TTIC 31210: Advanced Natural Language Processing

Kevin Gimpel

Spring 2019

Lecture 1: Introduction

Course Overview

- Second time being offered (first was spring 2017)

- Prerequisite: TTIC 31190 (NLP)

- Aimed at second & third year PhD students

- My office hours (TTIC 531):
  - Mondays after class until 3:15pm
  - Wednesdays after class until 4pm
  - or by appointment

- Teaching assistant: Mingda Chen, 3rd year TTIC PhD student
  - TA office hours: Mondays 3-4pm, TTIC library (4th floor)

Course Web Page

```
https://ttic.uchicago.edu/~kgimpel/teaching/31210-s19/index.html
```

TTIC 31210: Advanced Natural Language Processing

lectures assignments

This is the course webpage for the Spring 2019 version of TTIC 31210: Advanced Natural Language Processing.
For the Spring 2017 course, go **here**.

Quarter: Spring 2019
Time: Monday/Wednesday 1:30-2:50pm
Location: Room 526 (fifth floor), TTIC

Instructor: Kevin Gimpel
Instructor Office Hours: Mondays 2:50-3:15pm, Wednesdays 2:50-4pm, Room 531
Teaching Assistant: Mingda Chen
Teaching Assistant Office Hours: Mondays 3-4pm, TTIC Library (fourth floor)

Prerequisites: TTIC 31190 or permission of the instructor.

Contents:

Roadmap

- intro (today)
- deep learning for NLP (5 lectures)
- structured prediction: sequence labeling, syntactic and semantic parsing, dynamic programming (4 lectures)
- generative models, latent variables, unsupervised learning, variational autoencoders (2 lectures)
- Bayesian methods in NLP (2 lectures)
- Bayesian nonparametrics in NLP (2 lectures)
- review & other topics (1 lecture)

- I will be away at a conference (NAACL) the last week of classes (June 3-5), so we will cancel those two classes

Assignments

- Mini-research projects: implementation, experimentation, analysis, developing new methods
- Assignment 1 has been posted; due April 16

Assignments

- 1 (due ~4/16):
  - language modeling: loss function comparison, error analysis
- 2 (due ~4/30):
  - attention in text classification, self-attention, multiple heads
- 3 (due ~5/14):
  - exact and approximate decoding for hidden Markov models for part-of-speech tagging
- 4 (due ~5/29):
  - Gibbs sampling for inference in hidden Markov models and unsupervised part-of-speech tagging
- 5 (due ~6/12):
  - unsupervised tokenization with Bayesian nonparametrics

Grading

- 5 assignments (15% for each)
  - 5th assignment will be due during finals week
  - no final exam
- class participation (25%)
  - includes coming to class, participating, submitting occasional handouts, rare in-class quizzes
- if you have good reason to miss class, let me know!

Collaboration Policy

- You are welcome to discuss assignments with others in the course, but solutions and code must be written individually

Lateness Policy

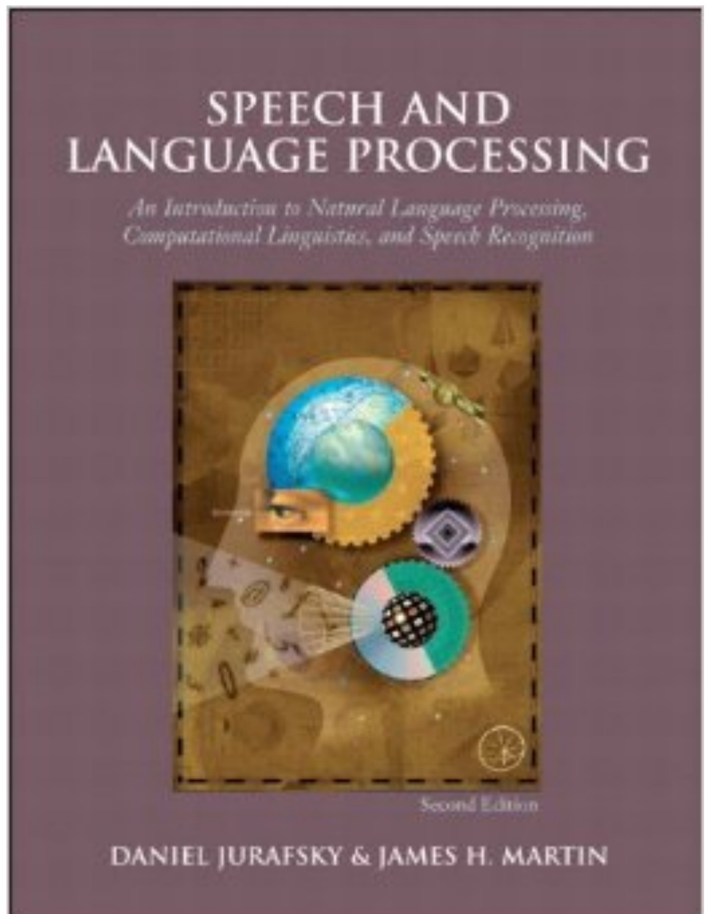
- If you turn in an assignment late, a penalty will be assessed (2% per hour late)

- You will have 4 late days to use as you wish during the quarter

- Late days must be used in whole increments
  - e.g., if you turn in an assignment 6 hours late and want to use a late day to avoid penalty, it will cost an entire late day to do so

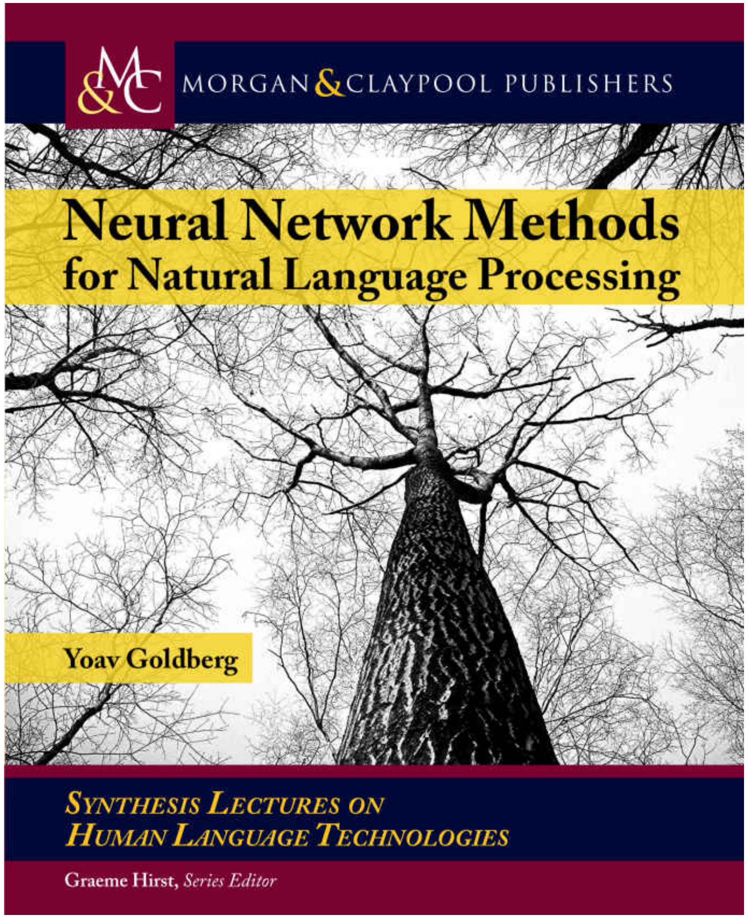
Optional Textbooks (1/3)

- Jurafsky & Martin. *Speech and Language Processing*, 2nd Ed. & 3rd Ed.
- Many chapters of 3rd edition are online
- Copies of 2nd edition available in TTIC library



