TTIC 31190: Natural Language Processing

Kevin Gimpel

Winter 2016

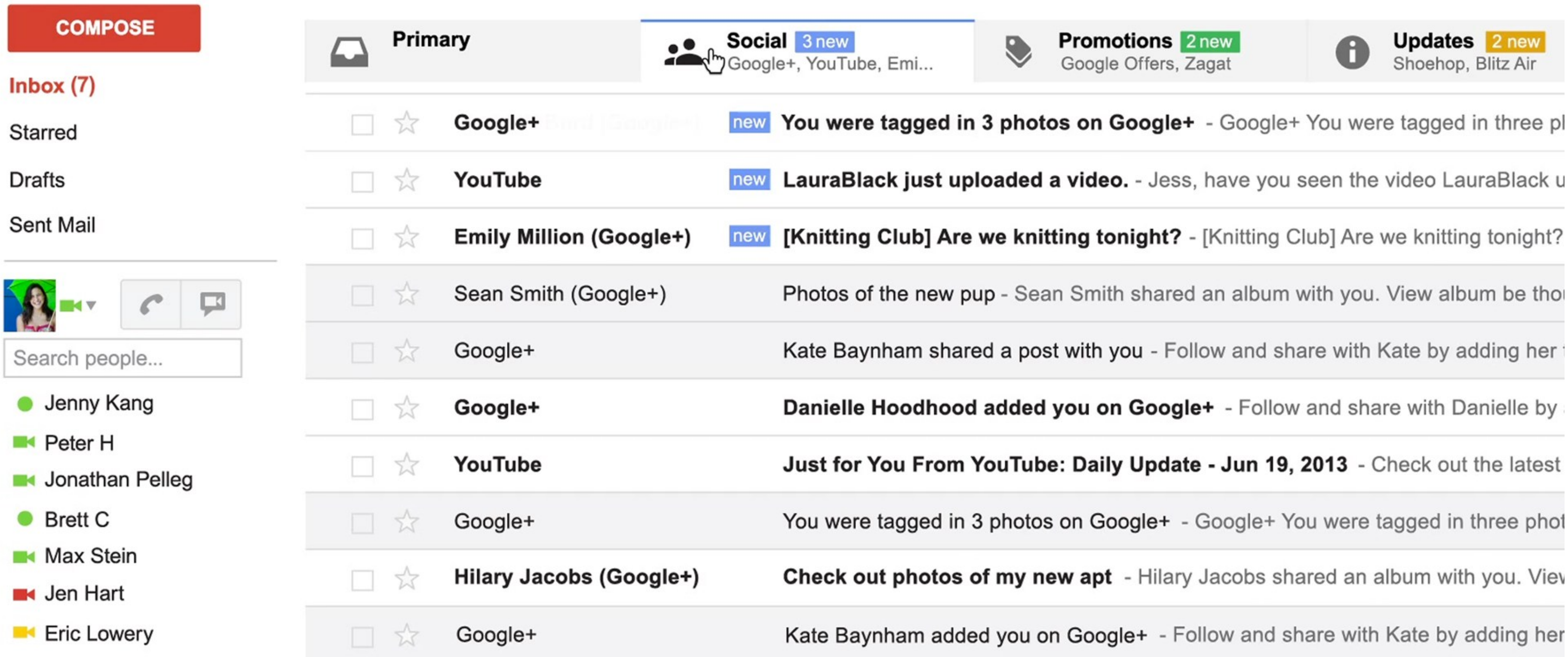
Lecture 1: Introduction

What is natural language processing?

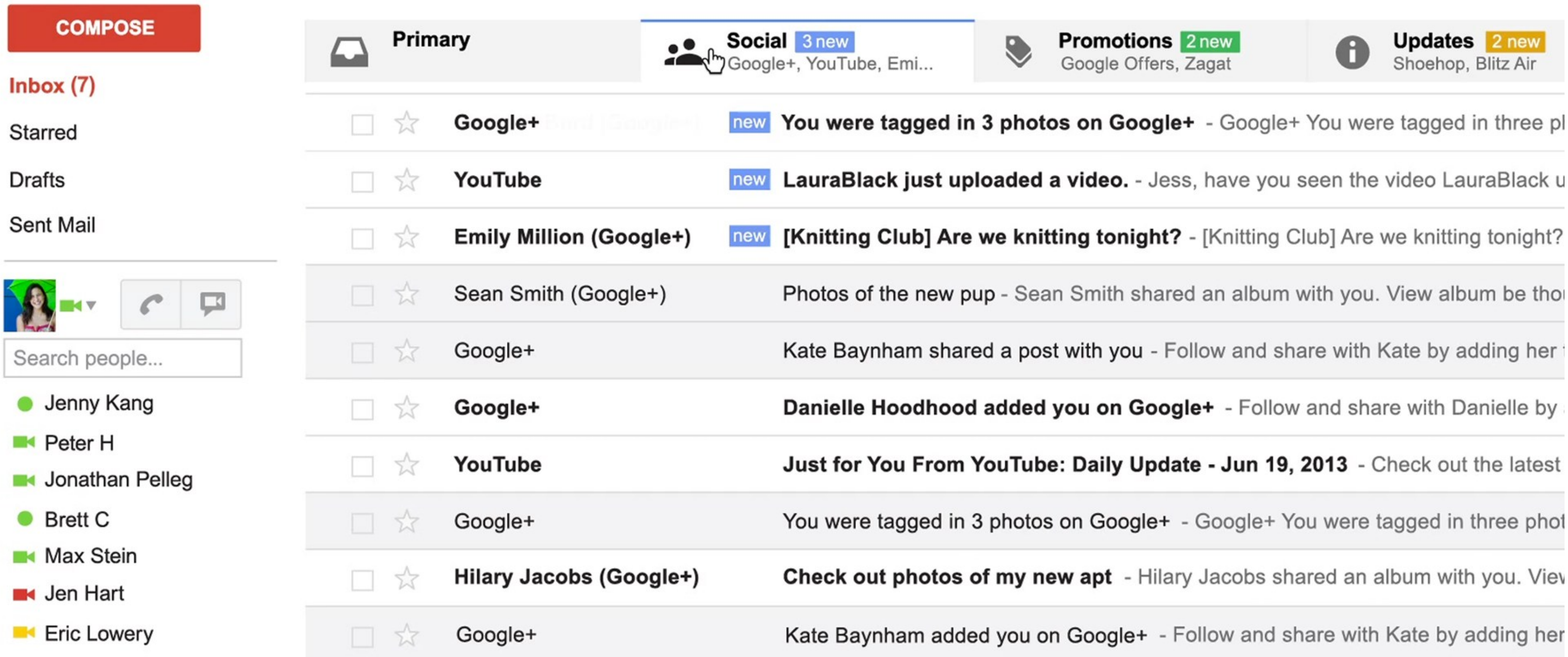
What is natural language processing?

