uttered by a former ...

slide courtesy of D. Yarowsky (modified)

p(class | token in context)

Accent Restoration in Spanish & French

Problem:

Input:	... deja travaille cote a cote ...
	$\Downarrow$
Output:	... déjà travaillé côte à côte ...

Examples:

... appeler l'autre **cote** de l'atlantique ...

$\Rightarrow$ *côté* (meaning side) or

$\Rightarrow$ *côte* (meaning coast)

. . . une famille des **pecheurs** . . .

$\Rightarrow$ *pêcheurs* (meaning fishermen) or

$\Rightarrow$ *pécheurs* (meaning sinners)

slide courtesy of D. Yarowsky (modified)

p(class | token in context)

Accent Restoration in Spanish & French

Training Data:

Pattern	Context		
(1) côté	... du laisser de	*cote*	faute de temps ...
" "	... appeler l' autre	*cote*	de l' atlantique ...
" "	... passe de notre	*cote*	de la frontiere ...
(2) côte	... vivre sur notre	*cote*	ouest toujours ...
" "	... creer sur la	*cote*	du labrador des ...
" "	travaillaient cote a	*cote*	, ils avaient ...

Test Data:

Pattern	Context		
???	... passe de notre	*cote*	de la frontiere ...
???	... creer sur la	*cote*	du labrador des ...

slide courtesy of D. Yarowsky (modified)

p(class | token in context)

Text-to-Speech Synthesis

Problem:

... slightly elevated *lead* levels ...
⇒ *lɛd* (as in *lead mine*) or
⇒ *li:d* (as in *lead role*)

Training Data:

Pronunciation	Context
(1) lɛd	... it monitors the *lead* levels in drinking ...
" "	... conference on *lead* poisoning in ...
" "	... strontium and *lead* isotope zonation ...
(2) li:d	... maintained their *lead* Thursday over ...
" "	... to Boston and *lead* singer for Purple ...
" "	... Bush a 17-point *lead* in Texas , only 3 ...

Test Data:

Pronunciation	Context
???	... median blood *lead* concentration was ..
???	... his double-digit *lead* nationwide . The ...

slide courtesy of D. Yarowsky (modified)

p(class | token in context)

Spelling Correction

Problem:

... and he fired presidential **aid/aide** Dick Morris after ...

$\Rightarrow$ *aid* or

$\Rightarrow$ *aide*

Training Data:

Spelling	Context
(1) aid	... and cut the foreign *aid/aide* budget in fiscal 1996 ...
" "	... they offered federal *aid/aide* for flood-ravaged states ...
(2) aide	... fired presidential *aid/aide* Dick Morris after ...
" "	... and said the chief *aid/aide* to Sen. Baker, Mr. John ...

Test Data:

Spelling	Context
???	... said the longtime *aid/aide* to the Mayor of St. ...
???	... will squander the *aid/aide* it receives from the ...

slide courtesy of D. Yarowsky (modified)

What features? Example: "word to [the] left [of correction]"

Word to left	Frequency as **Aid**	Frequency as **Aide**
foreign	718	1
federal	297	0
western	146	0
provide	88	0
covert	26	0
oppose	13	0
future	9	0
similar	6	0
presidential	0	63
chief	0	40
longtime	0	26
aids-infected	0	2
sleepy	0	1
disaffected	0	1
indispensable	2	1
practical	2	0
squander	1	0

Spelling correction using an n-gram language model ($n \geq 2$) would use words to left and right to help predict the true word.

Similarly, an HMM would predict a word's class using classes to left and right.

But we'd like to throw in all kinds of other features, too …

slide courtesy of D. Yarowsky (modified)

An assortment of possible cues ...

	Position	Collocation	lɛd	li:d
N-grams	+1 L	lead *level/N*	219	0
	-1 W	*narrow* lead	0	70
(word,	+1 W	lead *in*	207	898
lemma,	-1W,+1W	*of* lead *in*	162	0
part-of-speech)	-1W,+1W	*the* lead *in*	0	301
	+1P,+2P	lead *, <NOUN>*	234	7
Wide-context	±k W	*zinc* (in ±*k* words)	235	0
collocations	±k W	*copper* (in ±*k* words)	130	0
Verb-object	-V L	*follow/V* + lead	0	527
relationships	-V L	*take/V* + lead	1	665

generates a whole bunch of potential cues – use data to find out which ones work best

Word to left	Frequency as **Aid**	Frequency as **Aide**
foreign	718	1
federal	297	0
western	146	0
provide	88	0

slide courtesy of D. Yarowsky (modified)

An assortment of possible cues ...

	Position	Collocation	lɛd	li:d
N-grams	+1 L	lead *level/N*	219	0
	-1 W	*narrow* lead	0	70
(word,	+1 W	lead *in*	207	898
lemma,	-1W,+1W	*of* lead *in*	162	0
part-of-speech)	-1W,+1W	*the* lead *in*	0	301
	+1P,+2P	lead , *<NOUN>*	234	7
Wide-context	±k W	*zinc* (in ±*k* words)	235	0
collocations	±k W	*copper* (in ±*k* words)	130	0
Verb-object	-V L	*follow/V* + lead	0	527
relationships	-V L	*take/V* + lead	1	665

This feature is relatively weak, but weak features are still useful, especially since very few features will fire in a given context.

merged ranking of all cues of all these types

11.40	*follow/V* + lead	⇒ li:d
11.20	*zinc* (in ±*k* words)	⇒ lɛd
11.10	lead *level/N*	⇒ lɛd
10.66	*of* lead *in*	⇒ lɛd
10.59	*the* lead *in*	⇒ li:d
10.31	lead *role*	⇒ li:d

slide courtesy of D. Yarowsky (modified)

Final decision list for *lead* (abbreviated)

List of all features,
ranked by their weight.

