CMSC 473/673
Natural Language Processing

Instructor: Lara J. Martin (she/they)

TA: Duong Ta (he)

Slides modified from Dr. Frank Ferraro

Logistics

Finish the Catme – I will assign teams tomorrow!!

I’m working on HW 1 & grad assignment, hoping to release them soon

Learning Objectives

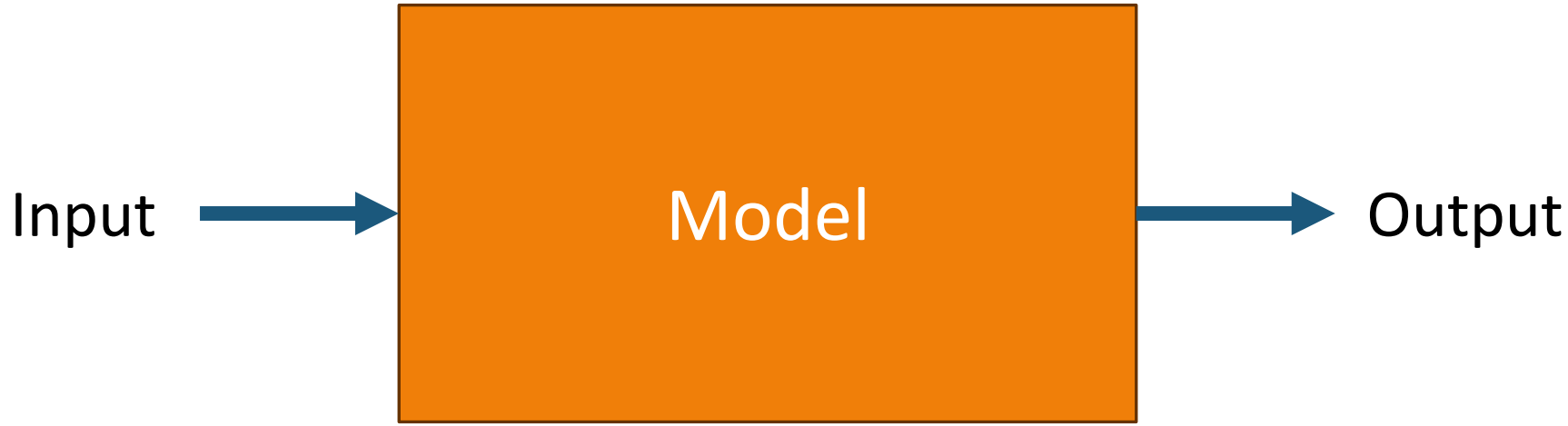
Formalize NLP Tasks at a high-level:

- What are the input/output for a particular task?
- What might the features be?
- What types of applications could the task be used for?

Calculate elementary processes on a dataset

Similar to what HW 1 will be

Review: Terminology



Learning vs decoding:

Training the model vs using the model

Review: 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

How are any of these fed to a model?

Review: 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			

Class vs Label vs Task?

Review: 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
6. Semantic annotation
7. Text generation