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

Text Classification

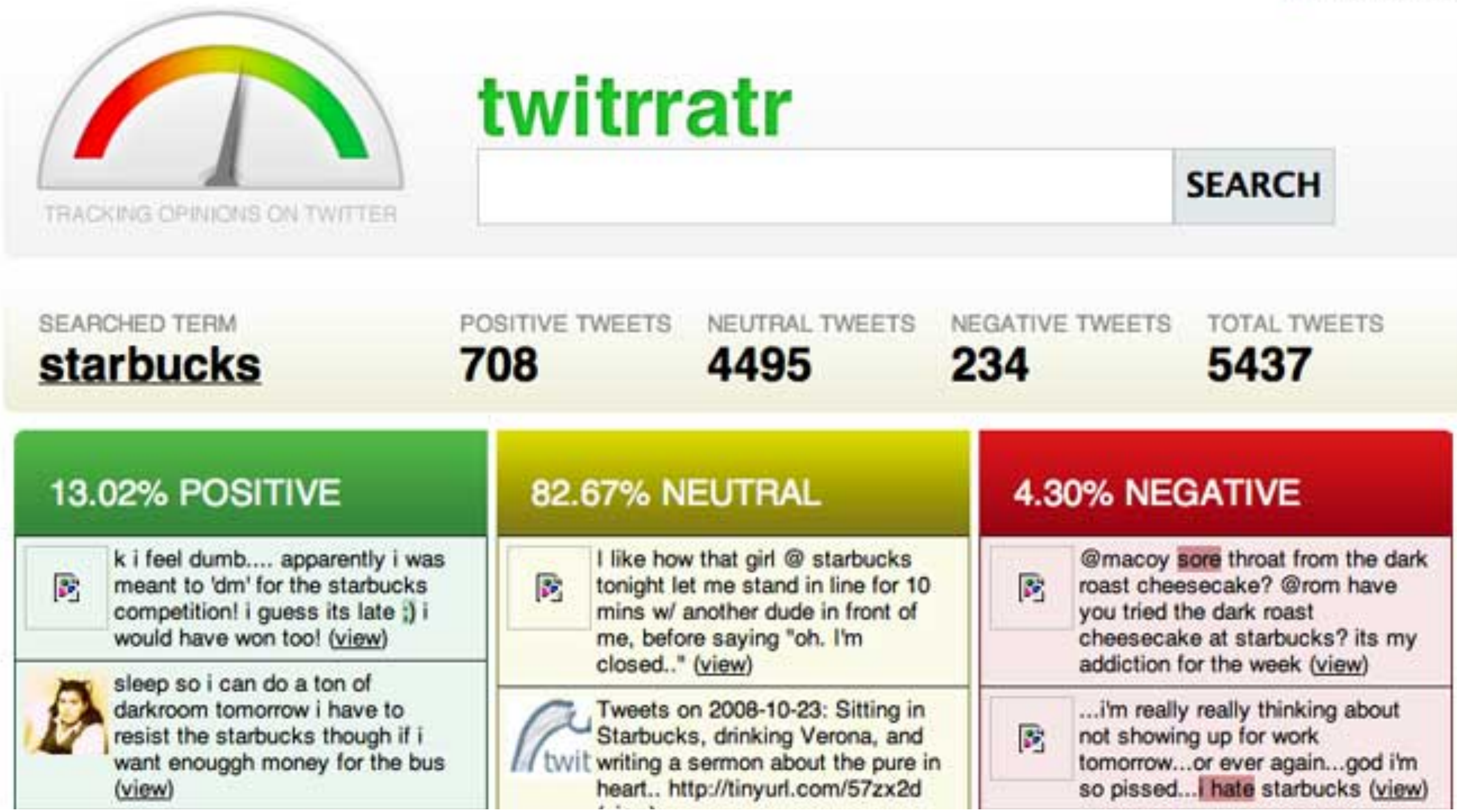


Text Classification



- spam / not spam
- priority level
- category (primary / social / promotions / updates)

Sentiment Analysis



Machine Translation

14:11 Uhr · Apple Watch · *fen*

Neue Umfrage: Kaufen Sie eine Apple Watch?

Seit gestern ist auch die genaue Preisstruktur der Apple Watch bekannt und viele Nutzer **befassen sich daher mit der Frage**, ob sie eine Apple Watch kaufen werden oder ob das Produkt nicht dem eigenen Geschmack entspricht. In unserer neuen Umfrage möchten wir gerne von Ihnen wissen, ob Sie schon eine Entscheidung getroffen haben - wird Ihre nächste Uhr eine Apple Watch und welches der drei Grundmodelle soll es dann sein? Oder hat Apple keine Chance, Sie als Käufer begrüßen zu können? Eine detaillierte Preisübersicht hatten wir in diesem Artikel zusammengestellt: @

Machine Translation

14:11 Uhr · Apple Watch · *fen*

Neue Umfrage: Kaufen Sie eine Apple Watch?

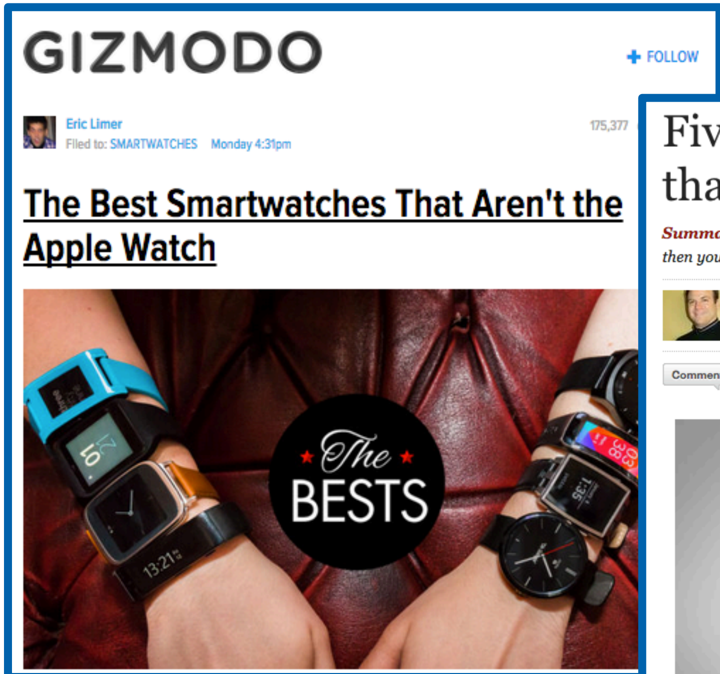
Seit gestern ist auch die genaue reisstruktur der Apple Watch bekannt und viele Nutzer **befassen sic her mit der Frage**, ob sie eine Apple