(These weights are for a simple "decision list" model where the single highest-weighted feature that fires gets to make the decision all by itself.

However, a log-linear model, which adds up the weights of all features that fire, would be roughly similar.)

LogL	Evidence	Pronunciation
11.40	*follow/V* + lead	$\Rightarrow$ li:d
11.20	*zinc* (in $\pm k$ words)	$\Rightarrow$ lɛd
11.10	lead *level/N*	$\Rightarrow$ lɛd
10.66	*of* lead *in*	$\Rightarrow$ lɛd
10.59	*the* lead *in*	$\Rightarrow$ li:d
10.51	lead *role*	$\Rightarrow$ li:d
10.35	*copper* (in $\pm k$ words)	$\Rightarrow$ lɛd
10.28	lead *time*	$\Rightarrow$ li:d
10.24	lead *levels*	$\Rightarrow$ lɛd
10.16	lead *poisoning*	$\Rightarrow$ lɛd
8.55	*big* lead	$\Rightarrow$ li:d
8.49	*narrow* lead	$\Rightarrow$ li:d
7.76	*take/V* + lead	$\Rightarrow$ li:d
5.99	lead , *NOUN*	$\Rightarrow$ lɛd
$\boxed{1.15}$	lead *in*	$\Rightarrow$ li:d
	○○○	

slide courtesy of D. Yarowsky (modified)

Text Annotation Tasks

1. Classify the entire document ("text categorization")
2. Classify word tokens individually
3. Classify word tokens in a sequence (i.e., order matters)
4. Identify phrases ("chunking")
5. Syntactic annotation (parsing)
6. Semantic annotation
7. Text generation

Slide courtesy Jason Eisner, with mild edits

Part of Speech Tagging

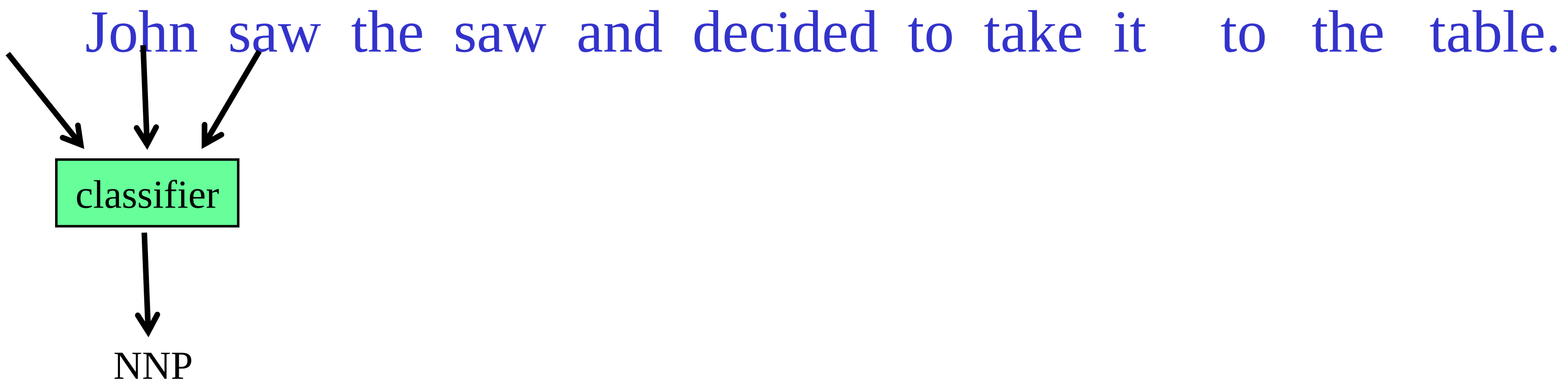
We could treat tagging as a token classification problem

- Tag each word independently given features of context
- And features of the word's spelling (suffixes, capitalization)

Slide courtesy Jason Eisner, with mild edits

Sequence Labeling as Classification

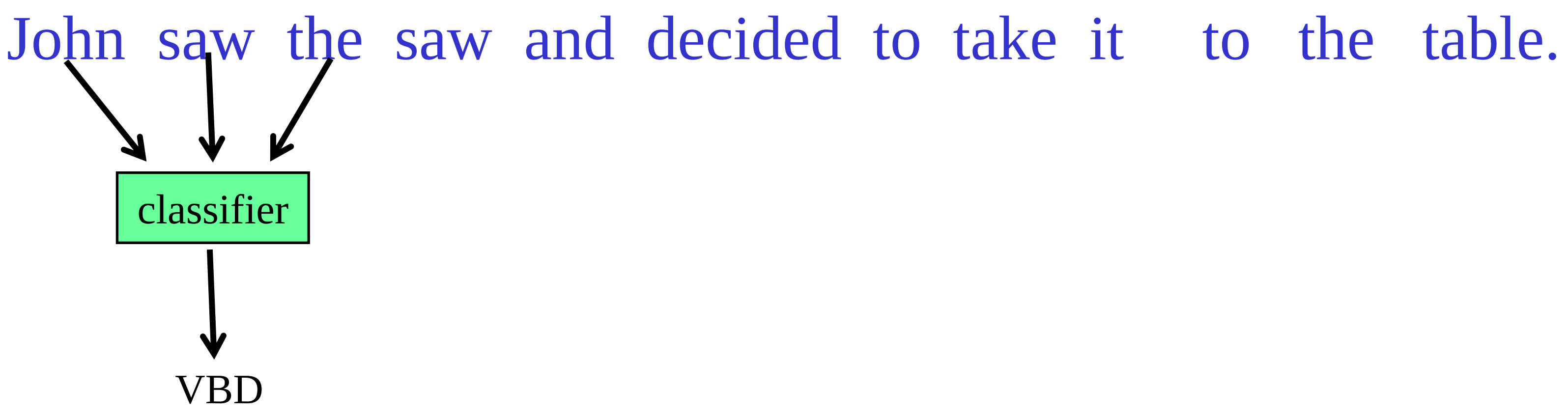
Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Slide courtesy Ray Mooney, with mild edits

