CS 106S Week 2

Sentiment Analysis and Refugee Tweets

Ben Yan, Spring 2025

cs106s.stanford.edu

Stanford | ENGINEERING
Computer Science

Welcome to Week 2 of Class!

Spring **Summer**

Hope that you're enjoying the spring weather!

The Map For Today

1. sentiment analysis overview
2. JavaScript objects – beautiful, complex
3. project: refugee sentiment on Twitter
4. implementation & check-off form!

Natural Language Processing

Gemini

Claude

Large Language Models

Sentiment Analysis

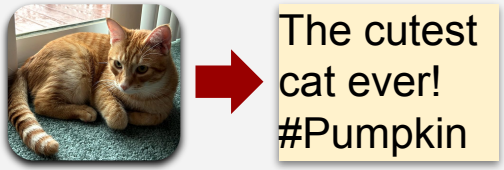
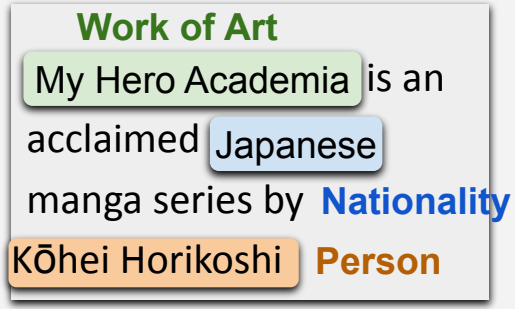
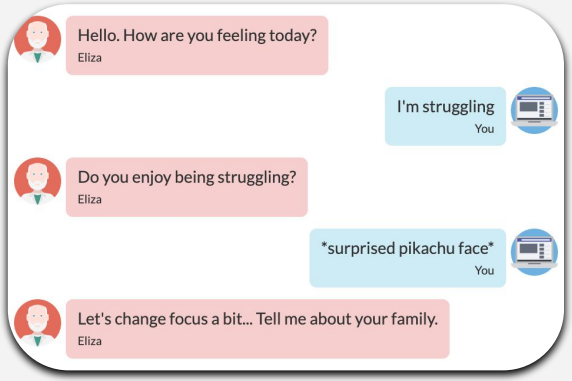


Image Captioning

Language Translation



Named Entity Recognition



Chatbots (e.g., 1960s ELIZA)

Speech Recognition

And a lot more!

Today: Sentiment Analysis

sen·ti·ment a·nal·y·sis

noun

the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc., is positive, negative, or neutral.
"companies have key lessons to learn about harnessing the power of social media and sentiment analysis"

Google Search / Oxford Languages

Today: Sentiment Analysis

sen·ti·ment a·nal·y·sis

noun

the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc., is positive, negative, or neutral.
"companies have key lessons to learn about harnessing the power of social media and sentiment analysis"

Google Search / Oxford Languages

Sentiment Analysis Out There

Google Product Search

HP Officejet 6500A Plus e-All-in-One Color Ink-jet - Fax / copier / printer / scanner
$89 online, **$100** nearby ★★★★☆ 377 reviews
September 2010 - Printer - HP - Inkjet - Office - Copier - Color - Scanner - Fax - 250 sh

Reviews

Summary - Based on 377 reviews

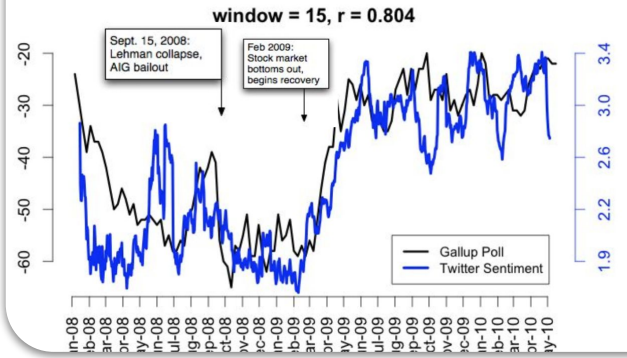
1 star	2	3	4 stars	5 stars

What people are saying

ease of use	"This was very easy to setup to four computers."
value	"Appreciate good quality at a fair price."
setup	"Overall pretty easy setup."
customer service	"I DO like honest tech support people."
size	"Pretty Paper weight."
mode	"Photos were fair on the high quality mode."