New Poll: Will you buy an Apple Watch?

von Ihnen wissen, ob Sie schon eine Entscheidung getroffen haben - wird Ihre nächste Uhr eine Apple Watch und welches der drei Grundmodelle soll es dann sein? Oder hat Apple keine Chance, Sie als Käufer begrüßen zu können? Eine detaillierte Preisübersicht hatten wir in diesem Artikel zusammengestellt: @

Question Answering

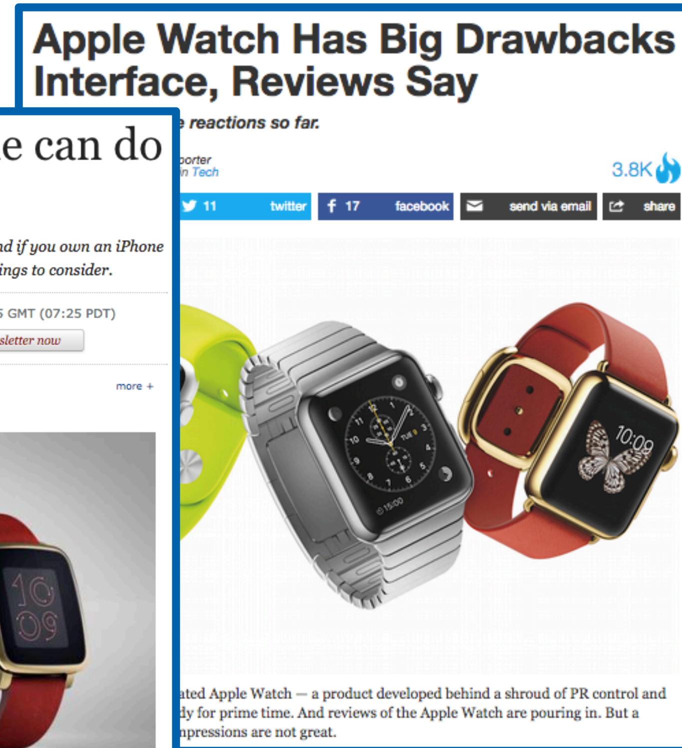
Summarization


GIZMODO
+ FOLLOW
Eric Limer
Filed to: SMARTWATCHES Monday 4:31pm
175,377
The Best Smartwatches That Aren't the Apple Watch


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 +

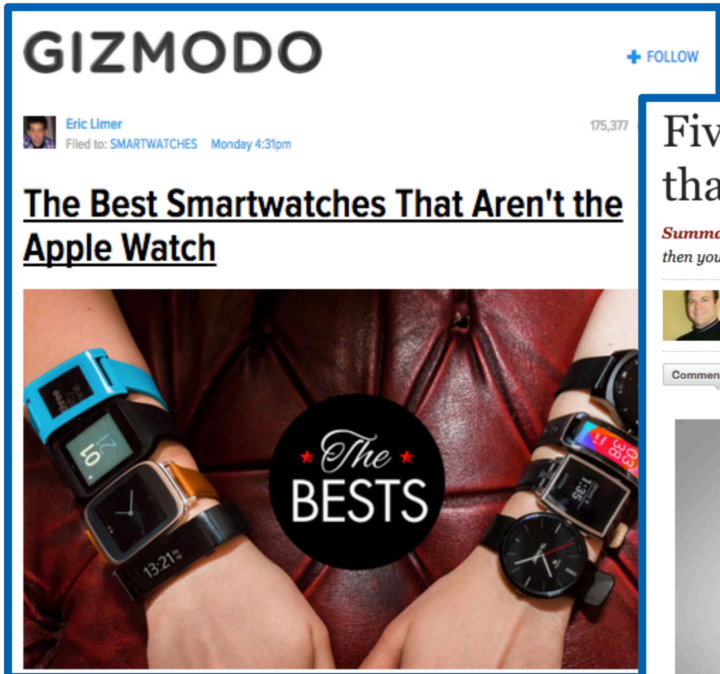

Apple Watch Has Big Drawbacks Interface, Reviews Say

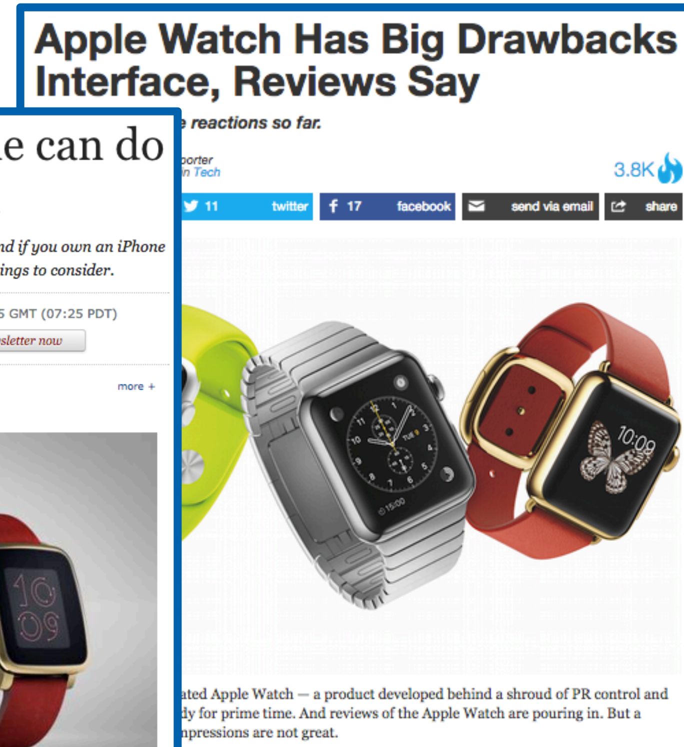
e reactions so far.

porter
in Tech
3.8K
11 twitter 17 facebook send via email share

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

Summarization





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

Dialog Systems

user: Schedule a meeting with Matt and David on Thursday.
computer: Thursday won't work for David. How about Friday?
user: I'd prefer Monday then, but Friday would be ok if necessary.