Sequence Labeling as Classification

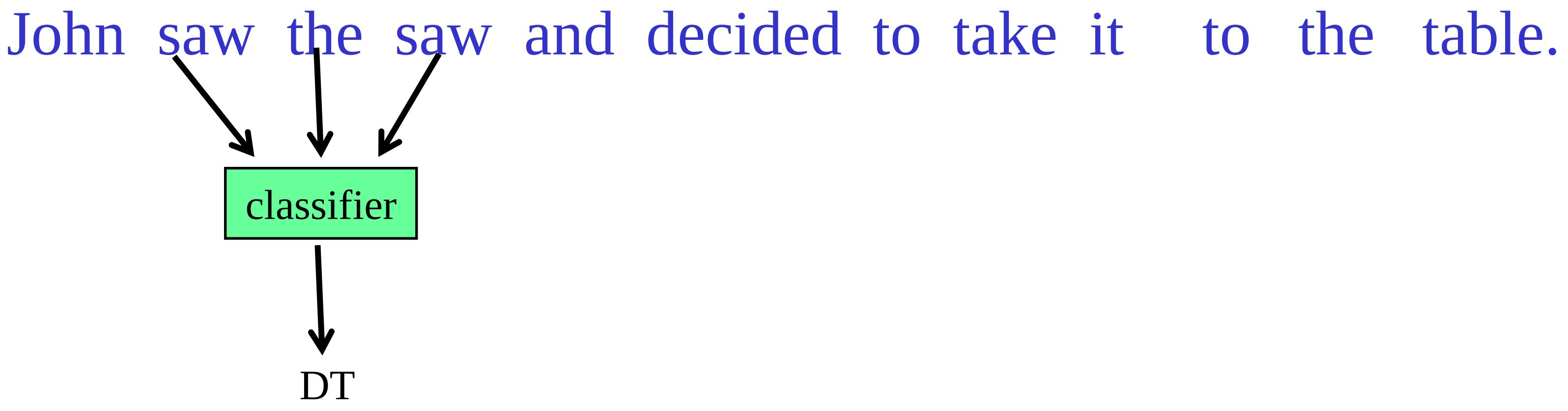
Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Slide courtesy Ray Mooney, with mild edits

Sequence Labeling as Classification

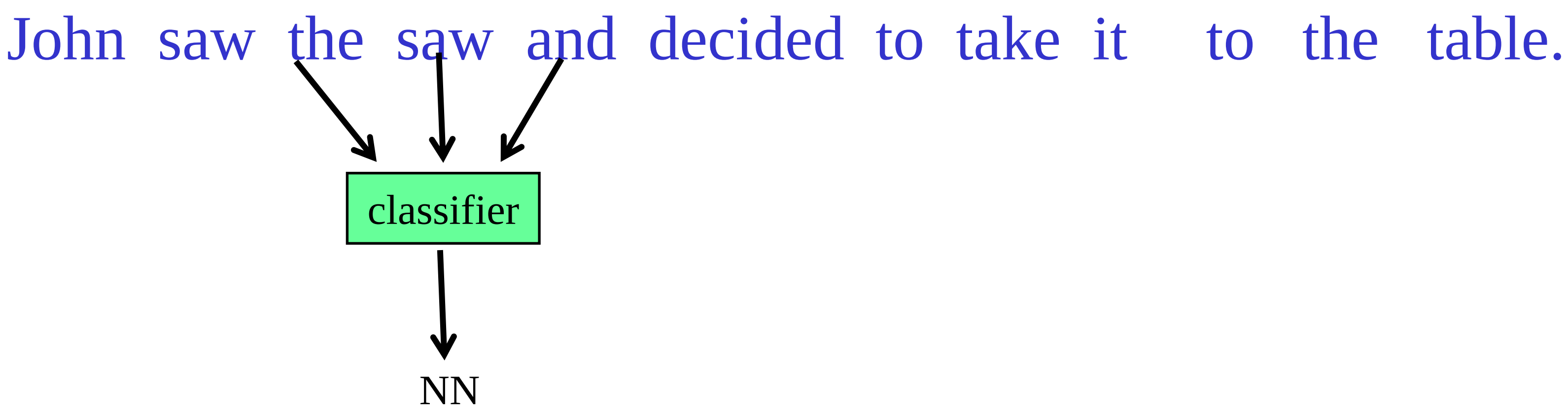
Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Slide courtesy Ray Mooney, with mild edits