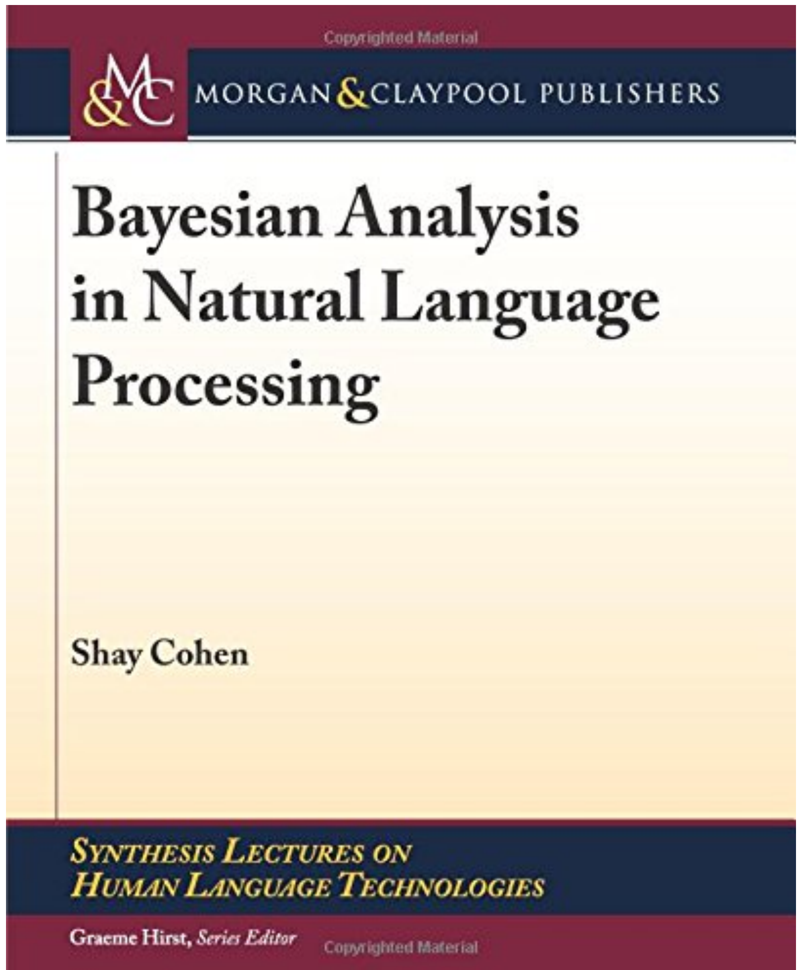
Optional Textbooks (2/3)

- Goldberg. *Neural Network Methods for Natural Language Processing*.
- Earlier draft (from 2015) available online
- Two copies on reserve in TTIC library



Optional Textbooks (3/3)

- Cohen. *Bayesian Analysis in Natural Language Processing*.
- Available in TTIC library



TTIC 31190 Topics

- words, morphology, lexical semantics
- text classification
- simple neural methods for NLP
- language modeling and word embeddings
- recurrent/recursive/convolutional networks in NLP
- sequence labeling, HMMs, dynamic programming
- syntax and syntactic parsing
- semantics, compositionality, semantic parsing
- machine translation and other NLP tasks

What is natural language processing?

What is natural language processing?

an experimental computer science research area that includes problems and solutions related to the understanding of human language