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	.

Part-of-Speech Tagging

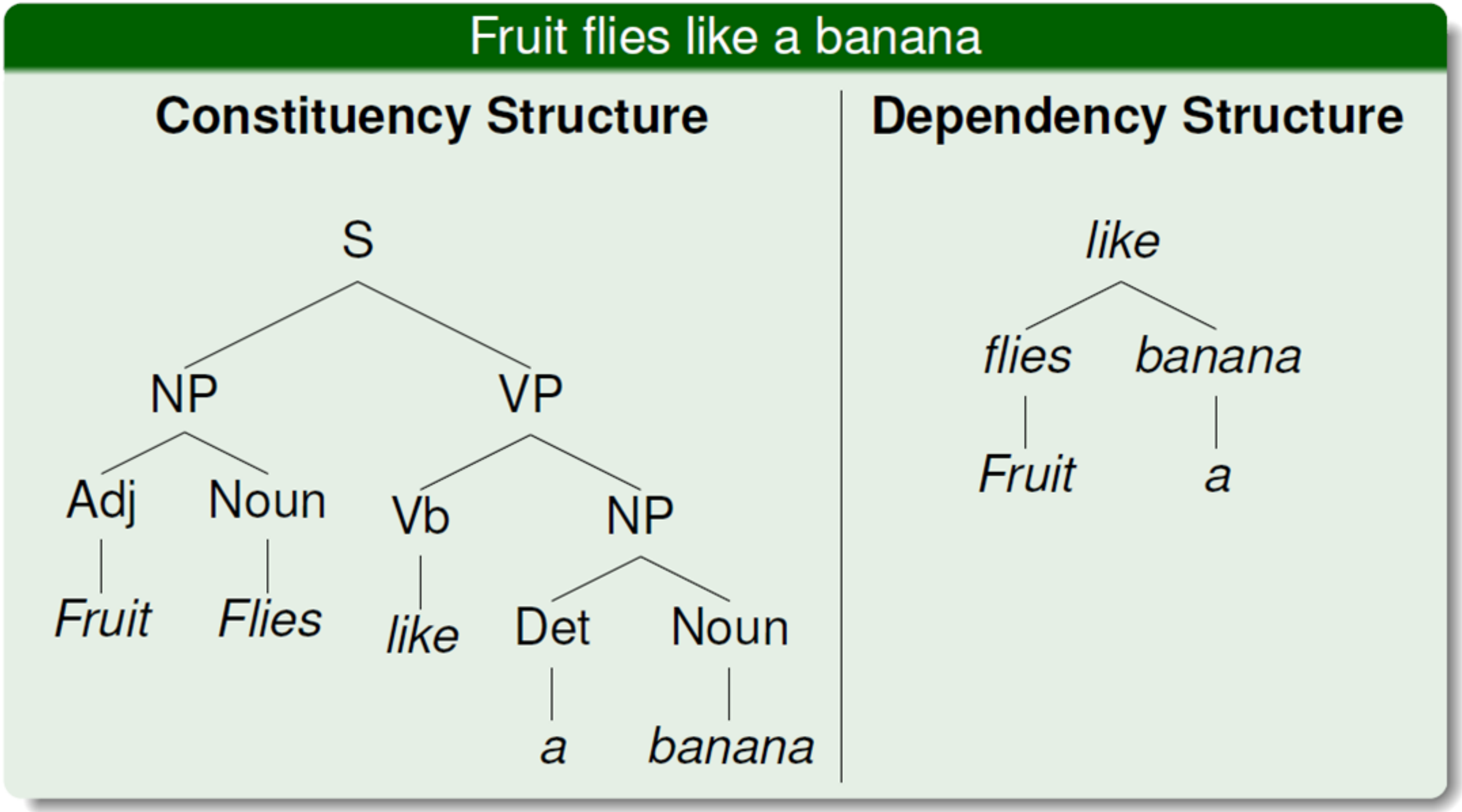
determiner	verb (past)	prep.	proper noun	proper noun	poss.	adj.	noun
Some	questioned	if	Tim	Cook	’s	first	product

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

Named Entity Recognition

Some questioned if $\underbrace{\text{Tim Cook}}_{\textbf{PERSON}}$’s first product would be a breakaway hit for $\underbrace{\text{Apple}}_{\textbf{ORGANIZATION}}$.

Syntactic Parsing



Entity Linking

```
en.wikipedia.org/wiki/Dell
Infobox type: company
```

```
en.wikipedia.org/wiki/Michael_Dell
Infobox type: person
```

Revenues of $14.5 billion were posted by <u>*Dell*</u>$_1$. <u>*The company*</u>$_1$ ...

Coreference Resolution

figure credit: Durrett & Klein (2014)

“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

Reading Comprehension

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

Conspicuous by their absence…

- speech recognition (see TTIC 31110)
- information retrieval and web search
- knowledge representation
- recommender systems

Computational Linguistics vs. Natural Language Processing

- how do they differ?

Computational Linguistics

This webpage contains a link to my lecture notes for Winter 2013.

Click here for lecture notes.

Computer Science CMSC 25020-1 and CMSC 35030-1

Winter 2013

John Goldsmith goldsmith@uchicago.edu. Office in CS: Ryerson 258. Also in Walker 201.

About this course

This is a course in the Computer Science department, intended for upper-level undergraduates, or graduate students, who have a good programming background. In general, we face the same kind of negotiation over choice of language that you might expect. If you want to submit code in C++, perl, or Python, that should be no problem; other choices are discussable, and the decision will have to be made by the instructor and the TA jointly.

Computational Biology vs. Bioinformatics

*“**Computational biology** = the study of biology using computational techniques. The goal is to learn new biology, knowledge about living systems. It is about science.*

***Bioinformatics** = the creation of tools (algorithms, databases) that solve problems. The goal is to build useful tools that work on biological data. It is about engineering.”*

--Russ Altman

Computational Linguistics vs. Natural Language Processing

- many people think of the two terms as synonyms
- computational linguistics is more inclusive; more likely to include sociolinguistics, cognitive linguistics, and computational social science
- NLP is more likely to use machine learning and involve engineering / system-building

Is NLP Science or Engineering?