Slide courtesy Jason Eisner, with mild edits

Document Categorization/Classification

Assigning subject categories, topics, or genres

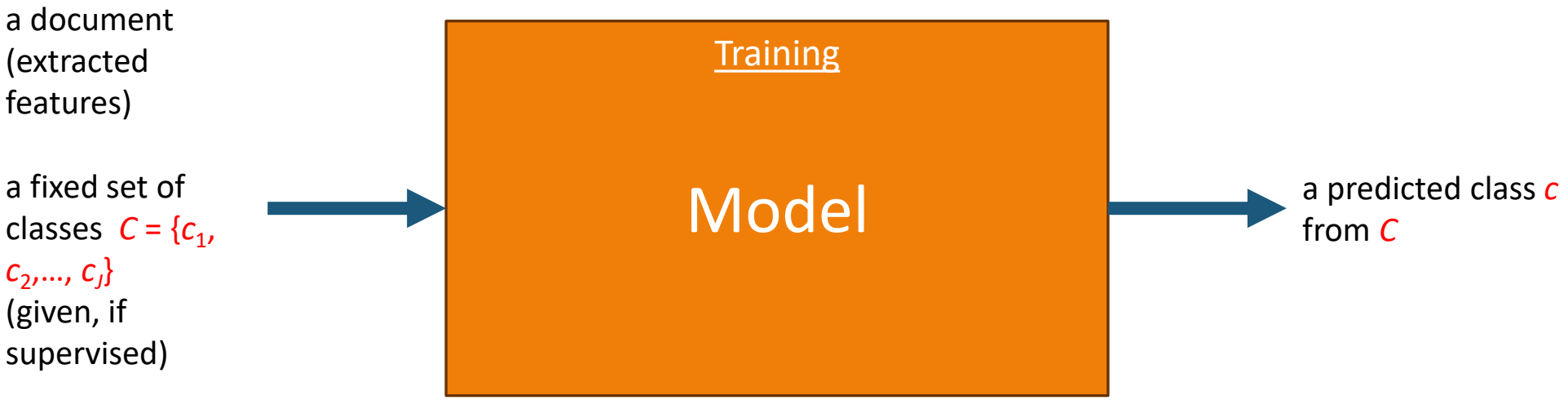
Spam detection

Authorship identification

Language Identification

Sentiment analysis

…



Document Categorization/Classification

Assigning subject categories, topics, or genres

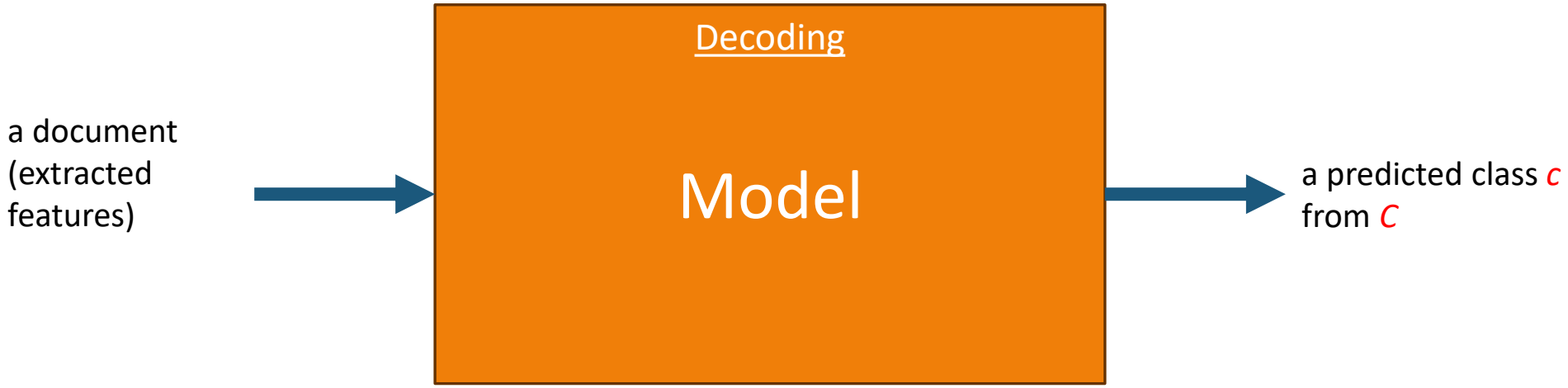
Spam detection

Authorship identification

Language Identification

Sentiment analysis

…



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
6. Semantic annotation
7. Text generation

Slide courtesy Jason Eisner, with mild edits

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)

Token Classification

Word pronunciation

Word sense disambiguation (WSD) within or across languages

Accent restoration

…

a word (extracted features)

the word's context

a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$ (given, if supervised)

a predicted class c from C

Example of features for token classification

	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)

Example of features for token classification

	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
10.51	lead *role*	⇒ li:d

slide courtesy of D. Yarowsky (modified)

Final decision list for ***lead*** (abbreviated)

List of all features, ranked by their "likelihood" from looking at all the features together.

LogL	Evidence	Pronunciation
11.40	*follow/V* + lead	⇒ li:d
11.20	*zinc* (in $\pm k$ words)	⇒ lɛd
11.10	lead *level/N*	⇒ lɛd
10.66	*of* lead *in*	⇒ lɛd
10.59	*the* lead *in*	⇒ li:d
10.51	lead *role*	⇒ li:d
10.35	*copper* (in $\pm k$ words)	⇒ lɛd
10.28	lead *time*	⇒ li:d
10.24	lead *levels*	⇒ lɛd
10.16	lead *poisoning*	⇒ lɛd
8.55	*big* lead	⇒ li:d
8.49	*narrow* lead	⇒ li:d
7.76	*take/V* + lead	⇒ li:d
5.99	lead , *NOUN*	⇒ lɛd
1.15	lead *in*	⇒ 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
6. Semantic annotation
7. Text generation