Sequence Labeling as Classification

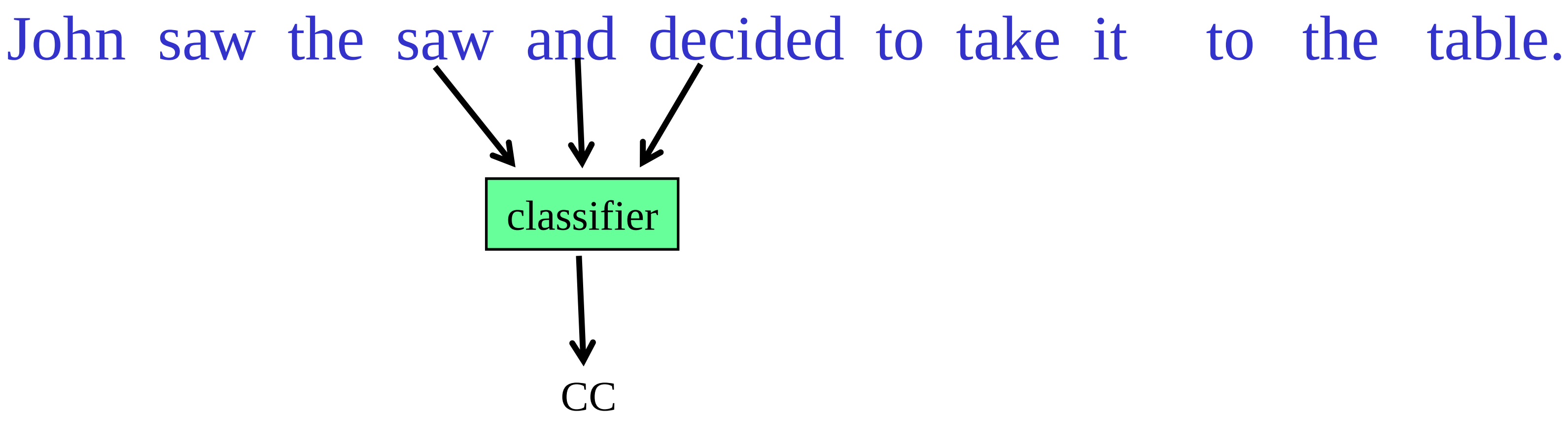
Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Slide courtesy Ray Mooney, with mild edits

Sequence Labeling as Classification

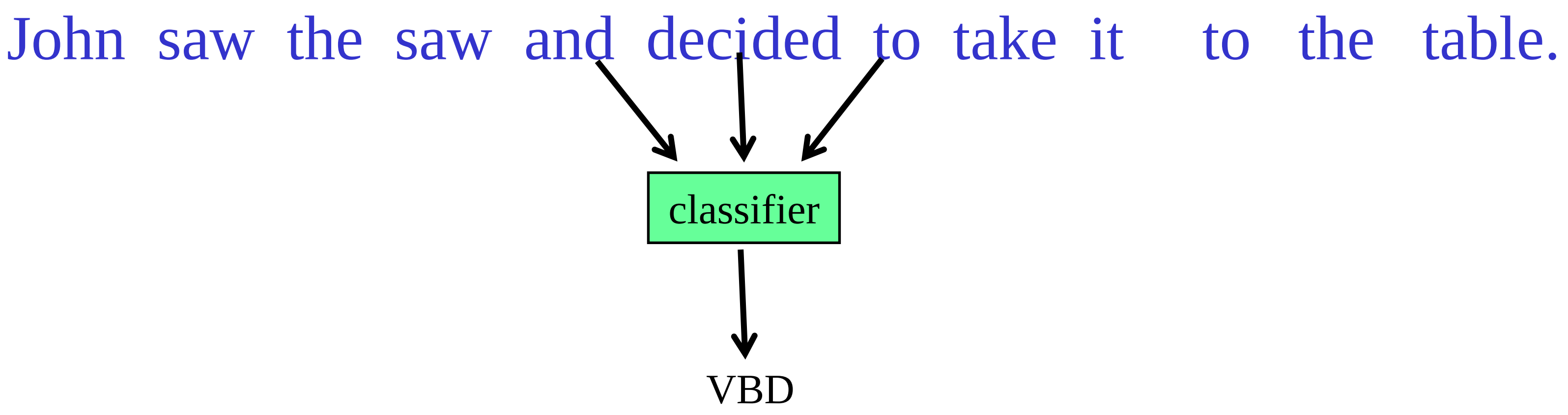
Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Slide courtesy Ray Mooney, with mild edits

Sequence Labeling as Classification

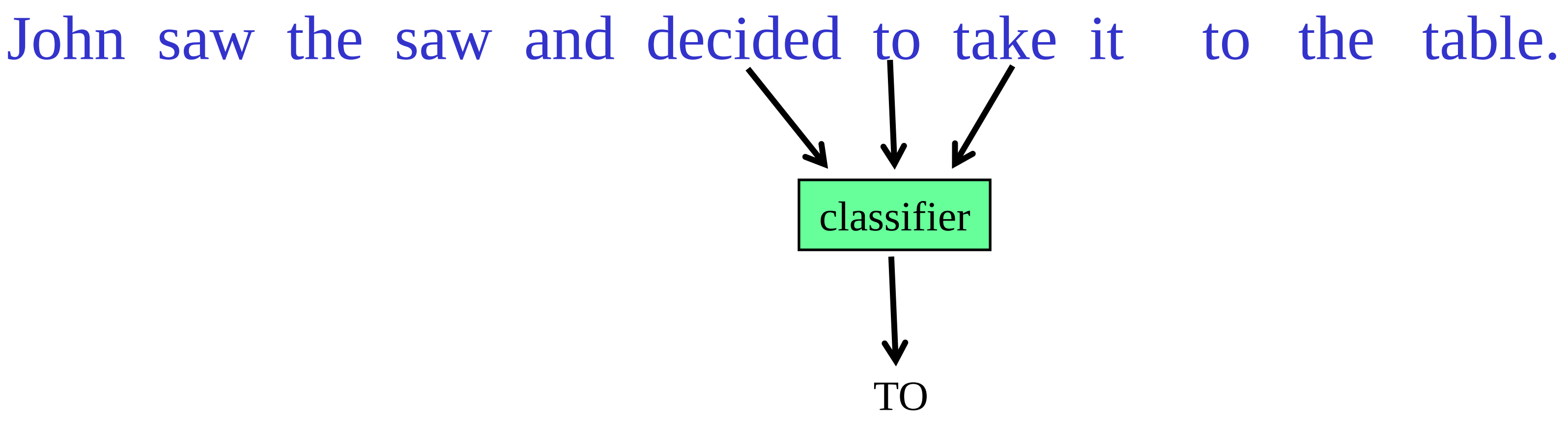
Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Slide courtesy Ray Mooney, with mild edits