- goal of NLP is to develop technology, which takes the form of engineering
- though we try to solve today's problems, we seek principles that will be useful for the future
- if science, it's not linguistics or cognitive science; it's the science of computational processing of language
- so I like to think that we're doing the science of engineering

Course Overview

- New course, first time being offered
- Aimed at first-year PhD students
- Instructor office hours: Mondays 3-4 pm, TTIC 531
- Teaching assistant: Lifu Tu, TTIC PhD student

Prerequisites

- No course prerequisites, but I will assume:
  - some programming experience (no specific language required)
  - familiarity with basics of probability, calculus, and linear algebra
- Undergraduates with relevant background are welcome to take the course. Please bring an enrollment approval form to me if you can't enroll online.

Grading

- 3 assignments (15% each)
- midterm exam (15%)
- course project (35%):
  - preliminary report and meeting with instructor (10%)
  - class presentation (5%)
  - final report (20%)
- class participation (5%)
- no final

Assignments

- Mixture of formal exercises, implementation, experimentation, analysis
- "Choose your own adventure" component based on your interests, e.g.:
  - exploratory data analysis
  - machine learning
  - implementation/scalability
  - model and error analysis
  - visualization

Project

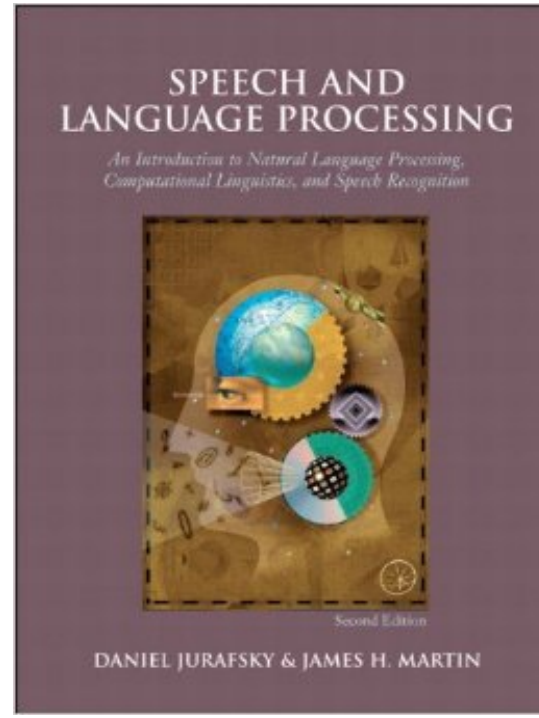
- Replicate [part of] a published NLP paper, or define your own project.
- The project may be done individually or in a group of two. Each group member will receive the same grade.
- More details to come.

Collaboration Policy

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

Textbooks

- All are optional
- Speech and Language Processing, 2nd Ed.
  - some chapters of 3rd edition are online
- The Analysis of Data, Volume 1: Probability
  - freely available online
- Introduction to Information Retrieval
  - freely available online



Roadmap

- classification
- words
- lexical semantics
- language modeling
- sequence labeling
- syntax and syntactic parsing
- neural network methods in NLP
- semantic compositionality
- semantic parsing
- unsupervised learning
- machine translation and other applications

Why is NLP hard?

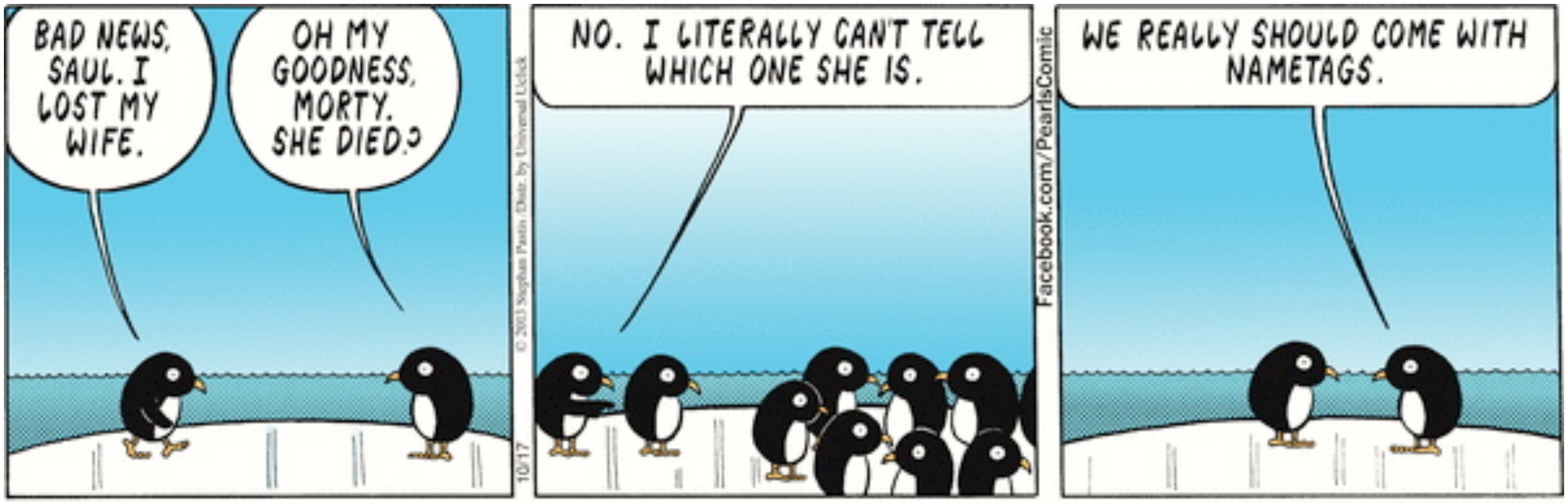
- ambiguity and variability of linguistic expression:
  - **variability**: many forms can mean the same thing
  - **ambiguity**: one form can mean many things

- there are many different kinds of ambiguity
- each NLP task has to address a distinct set of kinds