User-Facing Applications

Supporting Technologies

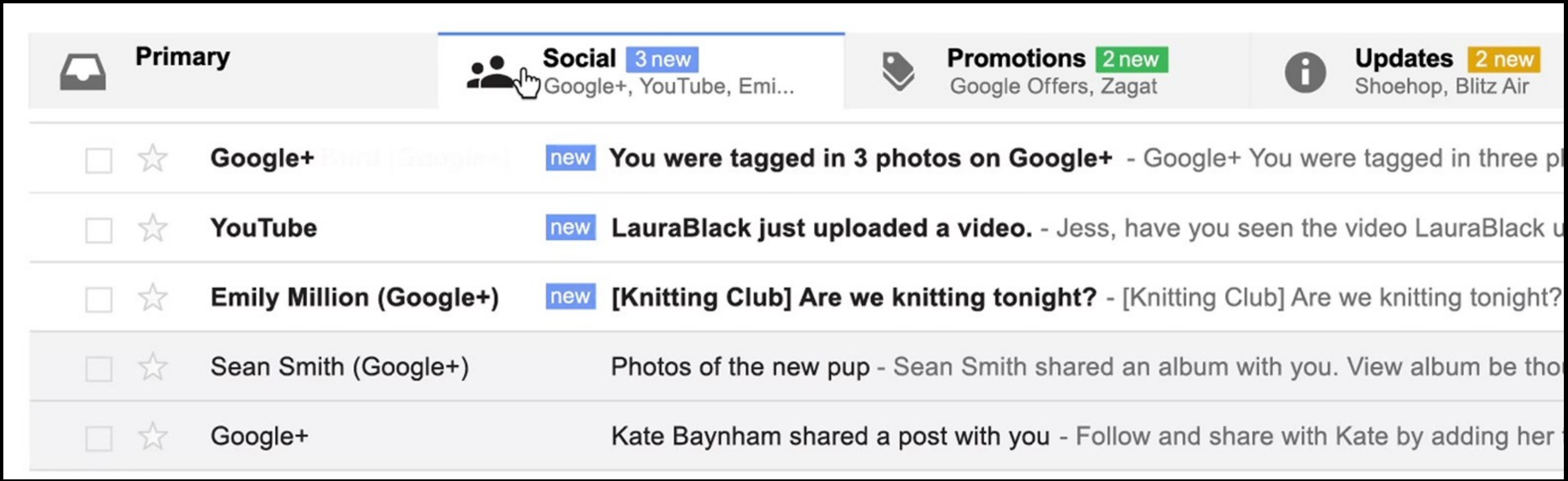
Language Understanding Capabilities

User-Facing Applications

Supporting Technologies

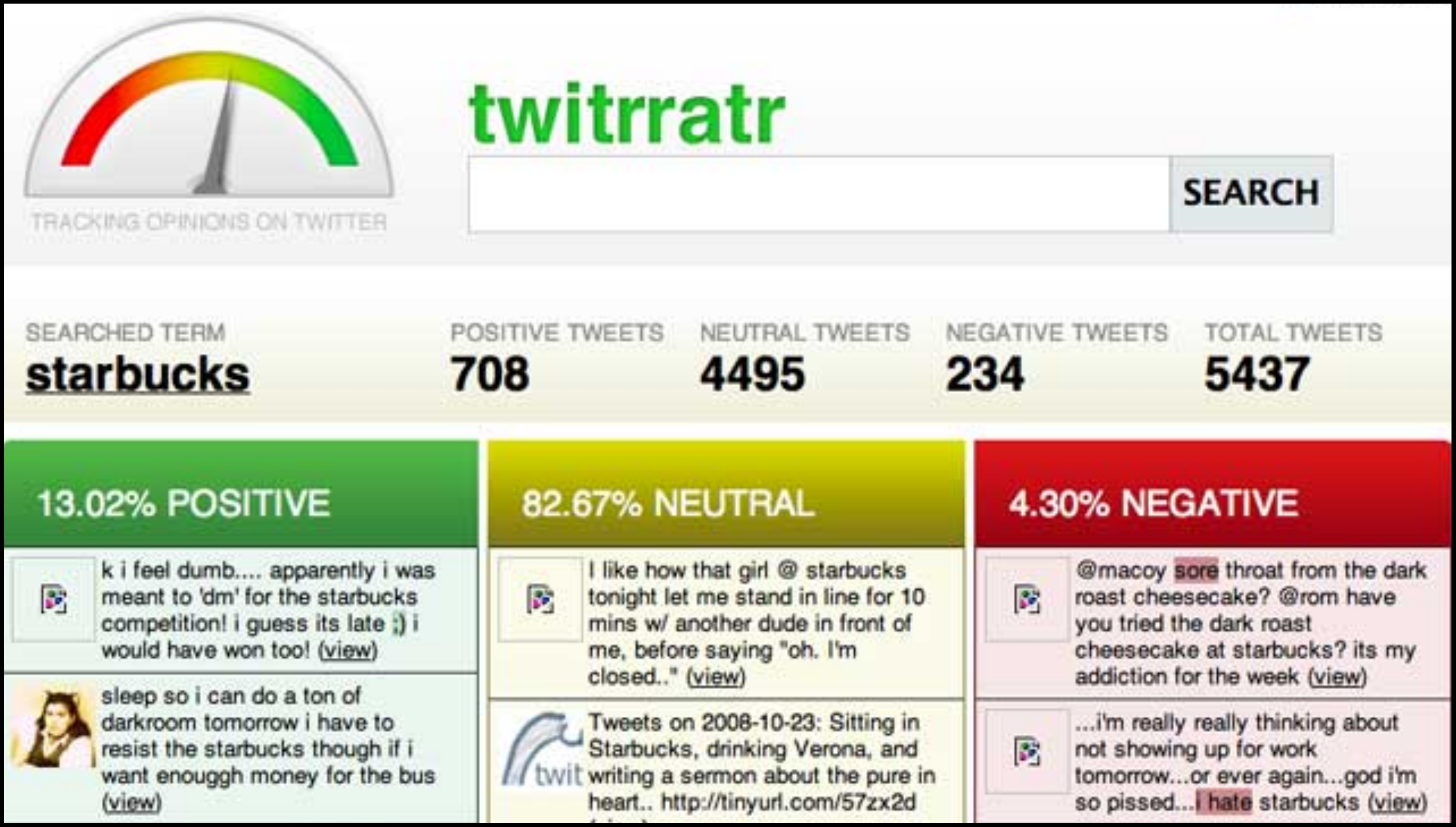
Language Understanding Capabilities

Text Classification

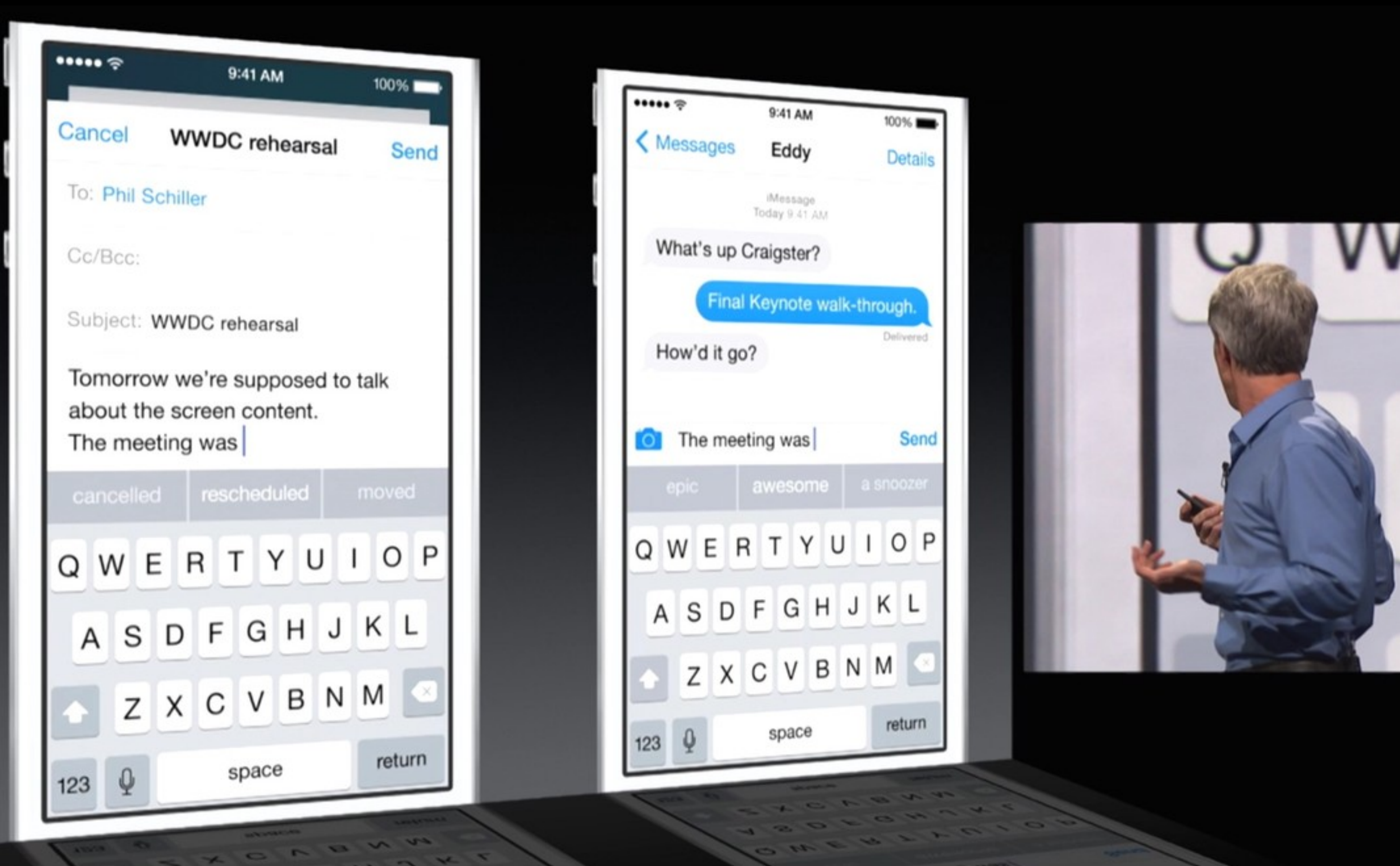


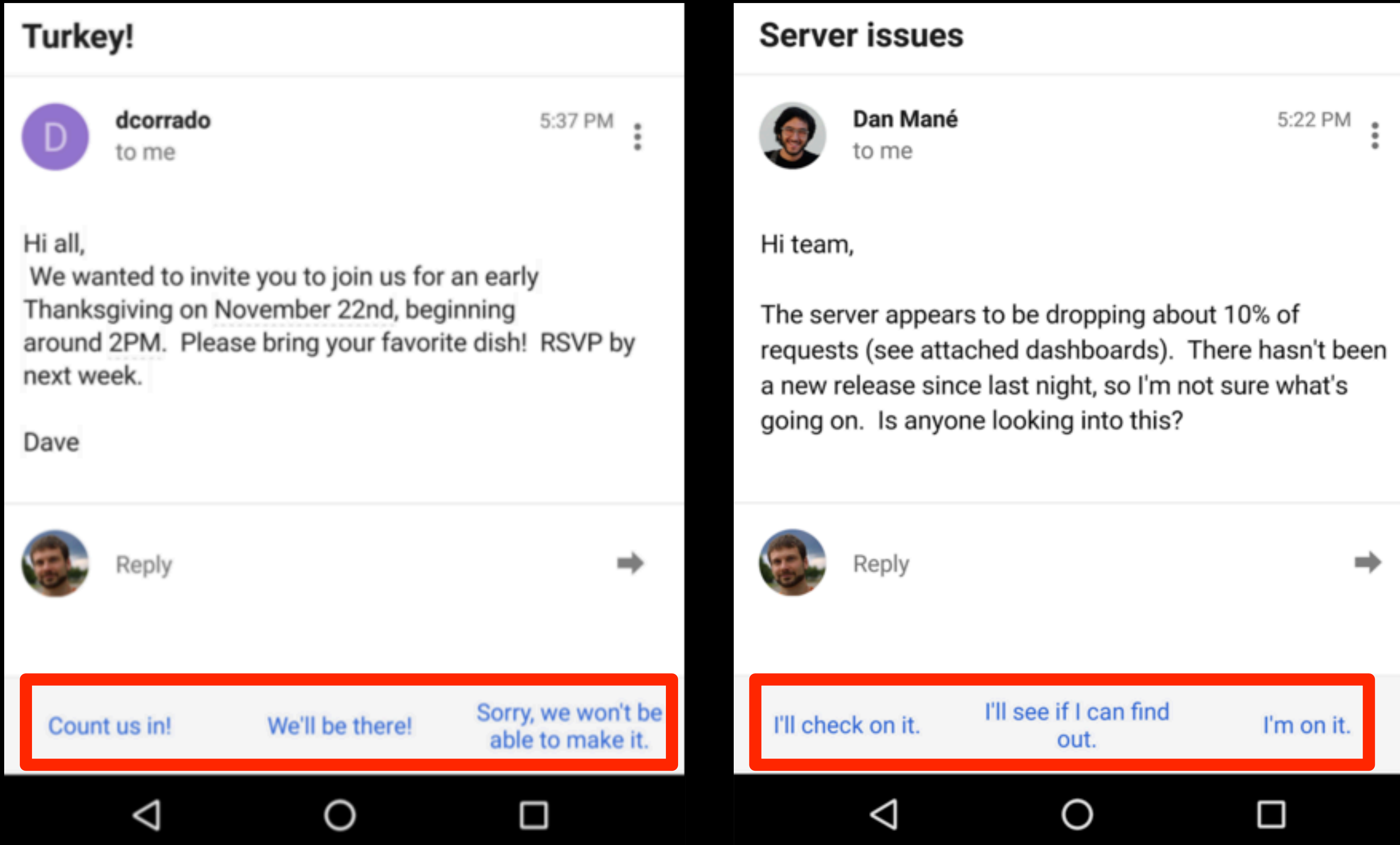
- spam / not spam
- priority level
- category
- sentiment