Slide courtesy Jason Eisner, with mild edits

Token Classification in a Sequence

Part of speech tagging

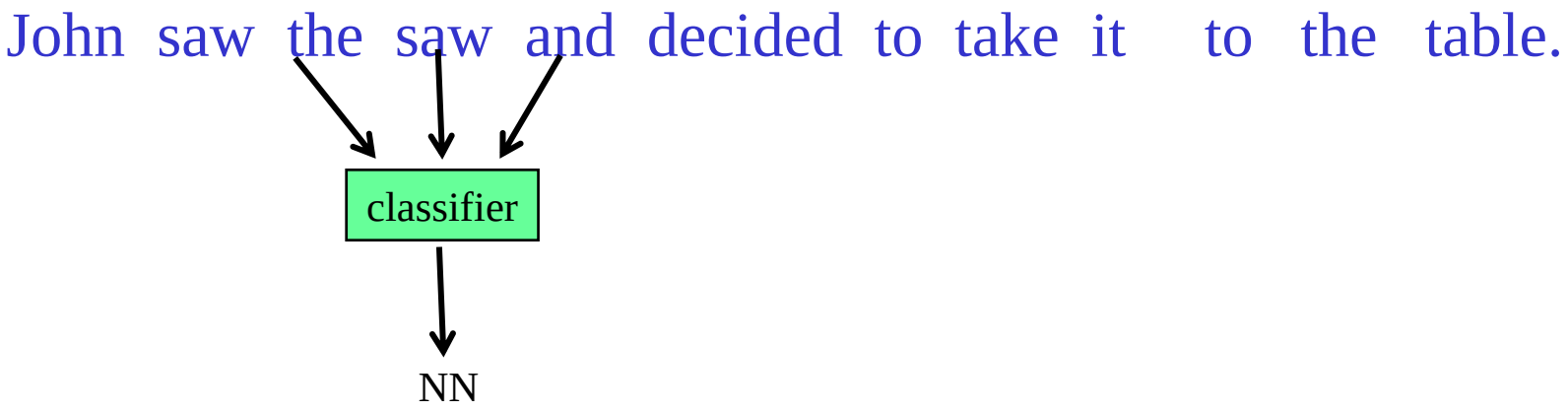
a sequence (extracted features)

a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$ (given for every element of the sequence, if supervised)

predicted classes c from C for every element of the sequence

Part of Speech (POS) Tagging

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

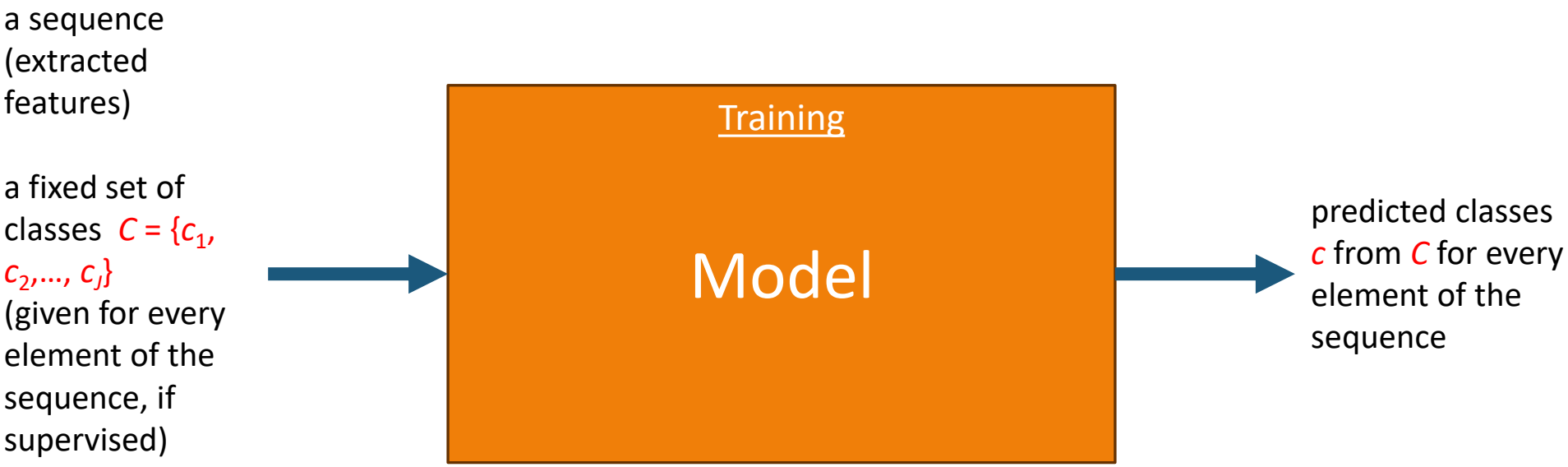


Slide courtesy Ray Mooney, with mild edits

Token Classification in a Sequence

Part of speech tagging

Word alignment



Machine Translation: Word Alignment

What kinds of features might we want to consider here?