Sequence Labeling as Classification

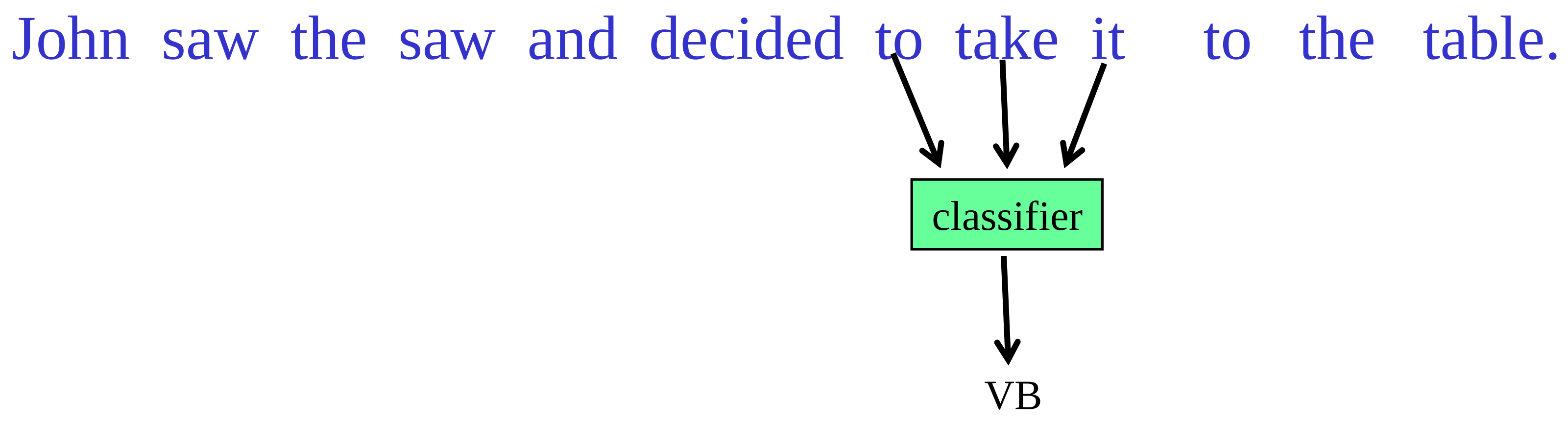
Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Slide courtesy Ray Mooney, with mild edits

Sequence Labeling as Classification

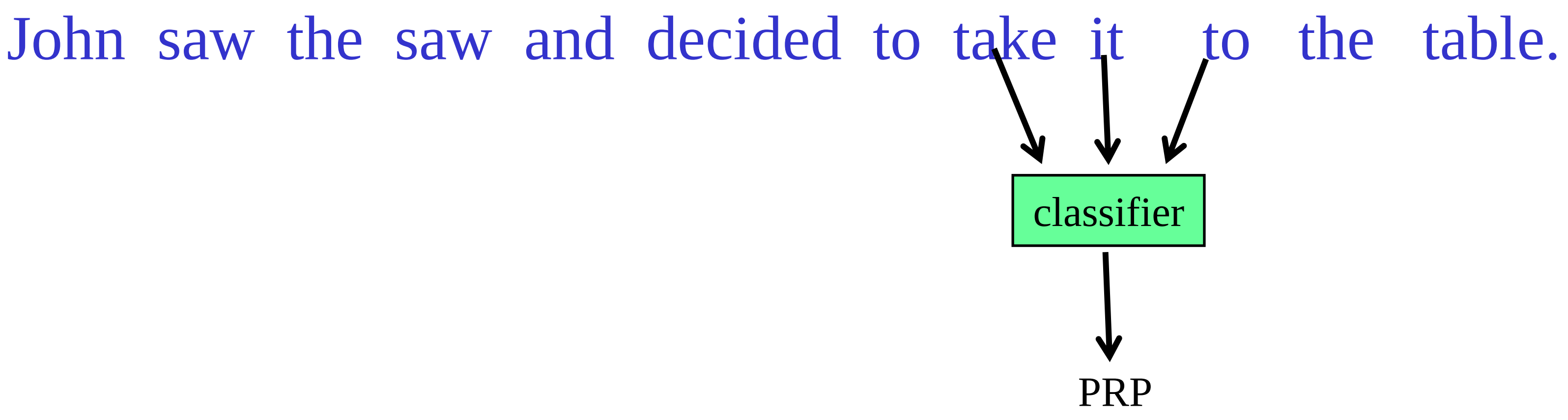
Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Slide courtesy Ray Mooney, with mild edits