Sentiment Analysis



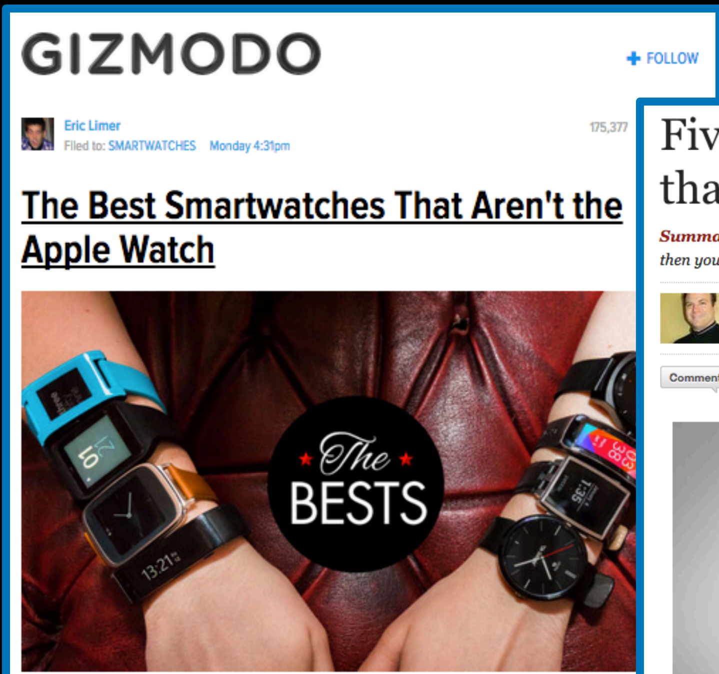
Auto-Complete




Turkey!
dcorrado
to me
5:37 PM
Hi all,
We wanted to invite you to join us for an early Thanksgiving on November 22nd, beginning around 2PM. Please bring your favorite dish! RSVP by next week.
Dave
Reply
Count us in!
We'll be there!
Sorry, we won't be able to make it.
Server issues
Dan Mané
to me
5:22 PM
Hi team,
The server appears to be dropping about 10% of requests (see attached dashboards). There hasn't been a new release since last night, so I'm not sure what's going on. Is anyone looking into this?
Reply
I'll check on it.
I'll see if I can find out.
I'm on it.

Machine Translation

Summarization



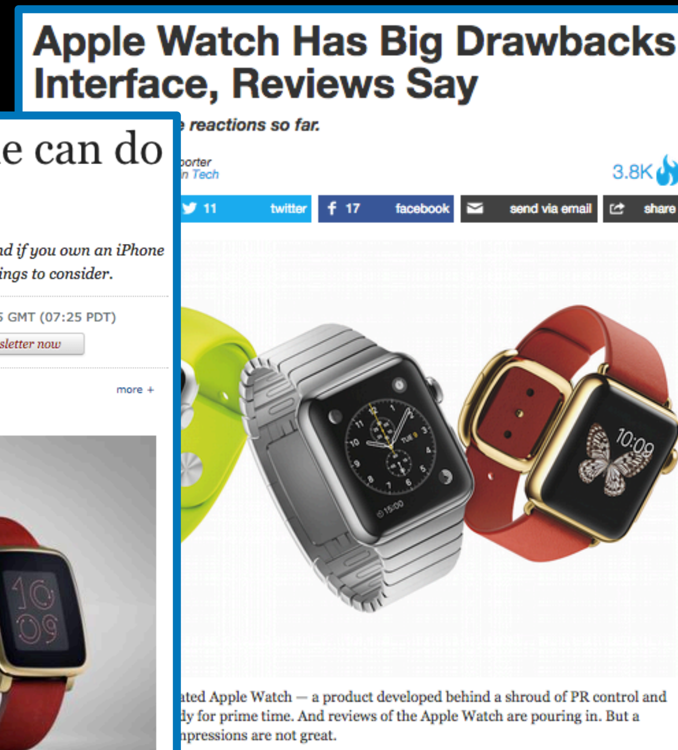

Five things the Pebble Time can do that the Apple Watch can't

Summary: The new Apple Watch isn't the only smartwatch to consider and if you own an iPhone then you should consider what the Pebble Time offers. Matthew lists five things to consider.

By Matthew Miller for The Mobile Gadgeteer | March 12, 2015 -- 14:25 GMT (07:25 PDT)

Follow @palmsolo 8,013 followers Get the ZDNet Microsoft newsletter now

Comments 5 Share on Facebook 1 Tweet 81 Share 6 more +


Apple Watch Has Big Drawbacks Interface, Reviews Say

reactions so far.

3.8K

11 twitter 17 facebook send via email share

ated Apple Watch — a product developed behind a shroud of PR control and ly for prime time. And reviews of the Apple Watch are pouring in. But a npressions are not great.

The Apple Watch has drawbacks. There are other smartwatches that offer more capabilities.

Question Answering

Question Answering

amazon alexa

"Alexa, who was President when Barack Obama was nine?"

"Alexa, how's my commute?

"Alexa, what's the weather?"

"Alexa, did the 49ers win?"

Dialog Systems

figure credit: Phani Marupaka

User-Facing Applications

Supporting Technologies

Language Understanding Capabilities

User-Facing Applications

Supporting Technologies

Language Understanding Capabilities

Word Sense Disambiguation

input	output
he's a **bass** in the choir .	$bass_3$
our **bass** is line-caught from the Atlantic .	$bass_4$

set of possible outputs = {$bass_1$, $bass_2$, …, $bass_8$}

- S: (n) **bass** (the lowest part of the musical range)
- S: (n) **bass**, bass part (the lowest part in polyphonic music)
- **S: (n) bass, basso (an adult male singer with the lowest voice)**
- S: (n) sea bass, **bass** (the lean flesh of a saltwater fish of the family Serranidae)
- S: (n) freshwater bass, **bass** (any of various North American freshwater fish with lean flesh (especially of the genus Micropterus))
- S: (n) **bass**, bass voice, basso (the lowest adult male singing voice)
- S: (n) **bass** (the member with the lowest range of a family of musical instruments)
- S: (n) **bass** (nontechnical name for any of numerous edible marine and freshwater spiny–finned fishes)

Part-of-Speech Tagging

Some questioned if Tim Cook ’s first product

would be a breakaway hit for Apple .

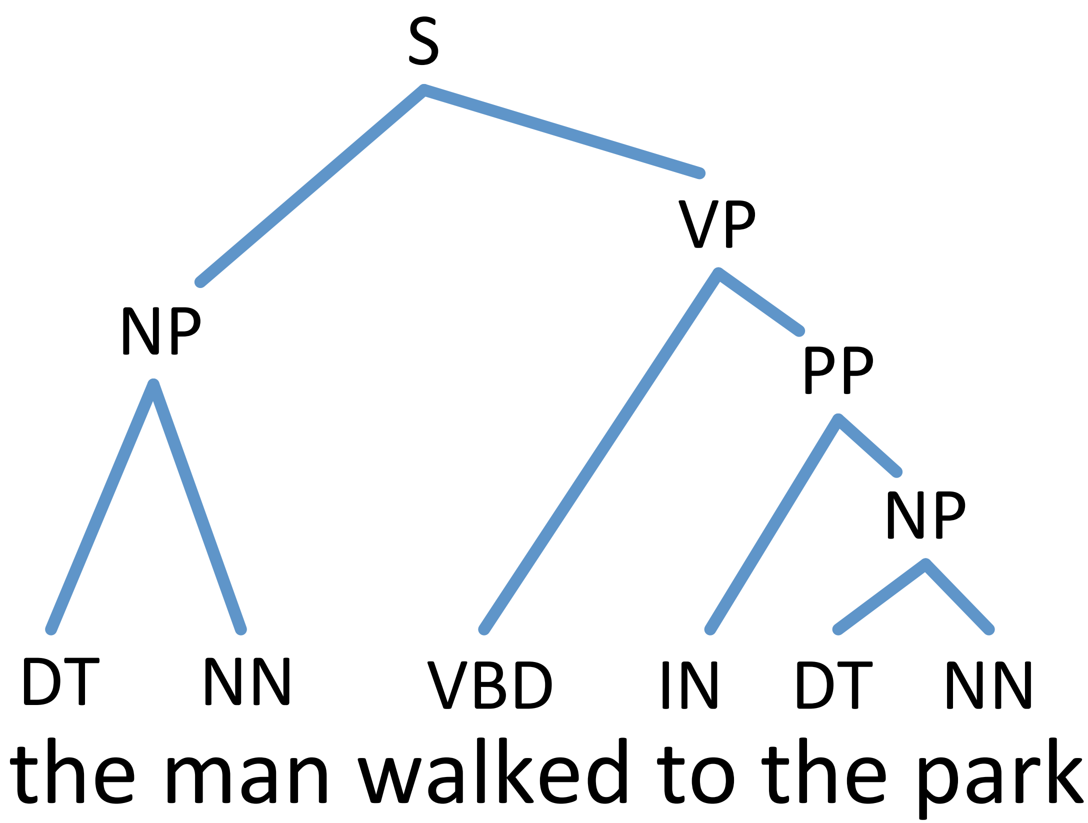
Part-of-Speech Tagging

determiner	verb (past)	prep.	proper noun	proper noun	poss.	adj.	noun
Some	questioned	if	Tim	Cook	’s	first	product

modal	verb	det.	adjective	noun	prep.	proper noun	punc.
would	be	a	breakaway	hit	for	Apple	.