https://towardsdatascience.com/machine-translation-a-short-overview-91343ff39c9f

Token Classification in a Sequence

Part of speech tagging

Word alignment

...

a sequence (extracted features)

a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$ (given for every element of the sequence, if supervised)

predicted classes c from C for every element of the sequence

In-Class Assignment

- 10 minutes to do it in class
- You can complete it after class
- Then submit it to Google Classroom

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
6. Semantic annotation
7. Text generation

Slide courtesy Jason Eisner, with mild edits

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

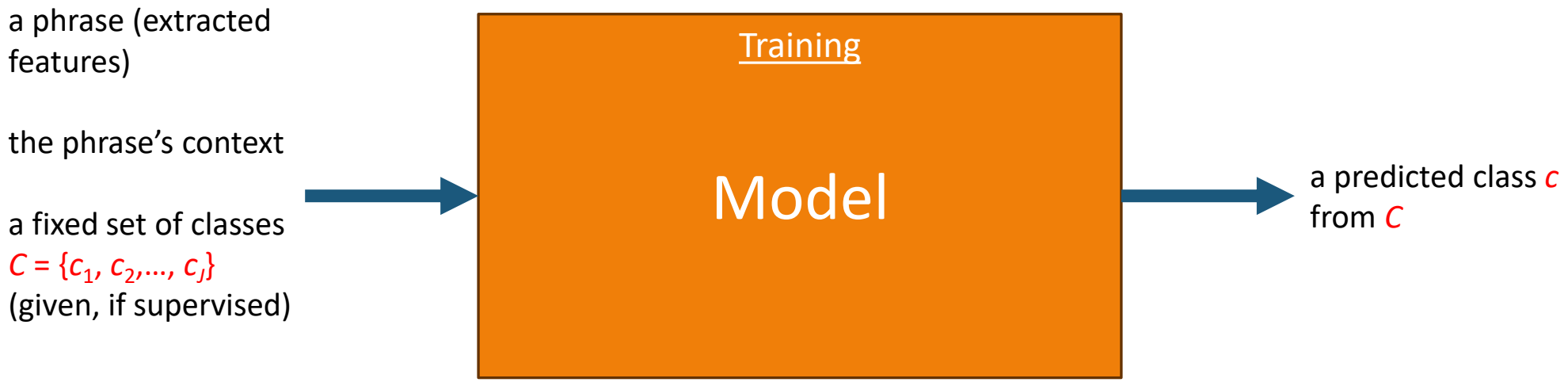
CHICAGO (AP) — Citing high fuel prices, United Airlines said Friday it has increased fares by $6 per round trip on flights to some cities also served by lower-cost carriers. American Airlines, a unit AMR, immediately matched the move, spokesman Tim Wagner said. United, a unit of UAL, said the increase 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