Sequence Labeling as Classification

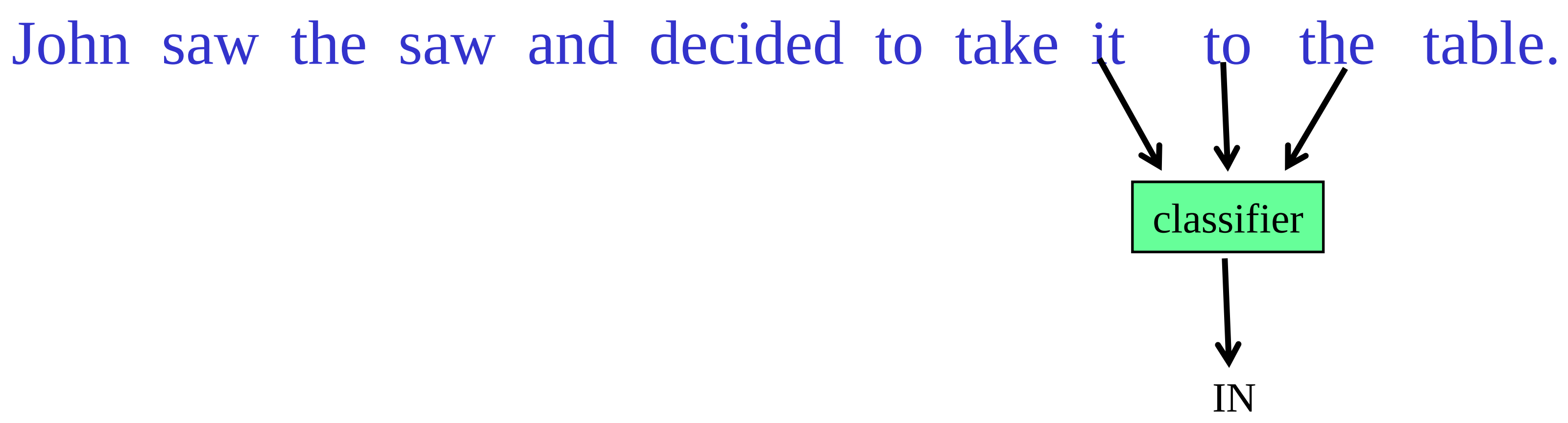
Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Slide courtesy Ray Mooney, with mild edits

Sequence Labeling as Classification

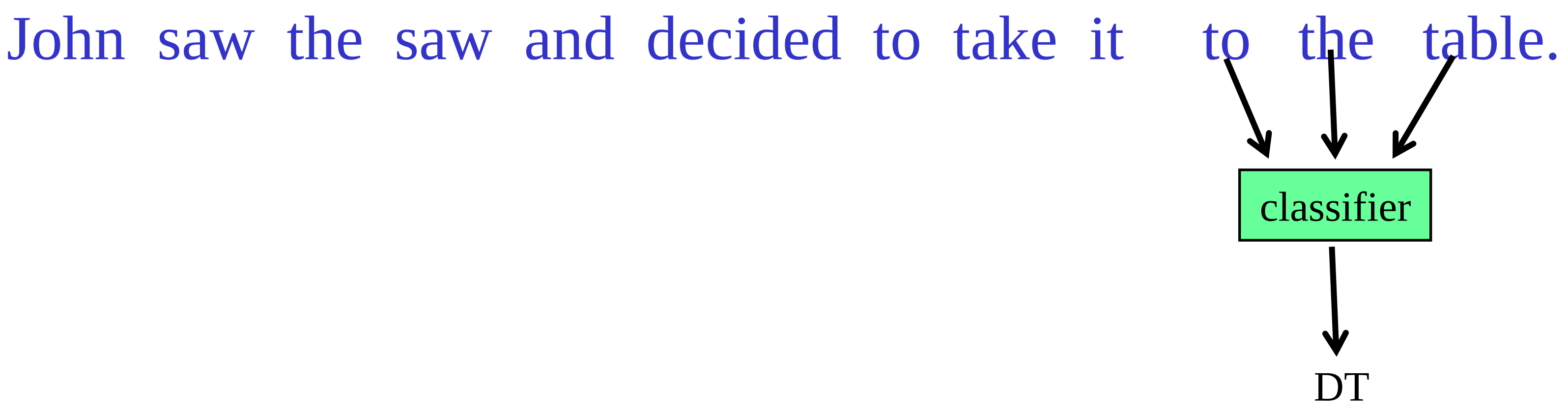
Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Slide courtesy Ray Mooney, with mild edits

Sequence Labeling as Classification

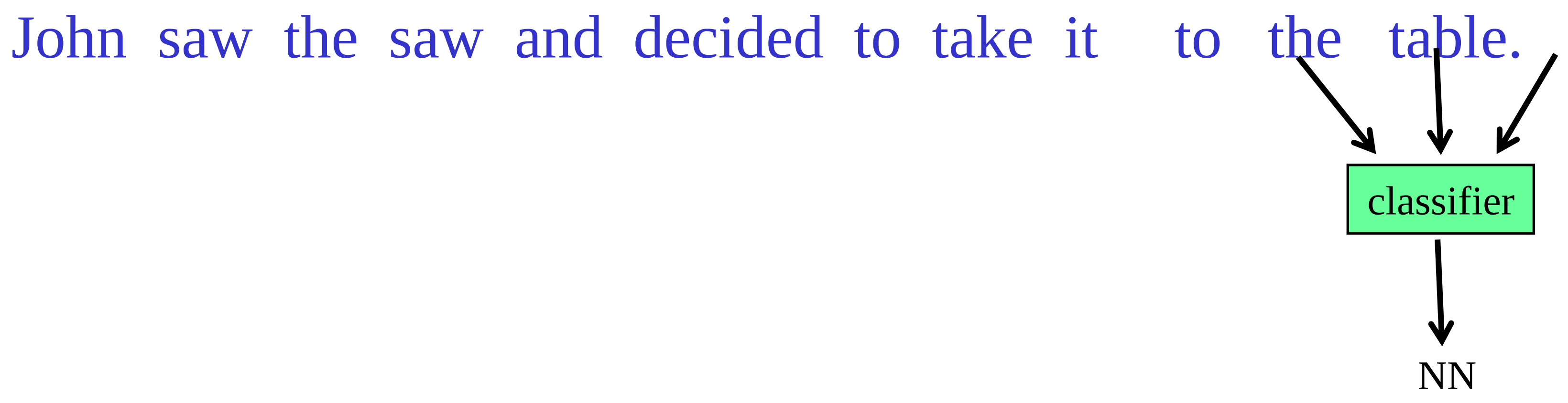
Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Slide courtesy Ray Mooney, with mild edits