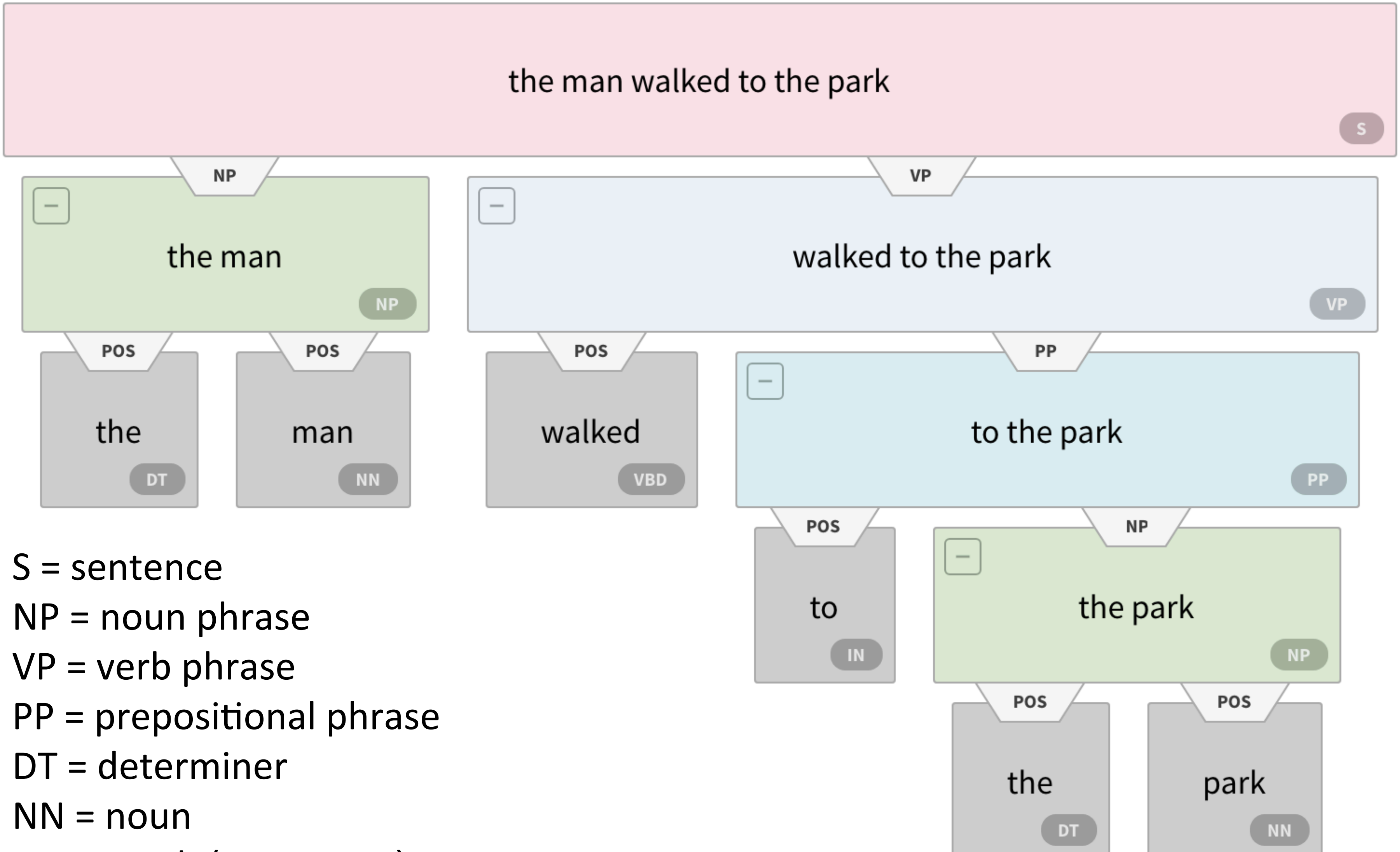
Constituency Parsing

(S (NP the man) (VP walked (PP to (NP the park))))



Key:
S = sentence
NP = noun phrase
VP = verb phrase
PP = prepositional phrase
DT = determiner
NN = noun
VBD = verb (past tense)
IN = preposition

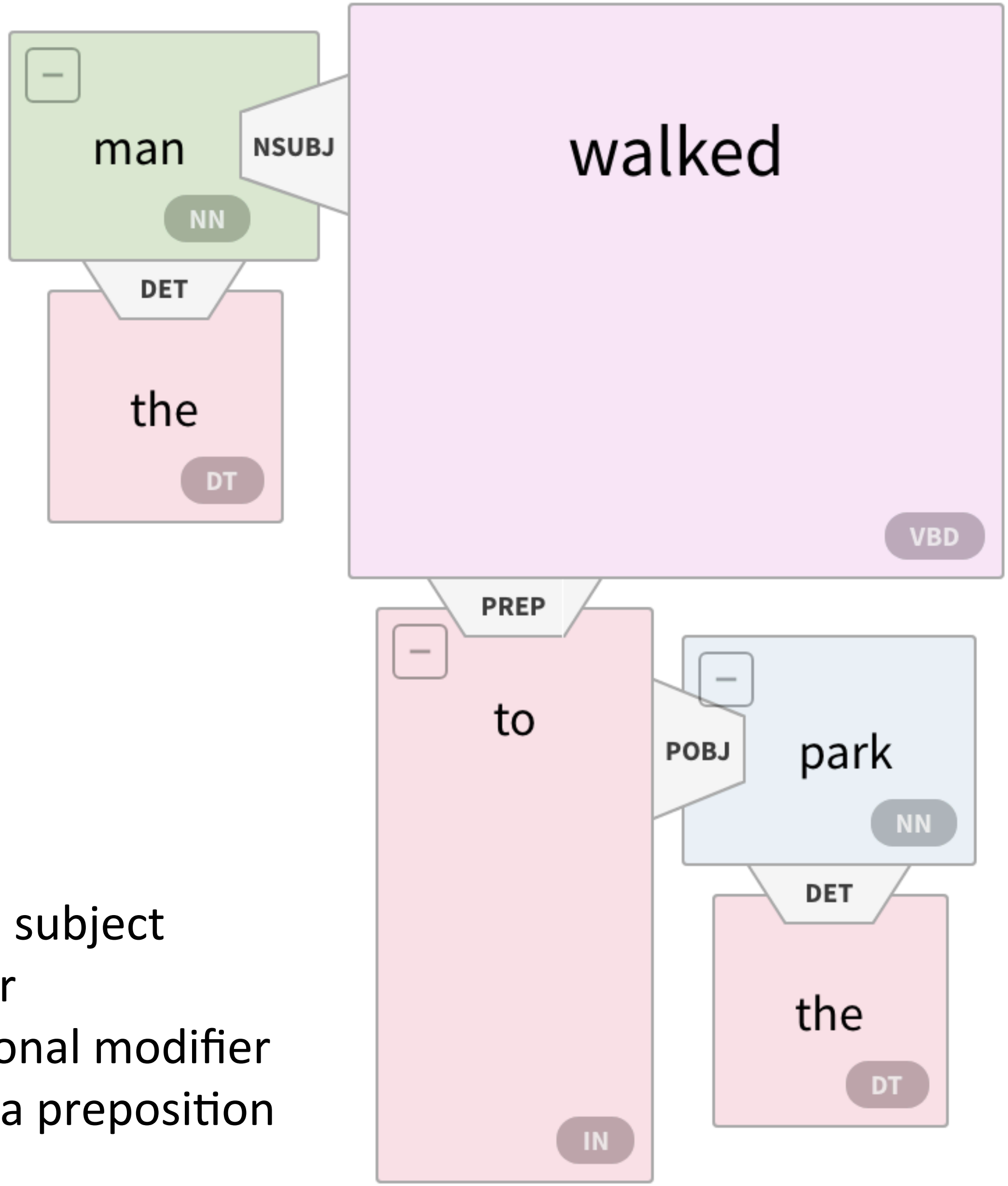
Constituency Parsing



S = sentence
NP = noun phrase
VP = verb phrase
PP = prepositional phrase
DT = determiner
NN = noun
VBD = verb (past tense)
IN = preposition