Twitter sentiment versus Gallup Poll of Consumer Confidence

Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. **2010.** From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In ICWSM-2010

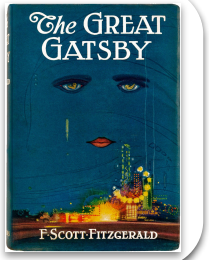


★★☆☆☆

There was one thing I really liked about *The Great Gatsby*.

It was short.

goodreads



How can we tell sentiment?

- look at a piece of text
- extrapolate “features” in the text
  - common features:
    - is a **word** positive or negative?

awesome | terrible | spectacular | unbelievable

    - is a **phrase** positive or negative?

terribly awesome | not good | what in the world was that

- classify the piece as positive or negative

Sentiment Analysis Examples!

Positive or negative movie review?

- unbelievably disappointing

- Full of zany characters and richly applied satire, and some great plot twists

- this is the greatest screwball comedy ever filmed

- It was pathetic. The worst part about it was the boxing scenes.

What words tell you each review is positive or negative? What words are kind of ambivalent?

Sentiment Analysis Examples!

Positive or negative movie review?

- unbelievably disappointing

- Full of zany characters and richly applied satire, and some great plot twists

- this is the greatest screwball comedy ever filmed

- It was pathetic. The worst part about it was the boxing scenes.

What words tell you each review is positive or negative? What words are kind of ambivalent?

A Bit More Challenging

Letterboxd

"how does wes anderson sleep at night knowing his best movie was the cat in the hat directed by bo welch"

"this cured my depression and brought it back full force in the span of 2 hours"

"over 75% of this movie is just memes"

"lost my mind everytime he started singing (so the whole movie)"

Linguistics of Restaurant Reviews

Negative Reviews

The bartender... absolutely horrible... we waited 10 min before we even got her attention... and then we had to wait 45 - FORTY FIVE! - minutes for our entrees... stalk the waitress to get the cheque... she didn't make eye contact or even break her stride to wait for a response ...

Negative sentiment language
horrible awful terrible bad disgusting

Past narratives about people
waited, didn't, was
he, she, his, her,
manager, customer, waitress, waiter

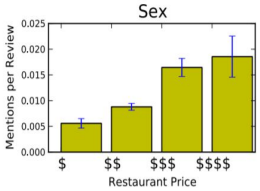
Frequent mentions of we and us
... **we** were ignored until **we** flagged down a waiter to get **our** waitress ..

Positive Reviews

What about positive reviews?
Sex, Drugs, and Dessert

addicted to pepper shooters
garlic noodles... my *drug of* choice
the fries are *like crack*

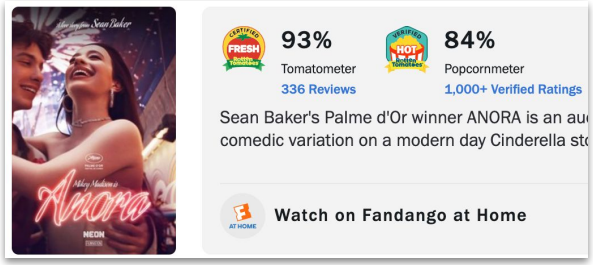
orgasmic pastry
sexy food
seductively seared fois gras

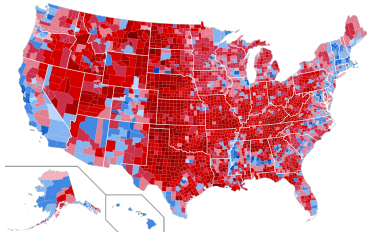


Prof. Dan Jurafsky.
Check out more **here**!