Word Sense Ambiguity

- many words have multiple meanings

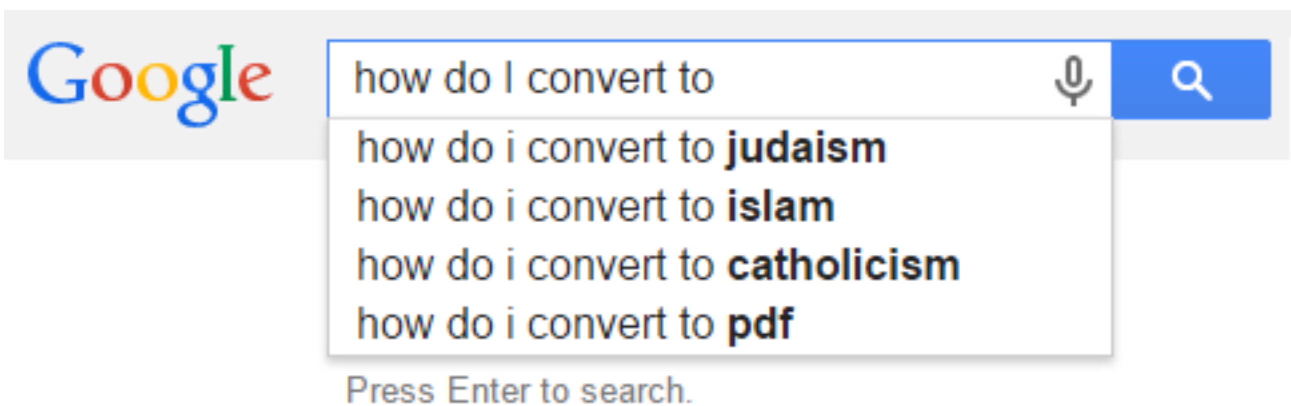
Word Sense Ambiguity



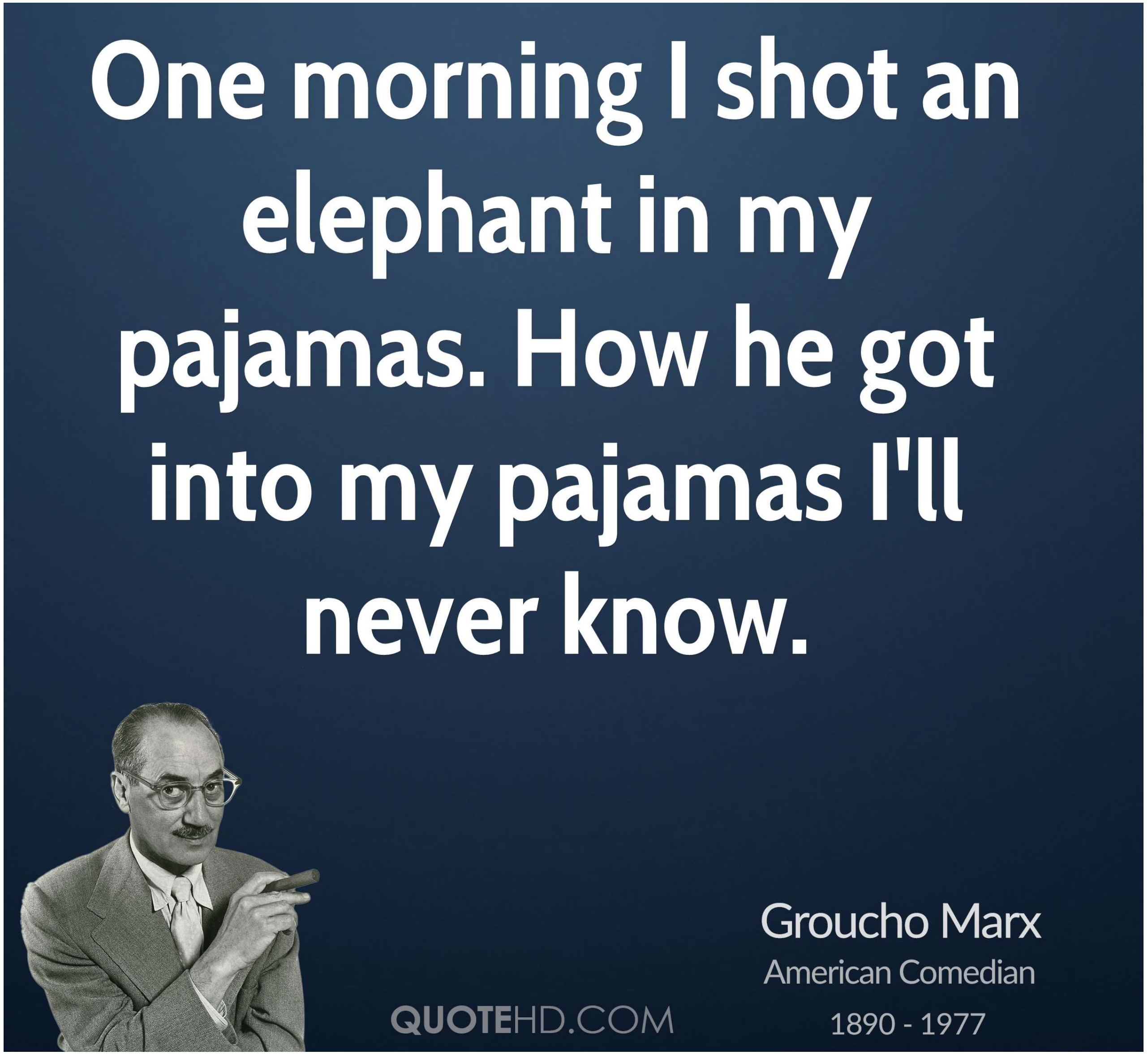
credit: A. Zwicky

Word Sense Ambiguity

credit: A. Zwicky



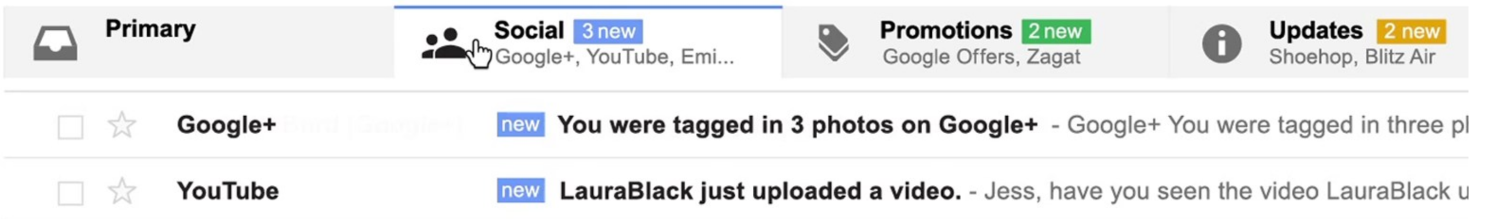
Attachment Ambiguity



Meaning Ambiguity

Text Classification

- simplest user-facing NLP application
- email (spam, priority, categories):



- sentiment:

- topic classification
- others?

What is a classifier?

What is a classifier?

- a function from inputs x to classification labels y

What is a classifier?