Chunking

Named entity recognition

Information extraction



Example: Information Extraction

As a task: **Filling slots in a database from sub-segments of text.**

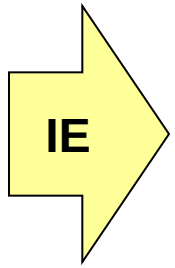
October 14, 2002, 4:00 a.m. PT

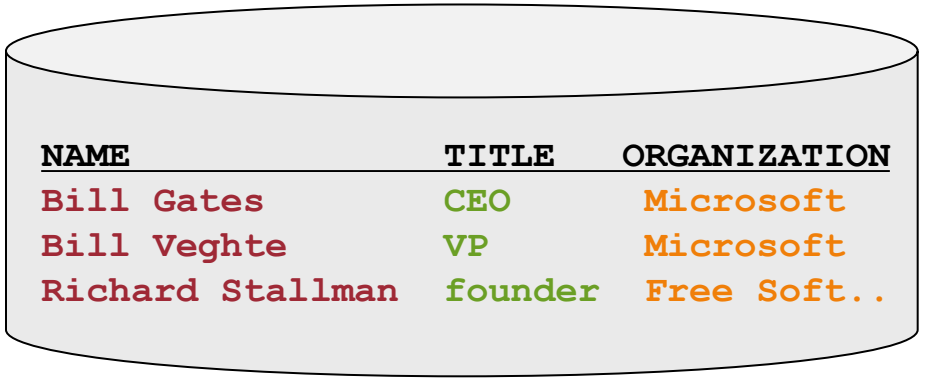
For years, Microsoft Corporation CEO Bill Gates railed against the economic philosophy of open-source software with Orwellian fervor, denouncing its communal licensing as a "cancer" that stifled technological innovation.

Today, Microsoft claims to "love" the open-source concept, by which software code is made public to encourage improvement and development by outside programmers. Gates himself says Microsoft will gladly disclose its crown jewels--the coveted code behind the Windows operating system--to select customers.

"We can be open source. We love the concept of shared source," said Bill Veghte, a Microsoft VP. "That's a super-important shift for us in terms of code access."

Richard Stallman, founder of the Free Software Foundation, countered saying…





NAME	TITLE	ORGANIZATION
Bill Gates	CEO	Microsoft
Bill Veghte	VP	Microsoft
Richard Stallman	founder	Free Soft..

Slide from Chris Brew, adapted from slide by William Cohen

Example applications for IE

Classified ads

Restaurant reviews

Bibliographic citations

Appointment emails

Legal opinions

Papers describing clinical medical studies

Task vs application?

Slide courtesy Jason Eisner, with mild edits

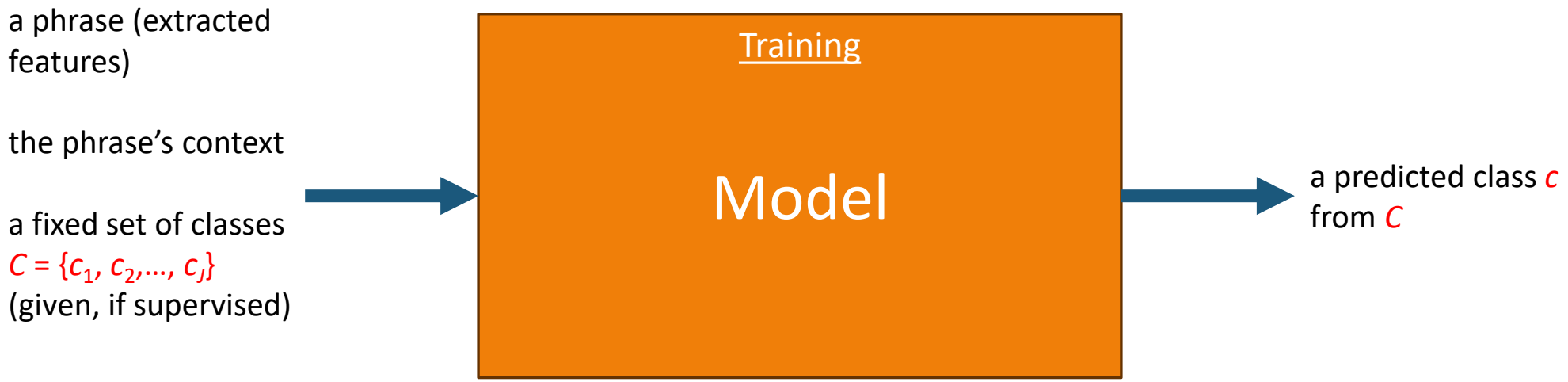
Chunking

Named entity recognition

Information extraction

Identifying idioms

…



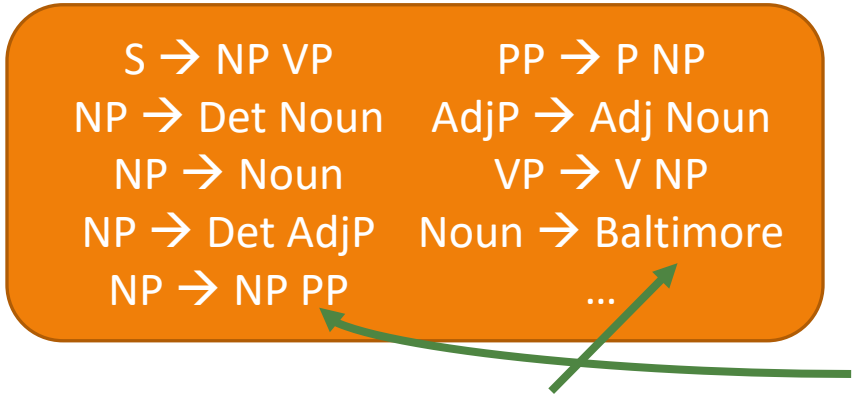
Text Annotation Tasks

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 (syntax parsing)
6. Semantic annotation
7. Text generation