Uses of Sentiment Analysis

- ★ **movie reviews:** what does the public think of a movie?
- ★ **products:** what do people think of the new iPhone?
- ★ **public sentiment:** reaction to news? current events?
- ★ **politics:** what do people think of x issue?
- ★ **prediction:** what will the outcome of an election be based on social media data (and polling ofc)?

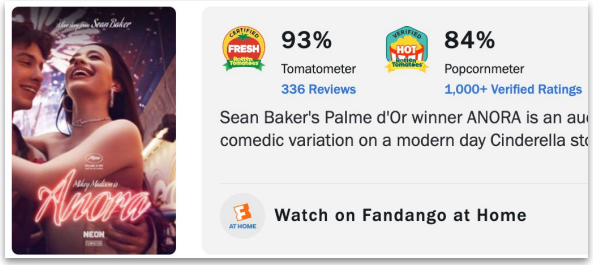


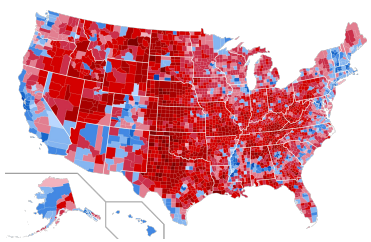


Windows 8

Uses of Sentiment Analysis

- ★ **movie reviews:** what does the public think of a movie?
- ★ **products:** what do people think of the new iPhone?
- ★ **public sentiment:** reaction to news? current events?
- ★ **politics:** what do people think of x issue?
- ★ **prediction:** what will the outcome of an election be based on social media data (and polling ofc)?





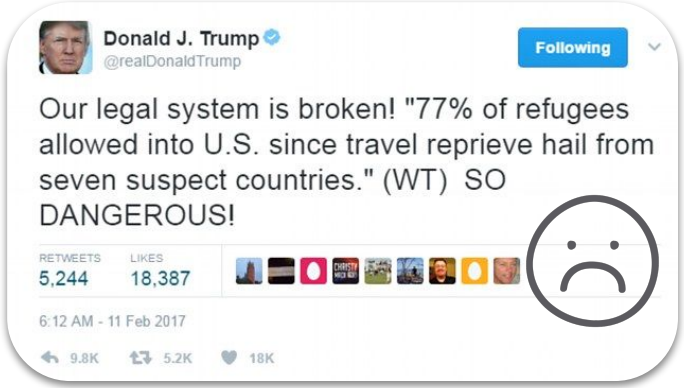
Windows 8

In-Class Project

Build a sentiment classifier for refugee tweets.

Can we classify tweets as **pro** or **anti**-refugee?

🐦 Why Twitter? 🐦



Motivation / Why?

- **Bots and automated tools that detect hateful tweets** and respond with educational resources
  - Sway public opinion, combat misinformation

- **Understanding general opinion** in the US
  - Geo-tagging of tweets for mapping sentiment
  - Analytics for upcoming 2026 midterm elections

Training & Testing Splits

Training Set | **Test Set**

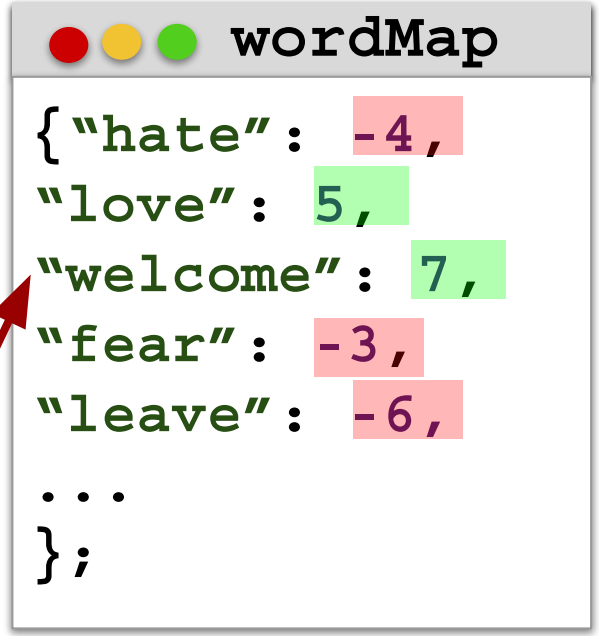
We use this to train/tune the model i.e. learn the sentiment of each word