AllenNLP

Dependency Parsing



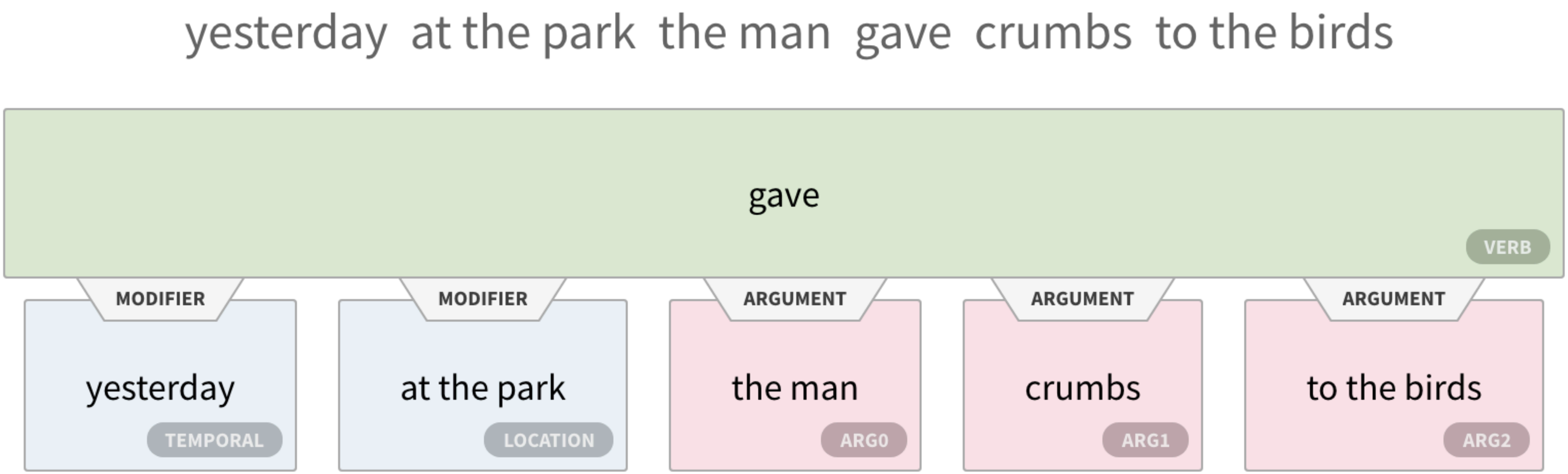
Key:
NSUBJ = nominal subject
DET = determiner
PREP = propositional modifier
POBJ = object of a preposition

AllenNLP

Semantic Parsing

- semantic role labeling (SRL)
- frame-semantic parsing
- semantic dependency formalisms
- abstract meaning representation (AMR)

Semantic Role Labeling



ARG0 = usually *agent*

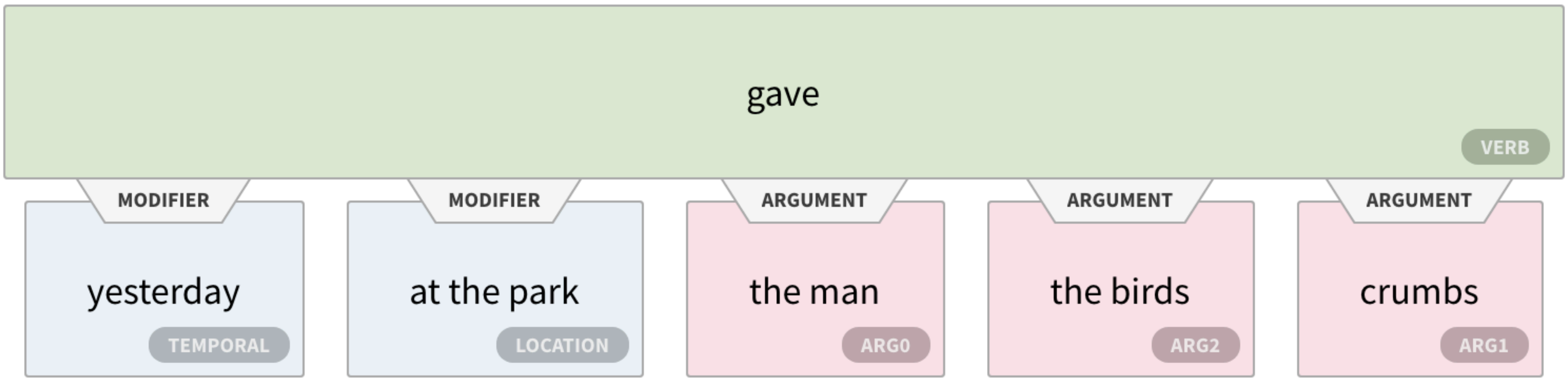
ARG1 = typically *patient* or *theme*

ARG2 = often *beneficiary*

Semantic Role Labeling

yesterday at the park the man gave crumbs to the birds

yesterday at the park the man gave the birds crumbs



ARG0 = usually *agent*

ARG1 = typically *patient* or *theme*

ARG2 = often *beneficiary*

Named Entity Recognition

Some questioned if Tim Cook's first product would be a breakaway hit for Apple.

PERSON **ORGANIZATION**

Entity Linking

Some questioned if Tim Cook's first product would be a breakaway hit for Apple.

Coreference Resolution

The boy threw some bread to a group of birds .
They fought over it as he watched .

2 The boy threw 1 some bread to 0 a group of birds .
0 They fought over 1 it as 2 he watched .

AllenNLP

User-Facing Applications

Supporting Technologies

Language Understanding Capabilities

User-Facing Applications

Supporting Technologies

Language Understanding Capabilities

Winograd Schema Coreference Resolution

The man couldn't lift his son because **he** was so weak.

The man couldn't lift his son because **he** was so heavy.

Winograd Schema Coreference Resolution

The man couldn't lift his son because **he** was so weak.

man

The man couldn't lift his son because **he** was so heavy.

son

Natural Language Inference

- A **entails** B:

  A: yesterday's game was canceled due to the rain.

  B: it rained yesterday.

- B **contradicts** A:

  A: yesterday's game was canceled due to the rain.

  B: it didn't rain yesterday.

- A and B are **neutral**:

  A: yesterday's game was canceled due to the rain.

  B: since I had the day free, I cleaned my basement.

Reading Comprehension Question Answering

Once there was a boy named Fritz who loved to draw. He drew everything. In the morning, he drew a picture of his cereal with milk. His papa said, “Don’t draw your cereal. Eat it!”

After school, Fritz drew a picture of his bicycle. His uncle said, “Don't draw your bicycle. Ride it!”

…

MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text

Reading Comprehension Question Answering

Once there was a boy named Fritz who loved to draw. He drew everything. In the morning, he drew a picture of his cereal with milk. His papa said, “Don’t draw your cereal. Eat it!”

After school, Fritz drew a picture of his bicycle. His uncle said, “Don't draw your bicycle. Ride it!”

…

What did Fritz draw first?

A) the toothpaste

B) his mama

C) cereal and milk

D) his bicycle

Reading Comprehension Question Answering

Once there was a boy named Fritz who loved to draw. He drew everything. In the morning, he drew a picture of his cereal with milk. His papa said, “Don’t draw your cereal. Eat it!”

After school, Fritz drew a picture of his bicycle. His uncle said, “Don't draw your bicycle. Ride it!”

…

What did Fritz draw first?

A) the toothpaste

B) his mama

C) cereal and milk

D) his bicycle

Reading Comprehension Question Answering

A Turing machine is a mathematical model of a general computing machine. It is a theoretical device that manipulates symbols contained on a strip of tape. Turing machines are not intended as a practical computing technology, but rather as a thought experiment representing a computing machine—anything from an advanced supercomputer to a mathematician with a pencil and paper. It is believed that if a problem can be solved by an algorithm, there exists a Turing machine that solves the problem. Indeed, this is the statement of the Church–Turing thesis. Furthermore, it is known that everything that can be computed on other models of computation known to us today, such as a RAM machine, Conway's Game of Life, cellular automata or any programming language can be computed on a Turing machine. Since Turing machines are easy to analyze mathematically, and are believed to be as powerful as any other model of computation, the Turing machine is the most commonly used model in complexity theory.