Slide courtesy Jason Eisner, with mild edits

Syntax Parsing

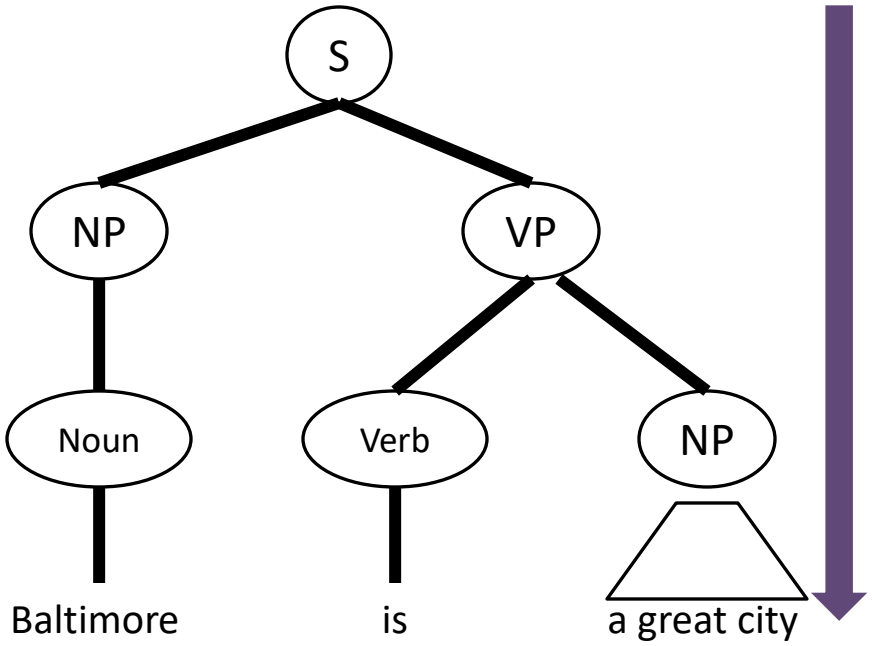
Context Free Grammar



Set of rewrite rules, comprised of terminals and non-terminals

Generate from a Context Free Grammar

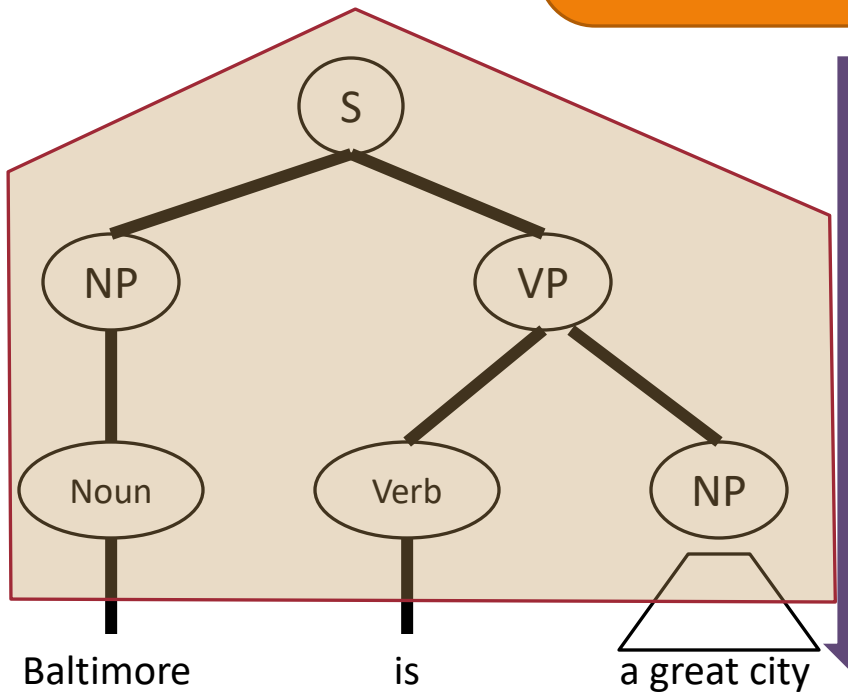
S → NP VP	PP → P NP
NP → Det Noun	AdjP → Adj Noun
NP → Noun	VP → V NP
NP → Det AdjP	Noun → Baltimore
NP → NP PP	…