We use this to evaluate the model's performance

- ❑ Machine learning tasks require abundant human-labeled data
- ❑ Our data is in **trainTweets.js** and **testTweets.js, content warning**

‘Key’ Approach

- **give each word a value**
- represents **how pro/anti-refugee sentiment** it is
- determine how pro/anti-refugee a given word is from the training set
- **ultimate goal to create wordMap**
- training will be the process of making wordMap



wordMap

wordMap[word] = (# of times the word appears in positive tweets) − (# of times the words appears in negative tweets)

Tweets within the **training set**

Review of Objects / Dictionaries / Maps

Python
CS106A/AX

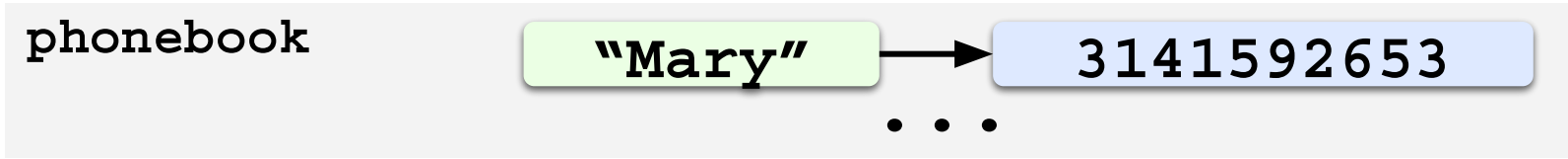
```
phonebook = dict()
phonebook["Mary"] = 3141592653
```

C++
CS106B

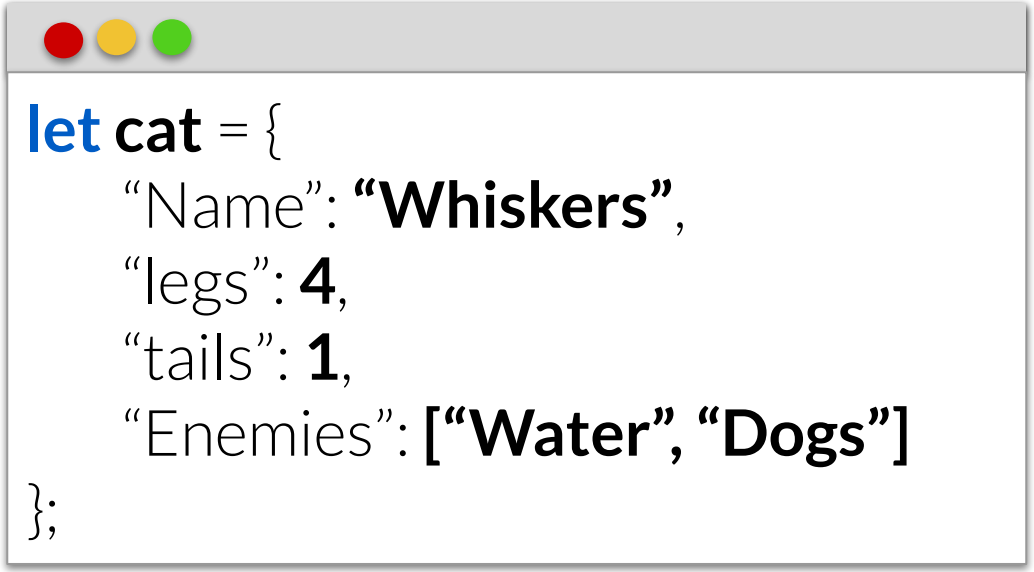
```
Map<string, int> phonebook;
phonebook["Mary"] = 3141592653;
```

Java
AP Comp Sci

```
HashMap<String, Integer> phonebook =
new HashMap<String, Integer>();
phonebook.put("Mary", 3141592653);
```