What is the term for a mathematical model that theoretically represents a general computing machine?
Ground Truth Answers: A Turing machine | A Turing machine | Turing machine
Prediction: A Turing machine

It is generally assumed that a Turing machine can solve anything capable of also being solved using what?
Ground Truth Answers: an algorithm | an algorithm | an algorithm
Prediction: RAM machine, Conway's Game of Life, cellular automata or any programming language

What is the most commonplace model utilized in complexity theory?
Ground Truth Answers: the Turing machine | the Turing machine | Turing machine
Prediction: Turing machine

What does a Turing machine handle on a strip of tape?
Ground Truth Answers: symbols | symbols | symbols
Prediction: general computing machine

SQuAD

The Stanford Question Answering Dataset

Sentence Similarity

Input	Output
Other ways are needed. We must find other ways.	4.4
Pakistan bomb victims’ families end protest Pakistan bomb victims to be buried after protest ends	2.6
I absolutely do believe there was an iceberg in those waters. I don't believe there was any iceberg at all anywhere near the Titanic.	1.2

Word Prediction

he bent down and searched the large container, trying to find anything else hidden in it other than the _____

Word Prediction

he turned to one of the cops beside him. “search the entire coffin.” the man nodded and bustled forward towards the coffin.

he bent down and searched the large container, trying to find anything else hidden in it other than the _____

Why is NLP hard?

ambiguity: one form can mean many things

variability: many forms can mean the same thing

Why is NLP hard?

ambiguity: one form can mean many things

variability: many forms can mean the same thing

many different kinds of variability and ambiguity

each NLP task must address distinct kinds

Example: Hyperlinks in Wikipedia

Wikipedia Articles

bar

```
bar (law)
bar (establishment)
bar association
bar (unit)
medal bar
bar (music)
```

…

…

…

Example: Hyperlinks in Wikipedia

Wikipedia Articles

bar

`bar (law)`

`bar (establishment)`

`bar association`

`bar (unit)`

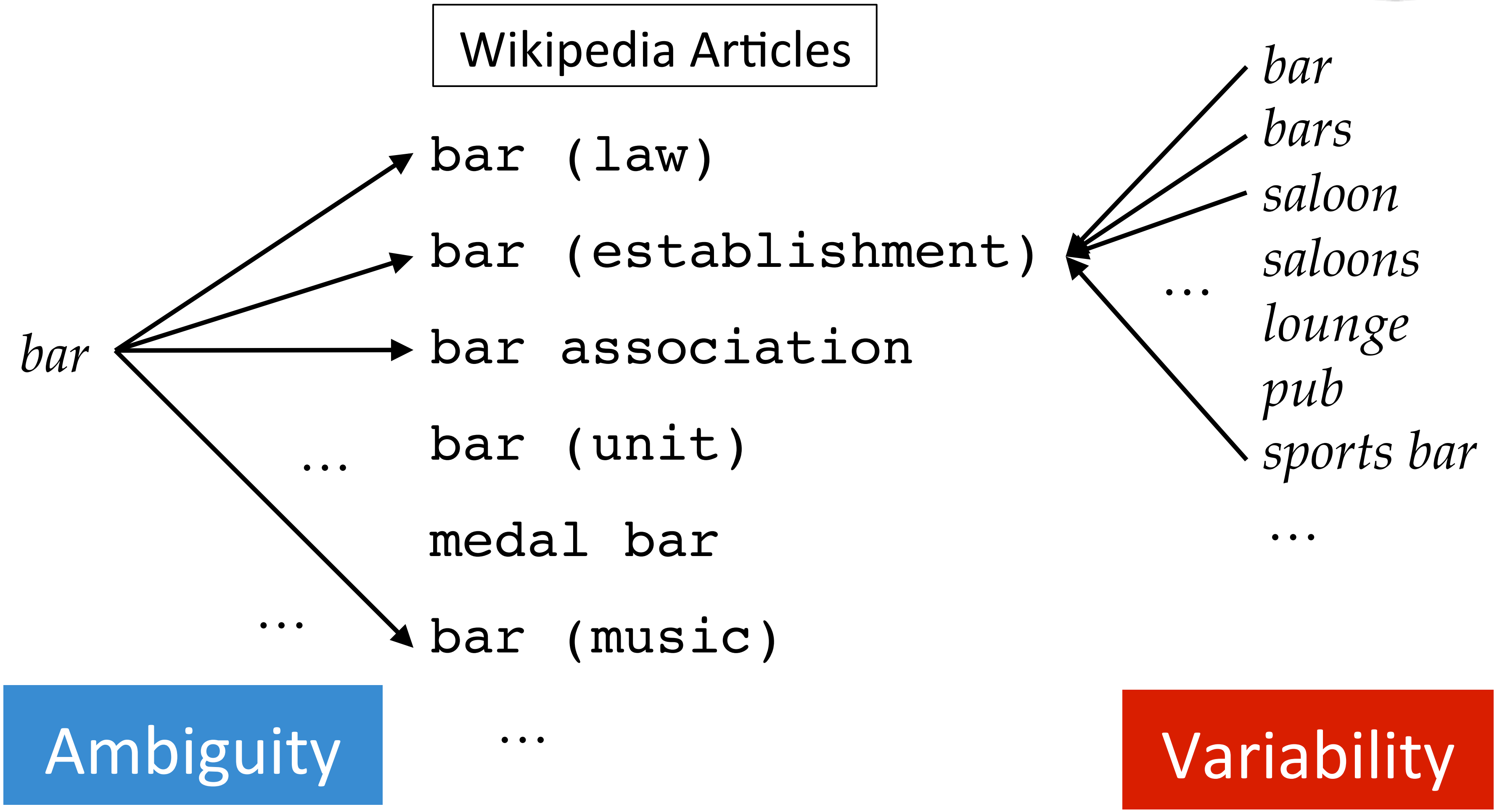
`medal bar`

`bar (music)`

…

Ambiguity

Example: Hyperlinks in Wikipedia



What is a classifier?

- a function from inputs ***x*** to outputs ***y***
- one simple type of classifier:
  - for any input ***x***, assign a score to each output ***y***, parameterized by parameters ***w***:

$$\text{score}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$$

  - classify by choosing highest-scoring output:

$$\text{classify}(\boldsymbol{x}, \boldsymbol{w}) = \underset{\boldsymbol{y}}{\text{argmax}} \ \text{score}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$$

Notation

$$\mathbf{u} = \text{a vector}$$

$$u_i = \text{entry } i \text{ in the vector}$$

$$\mathbf{W} = \text{a matrix}$$

$$w_{ij} = \text{entry } (i,j) \text{ in the matrix}$$

$$\boxed{\boldsymbol{x} = \text{a structured object}}$$

$$x_i = \text{item } i \text{ in the structured object}$$

Modeling, Inference, Learning

inference: solve argmax

modeling: define score function

$$\mathrm{classify}(\boldsymbol{x}, \boldsymbol{w}) = \underset{\boldsymbol{y}}{\mathrm{argmax}} \ \mathrm{score}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$$

learning: choose $\boldsymbol{w}$

Applications of our Classifier Framework

task	input (x)	output (y)	output space ($\mathcal{L}$)	size of $\mathcal{L}$
text classification	a sentence	gold standard label for $\boldsymbol{x}$	pre-defined, small label set (e.g., {positive, negative})	2-10
word sense disambiguation	instance of a particular word (e.g., *bass*) with its context	gold standard word sense of target word	pre-defined sense inventory from WordNet for *bass*	2-30
learning skip-gram word embeddings	instance of a word in a corpus	a word in the context of x in a corpus	vocabulary	$\lvert V \rvert$
part-of-speech tagging	a sentence	gold standard part-of-speech tags for $\boldsymbol{x}$	all possible part-of-speech tag sequences with same length as $\boldsymbol{x}$	$\lvert P \rvert^{\lvert \boldsymbol{x} \rvert}$

Applications of our Classifier Framework

task	input (x)	output (y)	output space ($\mathcal{L}$)	size of $\mathcal{L}$
text classification	a sentence	gold standard label for x	pre-defined, small label set (e.g., {positive, negative})	2-10
word sense disambiguation	instance of a particular word (e.g., *bass* … its cont…	gold standard word sense of …	pre-defined sense inventory from …	2-30
learning skip-gram word embeddings	instance … word in a c…	… a corpus		
part-of-speech tagging	a sentence	gold standard part-of-speech tags for x	all possible part-of-speech tag sequences with same length as x	$\|P\|^{\|x\|}$

exponential in size of input!
"structured prediction"