Sequence Labeling as Classification

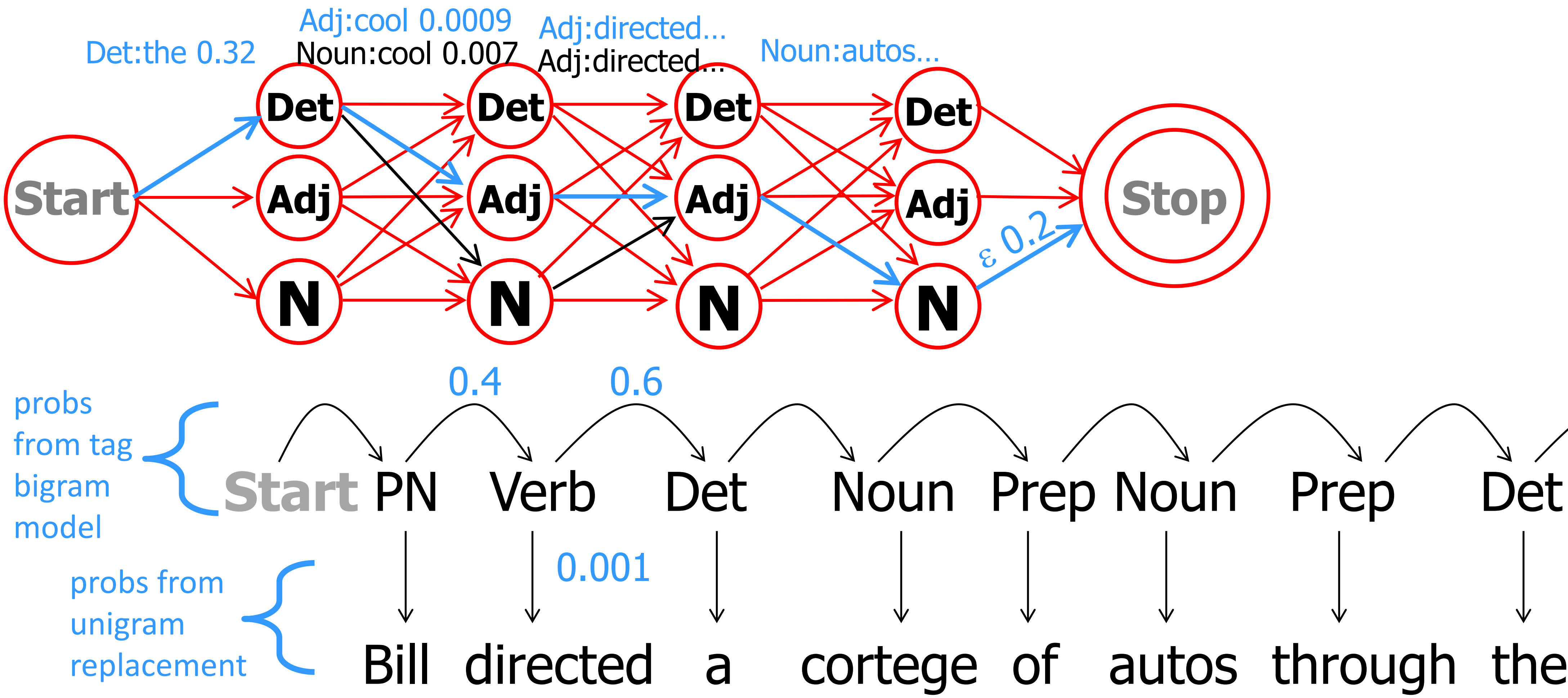
Classify each token independently but use as input features, information about the surrounding tokens (sliding window).



Slide courtesy Ray Mooney, with mild edits

Part of Speech Tagging

Or we could use an HMM:



Part of Speech Tagging

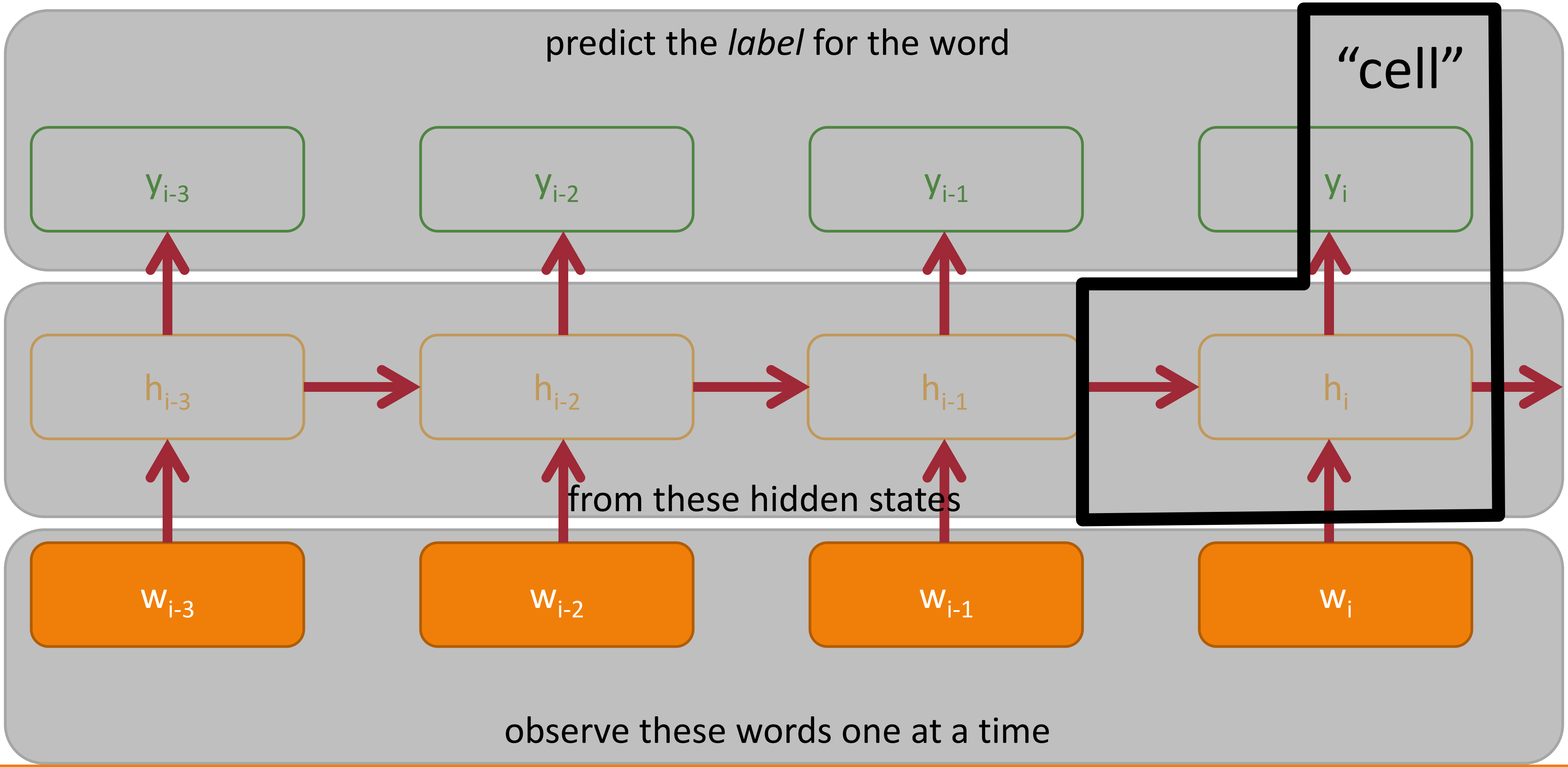
We could treat tagging as a token classification problem
- Tag each word independently given features of context
- And features of the word's spelling (suffixes, capitalization)

Or we could use an HMM:
- The point of the HMM is basically that the tag of one word might depend on the tags of adjacent words.

Combine these two ideas??
- We'd like rich features (e.g., in a log-linear model), but we'd also like our feature functions to depend on adjacent tags.
- So, the problem is to predict **all** tags together.

Slide courtesy Jason Eisner, with mild edits

Can We Use Neural, Recurrent Methods?



Text Annotation Tasks

1. Classify the entire document ("text categorization")
2. Classify word tokens individually
3. Classify word tokens in a sequence
4. Identify phrases ("chunking")
5. Syntactic annotation (parsing)
6. Semantic annotation
7. Text generation

Slide courtesy Jason Eisner, with mild edits

Example: Finding Named Entities

Named entity recognition (NER)

Identify proper names in texts, and classification into a set of predefined categories of interest

- Person names
- Organizations (companies, government organisations, committees, etc.)
- Locations (cities, countries, rivers, etc.)
- Date and time expressions
- Measures (percent, money, weight, etc.),
- email addresses, web addresses, street addresses, etc.
- Domain-specific: names of drugs, medical conditions,
- names of ships, bibliographic references etc.

NE Types

Type	Tag	Sample Categories
People	PER	Individuals, fictional characters, small groups
Organization	ORG	Companies, agencies, political parties, religious groups, sports teams
Location	LOC	Physical extents, mountains, lakes, seas
Geo-Political Entity	GPE	Countries, states, provinces, counties
Facility	FAC	Bridges, buildings, airports
Vehicles	VEH	Planes, trains, and automobiles

Type	Example
People	*Turing* is often considered to be the father of modern computer science.
Organization	The *IPCC* said it is likely that future tropical cyclones will become more intense.
Location	The *Mt. Sanitas* loop hike begins at the base of *Sunshine Canyon*.
Geo-Political Entity	*Palo Alto* is looking at raising the fees for parking in the University Avenue district.
Facility	Drivers were advised to consider either the *Tappan Zee Bridge* or the *Lincoln Tunnel*.
Vehicles	The updated *Mini Cooper* retains its charm and agility.

Slide courtesy Jim Martin

Named Entity Recognition

CHICAGO (AP) — Citing high fuel prices, United Airlines said Friday it has increased fares by $6 per round trip on flights to some cities also served by lower-cost carriers. American Airlines, a unit AMR, immediately matched the move, spokesman Tim Wagner said. United, a unit of UAL, said the increase took effect Thursday night and applies to most routes where it competes against discount carriers, such as Chicago to Dallas and Atlanta and Denver to San Francisco, Los Angeles and New York.

Slide courtesy Jim Martin