- a function from inputs *x* to classification labels *y*
- one simple type of classifier:
  - for any input *x*, assign a score to each label *y*, parameterized by vector θ:

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

What is a classifier?

- a function from inputs *x* to classification labels *y*
- one simple type of classifier:
  - for any input *x*, assign a score to each label *y*, parameterized by vector θ:

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

  - classify by choosing highest-scoring label:

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

Course Philosophy

- From reading papers, one gets the idea that machine learning concepts are monolithic, opaque objects
  - e.g., naïve Bayes, logistic regression, SVMs, CRFs, neural networks, LSTMs, etc.
- Nothing is opaque
- Everything can be dissected, which reveals connections
- The names above are useful shorthand, but not useful for gaining understanding

Course Philosophy

- We will draw from machine learning, linguistics, and algorithms, but technical material will be (mostly) self-contained; we won't use many black boxes

- We will focus on declarative (rather than procedural) specifications, because they highlight connections and differences

Modeling, Inference, Learning

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

Modeling, Inference, Learning

modeling: define score function

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

- **Modeling**: How do we assign a score to an (x,y) pair using parameters θ?

Modeling, Inference, Learning

inference: solve argmax

modeling: define score function

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

- **Inference**: How do we efficiently search over the space of all labels?

Modeling, Inference, Learning

inference: solve argmax

modeling: define score function

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

learning: choose $\boldsymbol{\theta}$

- **Learning**: How do we choose $\boldsymbol{\theta}$?

Modeling, Inference, Learning

inference: solve argmax

modeling: define score function

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

learning: choose $\boldsymbol{\theta}$

- **We will use this same paradigm throughout the course, even when the output space size is exponential in the size of the input or is unbounded (e.g., machine translation)**

Notation

- We'll use boldface for vectors:

$$\boldsymbol{\theta}$$

- Individual entries will use subscripts and no boldface, e.g., for entry i:

$$\theta_i$$

Modeling: Linear Models

- Score function is linear in $\boldsymbol{\theta}$:

$$\mathrm{score}(x, y, \boldsymbol{\theta}) = \sum_i \theta_i f_i(x, y) = \boldsymbol{\theta} \cdot \boldsymbol{f}(x, y) = \boldsymbol{\theta}^\top \boldsymbol{f}(x, y)$$

- $\boldsymbol{f}$: feature function vector
- $\boldsymbol{\theta}$: weight vector

Modeling: Linear Models

- Score function is linear in $\boldsymbol{\theta}$:

$$\text{score}(x, y, \boldsymbol{\theta}) = \sum_i \theta_i f_i(x, y) = \boldsymbol{\theta} \cdot \boldsymbol{f}(x, y) = \boldsymbol{\theta}^\top \boldsymbol{f}(x, y)$$

- $\boldsymbol{f}$: feature function vector
- $\boldsymbol{\theta}$: weight vector
- How do we define $\boldsymbol{f}$?

Defining Features

- This is a large part of NLP
- Last 20 years: **feature engineering**
- Last 2 years: **representation learning**

Defining Features

- This is a large part of NLP
- Last 20 years: **feature engineering**
- Last 2 years: **representation learning**

- In this course, we'll do both
- Learning representations doesn't mean that we don't have to look at the data or the output!
- There's still plenty of engineering required in representation learning

Feature Engineering

- Often decried as “costly, hand-crafted, expensive, domain-specific”, etc.
- But in practice, simple features typically give the bulk of the performance

- Let’s get concrete: how should we define features for text classification?

Feature Engineering for Text Classification

$$\mathrm{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) = \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

Feature Engineering for Text Classification

$$\text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) = \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

$\boldsymbol{x}$ is now a vector because it is a sequence of words

Feature Engineering for Text Classification

$$\text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) = \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

$\boldsymbol{x}$ is now a vector because it is a sequence of words

let's consider sentiment analysis:
$y \in \{\text{positive}, \text{negative}\}$

Feature Engineering for Text Classification

$$\text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) = \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

$\boldsymbol{x}$ is now a vector because it is a sequence of words

let's consider sentiment analysis:

$$y \in \{\text{positive}, \text{negative}\}$$

so, here is our sentiment classifier that uses a linear model:

$$\text{classify}_{\text{senti}}^{\text{linear}}(\boldsymbol{x}, \boldsymbol{\theta}) = \underset{y \in \{\text{positive}, \text{negative}\}}{\text{argmax}} \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

Feature Engineering for Text Classification

$$\text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) = \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

- Two features:

$$f_1(\boldsymbol{x}, y) = \mathbb{I}[y = \text{ positive}] \land \mathbb{I}[\boldsymbol{x} \text{ contains } \textit{great}]$$
$$f_2(\boldsymbol{x}, y) = \mathbb{I}[y = \text{ negative}] \land \mathbb{I}[\boldsymbol{x} \text{ contains } \textit{great}]$$

where $\mathbb{I}[S] = 1$ if S is true, 0 otherwise

Feature Engineering for Text Classification

$$\text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) = \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

- Two features:

$$f_1(\boldsymbol{x}, y) = \mathbb{I}[y = \text{ positive}] \wedge \mathbb{I}[\boldsymbol{x} \text{ contains } \textit{great}]$$
$$f_2(\boldsymbol{x}, y) = \mathbb{I}[y = \text{ negative}] \wedge \mathbb{I}[\boldsymbol{x} \text{ contains } \textit{great}]$$

where $\mathbb{I}[S] = 1$ if S is true, 0 otherwise

- What should the weights be?

$$\theta_1 > \theta_2? \qquad \theta_1 = \theta_2? \qquad \theta_1 < \theta_2?$$

Feature Engineering for Text Classification

- Two features:

$$f_1(\boldsymbol{x}, y) = \mathbb{I}[y = \text{ positive}] \wedge \mathbb{I}[\boldsymbol{x} \text{ contains } \textit{great}]$$
$$f_2(\boldsymbol{x}, y) = \mathbb{I}[y = \text{ negative}] \wedge \mathbb{I}[\boldsymbol{x} \text{ contains } \textit{great}]$$

- Let's say we set $\theta_1 > \theta_2$
- On sentences containing "***great***" in the Stanford Sentiment Treebank training data, this would get us an accuracy of 69%
- But "***great***'' only appears in 83/6911 examples

Feature Engineering for Text Classification

- Two features:

$$f_1(x, y) = \mathbb{I}[y = \text{ positive}] \wedge \mathbb{I}[x \text{ contains } \textit{great}]$$
$$f_2(x, y) = \mathbb{I}[y = \text{ negative}] \wedge \mathbb{I}[x \text{ contains } \textit{great}]$$

ambiguity: "***great***" can mean different things in different contexts

- On sentences containing "***great***" in the Stanford Sentiment Treebank training data, this would get us an accuracy of 69%
- But "***great***" only appears in 83/6911 examples

variability: many other words can indicate positive sentiment

- Usually, ***great*** indicates positive sentiment:

  *The most wondrous love story in years, it is a **great** film.*

  *A **great** companion piece to other Napoleon films .*

- Sometimes not. Why?