Baltimore is a great city

Assign Structure (Parse) with a Context Free Grammar

S → NP VP	PP → P NP
NP → Det Noun	AdjP → Adj Noun
NP → Noun	VP → V NP
NP → Det AdjP	Noun → Baltimore
NP → NP PP	…



Baltimore is a great city

$[_S\ [_{NP}\ [_{Noun}$ Baltimore]] $[_{VP}\ [_{Verb}$ is] $[_{NP}$ a great city]]]

bracket notation

(S (NP (Noun Baltimore))
(VP (V is)
(NP a great city)))

S-expression

Why is it useful?

Garden Path Sentences

The old man the boat .

Garden Path Sentences

The old man the boat .

Garden Path Sentences

The rat the cat the dog chased killed ate the malt.

Garden Path Sentences

The rat *that* the cat the dog chased killed ate the malt.

Garden Path Sentences

The rat *that* the cat *that* the dog chased killed ate the malt.

Garden Path Sentences

The rat *that* the cat *that* the dog chased killed ate the malt.

Garden Path Sentences

The rat *that* the cat *that* the dog chased killed ate the malt.

Garden Path Sentences

The rat *that* the cat *that* the dog chased killed ate the malt.

Garden Path Sentences

[The rat [the cat [the dog chased] killed] ate the malt].

Language can have recursive patterns

Syntactic parsing can help identify those

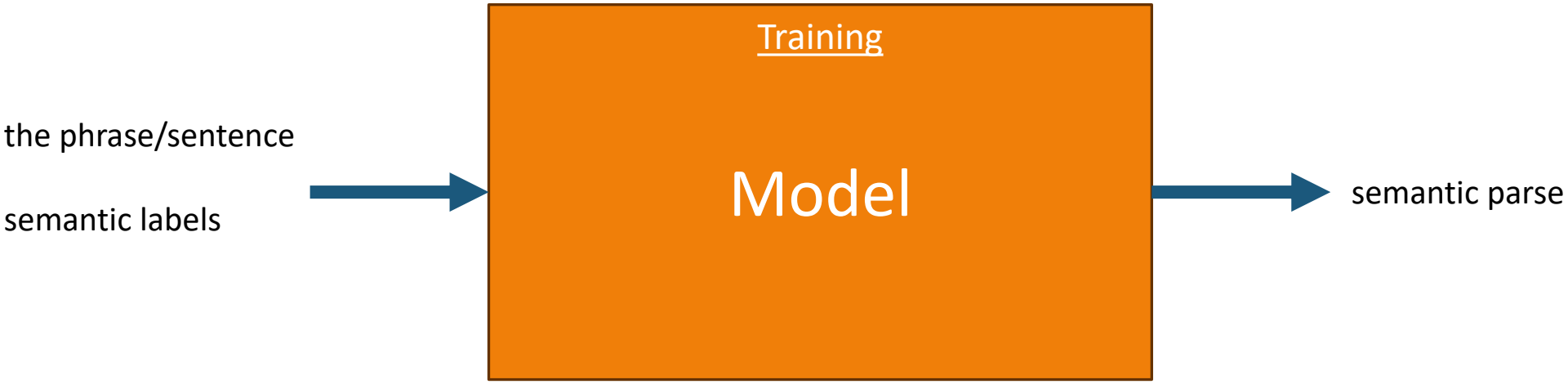
Text Annotation Tasks

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 (syntax parsing)
6. Semantic annotation
7. Text generation

Slide courtesy Jason Eisner, with mild edits

Semantic Parsing

Semantic role labeling (SRL)



Semantic Role Labeling (SRL)

For each predicate (e.g., verb)

1. find its arguments (e.g., NPs)
2. determine their **semantic roles**

John drove Mary from Austin to Dallas in his Toyota Prius.

The hammer broke the window.