Applications of Classifier Framework (continued)

task	input (x)	output (y)	output space ($\mathcal{L}$)	size of $\mathcal{L}$
named entity recognition	a sentence	gold standard named entity labels for $\boldsymbol{x}$ (BIO tags)	all possible BIO label sequences with same length as $\boldsymbol{x}$	$\lvert P \rvert^{\lvert \boldsymbol{x} \rvert}$
constituency parsing	a sentence	gold standard constituent parse (labeled bracketing) of $\boldsymbol{x}$	all possible labeled bracketings of $\boldsymbol{x}$	exponential in length of $\boldsymbol{x}$ (Catalan number)
dependency parsing	a sentence	gold standard dependency parse (labeled directed spanning tree) of $\boldsymbol{x}$	all possible labeled directed spanning trees of $\boldsymbol{x}$	exponential in length of $\boldsymbol{x}$
machine translation	a sentence	a translation of $\boldsymbol{x}$	all possible translations of $\boldsymbol{x}$	potentially infinite

Modeling

modeling: define score function

$$\text{classify}(\boldsymbol{x}, \boldsymbol{w}) = \operatorname*{argmax}_{\boldsymbol{y}} \ \text{score}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$$

Linear Models

- parameters are arranged in a vector $\mathbf{w}$
- score function is linear in $\mathbf{w}$:

$$\text{score}(\boldsymbol{x}, \boldsymbol{y}, \mathbf{w}) = \mathbf{w}^\top \mathbf{f}(\boldsymbol{x}, \boldsymbol{y}) = \sum_i w_i f_i(\boldsymbol{x}, \boldsymbol{y})$$

- $\mathbf{f}$: vector of feature functions
- each feature function can look at the entire input and output

Notation

$$\mathbf{u} = \text{a vector}$$

$$u_i = \text{entry } i \text{ in the vector}$$

$$\mathbf{W} = \text{a matrix}$$

$$w_{ij} = \text{entry } (i,j) \text{ in the matrix}$$

$$\boldsymbol{x} = \text{a structured object}$$

$$x_i = \text{item } i \text{ in the structured object}$$

Stochastic/Generative Models

model	tasks	context expansion
n-gram language models	language modeling (for MT, ASR, etc.)	increase *n*
hidden Markov models	part-of-speech tagging, named entity recognition, word clustering	increase order of HMM (e.g., bigram HMM → trigram HMM)
probabilistic context-free grammars	constituency parsing	increase size of rules, e.g., flattening, parent annotation, etc.

- all use maximum likelihood estimation + smoothing (though different kinds)
- form of features dependent on "generative story"
- all assign probabilities to sentences (or to pairs of <sentence, something else>)
- trade-off between increasing "context" and needing more data / better smoothing

Model Families

- linear models
  - lots of freedom in defining features, though feature engineering required for best performance
  - learning uses optimization of a loss function
  - one can (try to) interpret learned feature weights
- stochastic/generative models
  - linear models with simple “features” (counts of events)
  - learning is easy: count & normalize (but smoothing needed)
  - easy to generate samples
- neural models
  - less feature engineering required (“features” are learned)
  - learning uses optimization of a loss function
  - works well; hard to interpret

Inference

inference: solve argmax

$$\text{classify}(\boldsymbol{x}, \boldsymbol{w}) = \operatorname*{argmax}_{\boldsymbol{y}} \ \text{score}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$$

Inference for Structured Prediction

$$\text{classify}(\boldsymbol{x}, \boldsymbol{w}) = \underset{\boldsymbol{y}}{\text{argmax}} \ \ \text{score}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$$

- how do we efficiently search over the space of all structured outputs?

- this space may have size exponential in the size of the input, or be unbounded

Feature Locality

- **feature locality**: how "big" are your features?
- we need to be mindful of this to enable efficient inference
- features can be arbitrarily big in terms of the *input*
- but features **cannot** be arbitrarily big in terms of the *output*!

Inference in HMMs

$$\text{classify}(\boldsymbol{x}, \boldsymbol{w}) = \operatorname*{argmax}_{\boldsymbol{y}} \; p_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y})$$

- since the output is a sequence, this argmax requires iterating over an exponentially-large set
- we can use **dynamic programming (DP)** to solve these problems exactly
- for HMMs (and other sequence models), the algorithm for solving this is the **Viterbi algorithm**

Learning

$$\text{classify}(\boldsymbol{x}, \boldsymbol{w}) = \underset{\boldsymbol{y}}{\text{argmax}} \;\; \text{score}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w})$$

learning: choose $\boldsymbol{w}$

Cost Functions

- cost function: scores outputs against a gold standard

$$\text{cost} : \mathcal{L} \times \mathcal{L} \to \mathbb{R}_{\geq 0}$$

- should be as close as possible to the actual evaluation metric for your task

- typical cost for multi-class classification:

$$\text{cost}(y, y') = \mathbb{I}[y \neq y']$$

Empirical Risk Minimization
(Vapnik et al.)

- replace expectation with sum over examples:

$$\hat{\boldsymbol{w}} = \operatorname*{argmin}_{\boldsymbol{w}} \ \mathbb{E}_{P(\boldsymbol{x},y)}\left[\text{cost}(y, \text{classify}(\boldsymbol{x}, \boldsymbol{w}))\right]$$

$$\downarrow$$

$$\hat{\boldsymbol{w}} = \operatorname*{argmin}_{\boldsymbol{w}} \ \sum_{i=1}^{|\mathcal{T}|} \text{cost}(y^{(i)}, \text{classify}(\boldsymbol{x}^{(i)}, \boldsymbol{w}))$$

solution: replace "cost loss" (also called "0-1" loss) with a surrogate function that is easier to optimize

$$\hat{\boldsymbol{w}} = \operatorname*{argmin}_{\boldsymbol{w}} \sum_{i=1}^{|\mathcal{T}|} \text{cost}(y^{(i)}, \text{classify}(\boldsymbol{x}^{(i)}, \boldsymbol{w}))$$

generalize to permit any loss function

$$\hat{\boldsymbol{w}} = \operatorname*{argmin}_{\boldsymbol{w}} \sum_{i=1}^{|\mathcal{T}|} \text{loss}(\boldsymbol{x}^{(i)}, y^{(i)}, \boldsymbol{w})$$

cost loss / 0-1 loss: $\text{loss}_{\text{cost}}(\boldsymbol{x}, y, \boldsymbol{w}) = \text{cost}(y, \text{classify}(\boldsymbol{x}, \boldsymbol{w}))$

Surrogate Loss Functions

cost loss / 0-1 loss: $\text{loss}_{\text{cost}}(\boldsymbol{x}, y, \boldsymbol{w}) = \text{cost}(y, \text{classify}(\boldsymbol{x}, \boldsymbol{w}))$

perceptron loss:

$$\text{loss}_{\text{perc}}(\boldsymbol{x}, y, \boldsymbol{w}) = -\text{score}(\boldsymbol{x}, y, \boldsymbol{w}) + \max_{y' \in \mathcal{L}} \ \text{score}(\boldsymbol{x}, y', \boldsymbol{w})$$

hinge loss:

$$\text{loss}_{\text{hinge}}(\boldsymbol{x}, y, \boldsymbol{w}) = -\text{score}(\boldsymbol{x}, y, \boldsymbol{w}) + \max_{y' \in \mathcal{L}} (\text{score}(\boldsymbol{x}, y', \boldsymbol{w}) + \text{cost}(y, y'))$$

log loss:

$$\text{loss}_{\text{log}}(\boldsymbol{x}, y, \boldsymbol{w}) = -\log p_{\boldsymbol{w}}(y \mid \boldsymbol{x})$$

Score → Probability

- can turn score into probability by exponentiating (to make it positive) and normalizing:

$$p_{\boldsymbol{w}}(y \mid \boldsymbol{x}) \propto \exp\{\text{score}(\boldsymbol{x}, y, \boldsymbol{w})\}$$

$$p_{\boldsymbol{w}}(y \mid \boldsymbol{x}) = \frac{\exp\{\text{score}(\boldsymbol{x}, y, \boldsymbol{w})\}}{\sum_{y' \in \mathcal{L}} \exp\{\text{score}(\boldsymbol{x}, y', \boldsymbol{w})\}}$$

- this is often called a “softmax” function