- Usually, ***great*** indicates positive sentiment:

  *The most wondrous love story in years, it is a **great** film.*

  *A **great** companion piece to other Napoleon films .*

- Sometimes not. Why?

  Negation: *It's not a **great** monster movie .*

  Different sense: *There's a **great** deal of corny dialogue and preposterous moments .*

  Multiple sentiments: *A **great** ensemble cast can't lift this heartfelt enterprise out of the familiar.*

Feature Engineering for Text Classification

$$\text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) = \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

- What about a feature like the following?

$$f_3(\boldsymbol{x}, y) = \mathbb{I}[\boldsymbol{x} \text{ contains } \textit{great}]$$

- What should its weight be?

Feature Engineering for Text Classification

$$\text{score}(\boldsymbol{x}, y, \boldsymbol{\theta}) = \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

- What about a feature like the following?

$$f_3(\boldsymbol{x}, y) = \mathbb{I}[\boldsymbol{x} \text{ contains } \textit{great}]$$

- What should its weight be?
- Doesn't matter.
- Why?

$$\text{classify}^{\text{linear}}_{\text{senti}}(\boldsymbol{x}, \boldsymbol{\theta}) = \operatorname*{argmax}_{y \in \{\text{positive},\text{negative}\}} \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

Text Classification

our linear sentiment classifier:

$$\text{classify}_{\text{senti}}^{\text{linear}}(\boldsymbol{x}, \boldsymbol{\theta}) = \underset{y \in \{\text{positive},\text{negative}\}}{\text{argmax}} \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

Inference for Text Classification

$$\text{classify}_{\text{senti}}^{\text{linear}}(\boldsymbol{x}, \boldsymbol{\theta}) = \underset{y \in \{\text{positive}, \text{negative}\}}{\text{argmax}} \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

inference: solve argmax

Inference for Text Classification

$$\text{classify}_{\text{senti}}^{\text{linear}}(\boldsymbol{x}, \boldsymbol{\theta}) = \operatorname*{argmax}_{y \in \{\text{positive}, \text{negative}\}} \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

inference: solve argmax

- trivial (loop over labels)

Text Classification

$$\mathrm{classify}^{\mathrm{linear}}_{\mathrm{senti}}(\boldsymbol{x}, \boldsymbol{\theta}) = \underset{y \in \{\mathrm{positive}, \mathrm{negative}\}}{\mathrm{argmax}} \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

Learning for Text Classification

$$\text{classify}^{\text{linear}}_{\text{senti}}(\boldsymbol{x}, \boldsymbol{\theta}) = \operatorname*{argmax}_{y \in \{\text{positive}, \text{negative}\}} \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

learning: choose $\boldsymbol{\theta}$

Learning for Text Classification

$$\text{classify}^{\text{linear}}_{\text{senti}}(\boldsymbol{x}, \boldsymbol{\theta}) = \operatorname*{argmax}_{y \in \{\text{positive}, \text{negative}\}} \sum_i \theta_i f_i(\boldsymbol{x}, y)$$

learning: choose $\boldsymbol{\theta}$

- There are many ways to choose $\boldsymbol{\theta}$

Experimental Practice

- in the beginning, we just had data

Experimental Practice

- in the beginning, we just had data
- first innovation: split into train and test
  - motivation: simulate conditions of applying system in practice

Experimental Practice

- in the beginning, we just had data
- first innovation: split into train and test
  - motivation: simulate conditions of applying system in practice
- but, there's a problem with this…

Experimental Practice

- in the beginning, we just had data
- first innovation: split into train and test
  - motivation: simulate conditions of applying system in practice
- but, there's a problem with this…
  - we need to explore and evaluate methodological choices
  - after multiple evaluations on test, it is no longer a simulation of real-world conditions

Experimental Practice

- we need to explore/evaluate methodological choices
- what should we do?
  - some use cross validation on train, but this is slow and doesn’t quite simulate real-world settings (why?)

Experimental Practice

- we need to explore/evaluate methodological choices
- what should we do?
  - some use cross validation on train, but this is slow and doesn't quite simulate real-world settings (why?)
- second innovation: divide data into train, test, and a third set called development (dev) or validation (val)
  - use dev/val to evaluate choices
  - then, when ready to write the paper, evaluate the best model on test

Experimental Practice

- we need to explore/evaluate methodological choices
- what should we do?
  - some use cross validation on train, but this is slow and doesn't quite simulate real-world settings (why?)
- second innovation: divide data into train, test, and a third set called development (dev) or validation (val)
  - use dev/val to evaluate choices
  - then, when ready to write the paper, evaluate the best model on test
- are we done yet? no! there's still a problem:

Experimental Practice

- we need to explore/evaluate methodological choices
- what should we do?
  - some use cross validation on train, but this is slow and doesn't quite simulate real-world settings (why?)
- second innovation: divide data into train, test, and a third set called development (dev) or validation (val)
  - use dev/val to evaluate choices
  - then, when ready to write the paper, evaluate the best model on test
- are we done yet? no! there's still a problem:
  - overfitting to dev/val

Experimental Practice

- best practice: split data into train, development (dev), development test (devtest), and test
  - train model on train, tune hyperparameter values on dev, do preliminary testing on devtest, do final testing on test a single time when writing the paper
  - Even better to have even more test sets! test1, test2, etc.

- experimental credibility is a huge component of doing useful research
- when you publish a result, it had better be replicable without tuning anything on test

Don’t Cheat!

SECTIONS

The New York Times

SUBSCRIBE

LOG IN

TECHNOLOGY

Computer Scientists Are Astir After Baidu Team Is Barred From A.I. Competition

By **JOHN MARKOFF** JUNE 3, 2015

Email

Share

Tweet

Save

SAN FRANCISCO — A group of researchers at the Chinese web services company Baidu have been barred from participating in an international competition for artificial intelligence technology after organizers discovered that the Baidu scientists broke the contest’s rules.