- agent: Actor of an action
- patient: Entity affected by the action
- source: Origin of the affected entity
- destination: Destination of the affected entity
- instrument: Tool used in performing action.
- beneficiary: Entity for whom action is performed

Slide thanks to Ray Mooney (modified)

Other Current Semantic Annotation Tasks (similar to SRL)

PropBank – coarse-grained roles of verbs

NomBank – similar, but for nouns

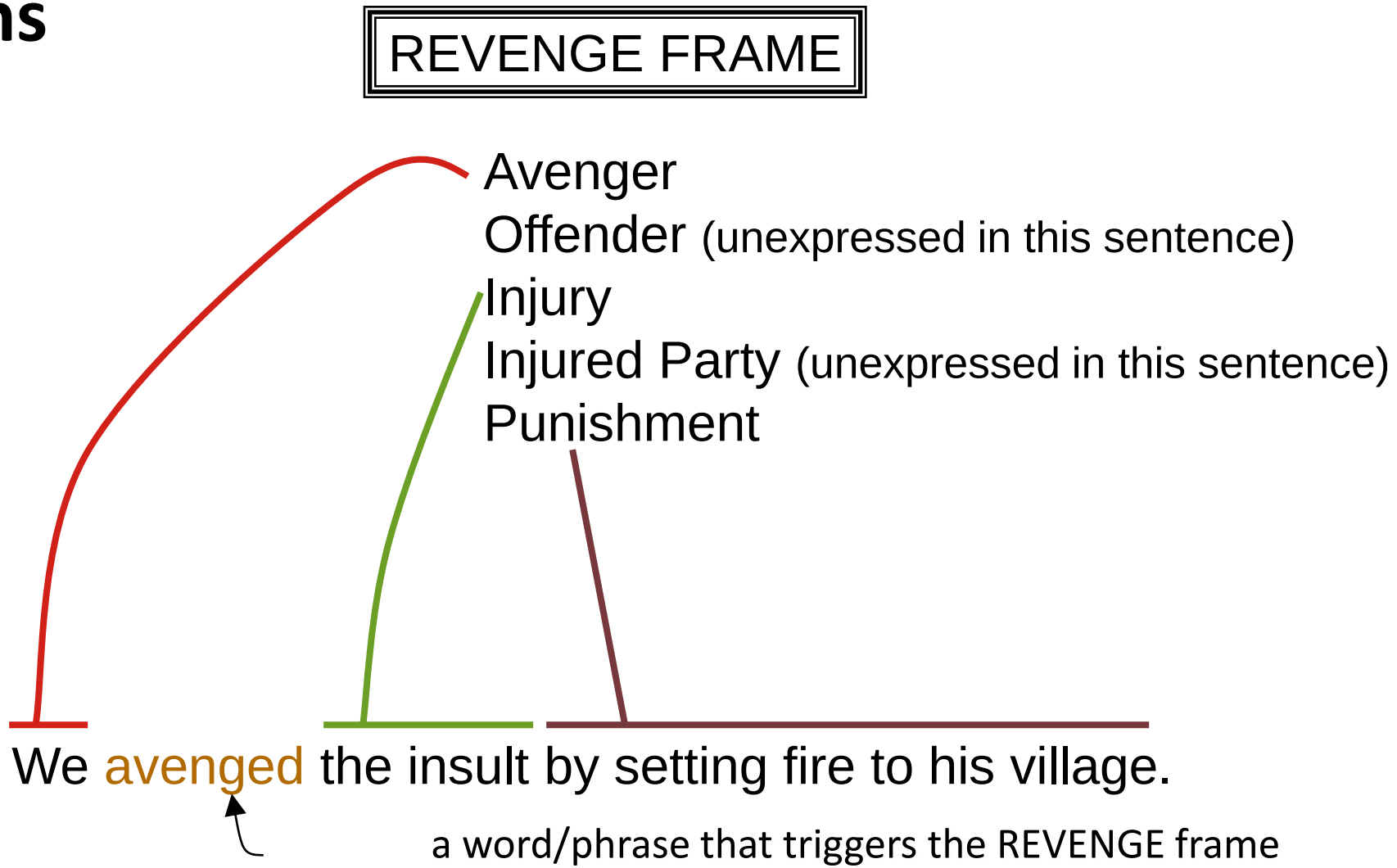
FrameNet – fine-grained roles of any word

TimeBank – temporal expressions

Slide courtesy Jason Eisner, with mild edits

What type of applications might this have?

FrameNet Example



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 (syntax parsing)
6. Semantic annotation
7. Text generation

Slide courtesy Jason Eisner, with mild edits

Text Generation

Question answering (QA)

Speech recognition (ASR)

Machine translation (MT)

Summarization

Generating text from a structured representation

…