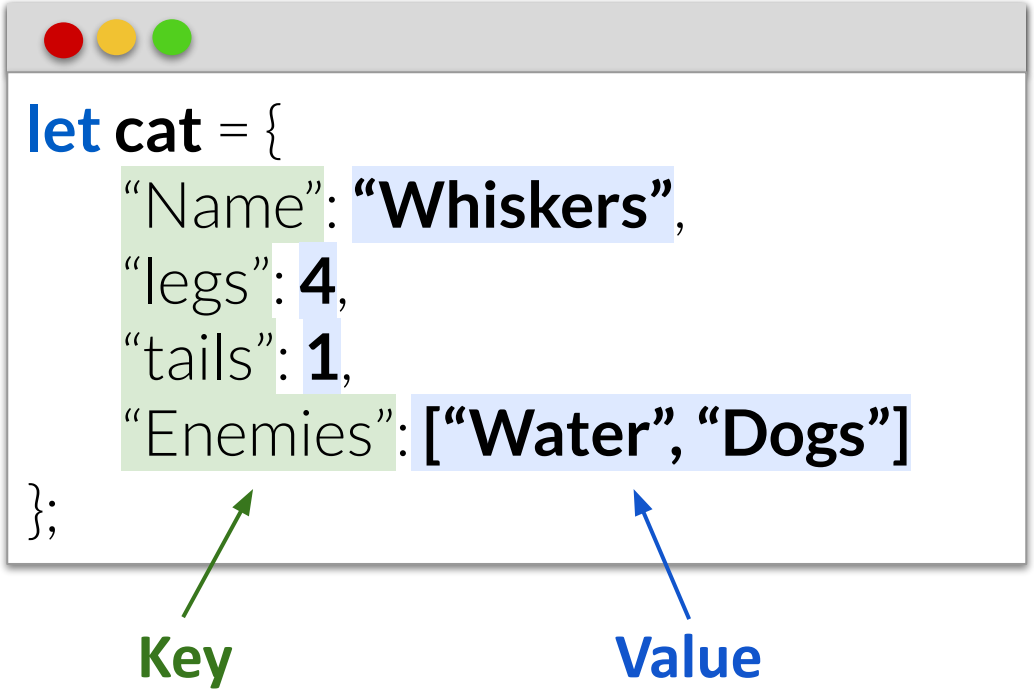


JavaScript Objects



```
let cat = {
    "Name": "Whiskers",
    "legs": 4,
    "tails": 1,
    "Enemies": ["Water", "Dogs"]
};
```

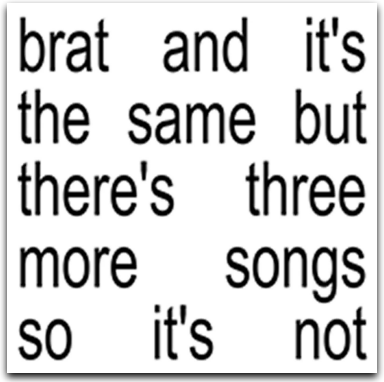
JavaScript Objects



In JavaScript, **objects** store data / **values** which you can access through their **properties or keys**

JavaScript Objects

```
let album = {
    "name": "Brat",
    "artist": "Charli XCX"
};
```

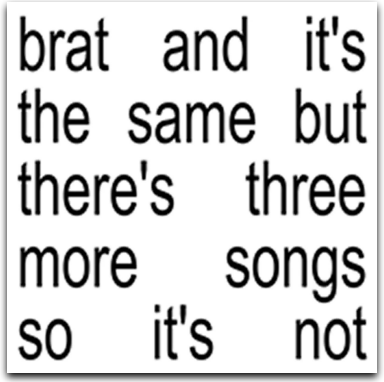



```
let name = album.name;          //"Brat"
let name = album["name"];       // "Brat"

album.name = "Brat extended"; // can change!
```

Other Object Functions

```
let album = {
    "name": "Brat",
    "artist": "Charli XCX"
};
```




```
album.hasOwnProperty("name")          // true
"name" in album                       // true

album["topHit"] = "Apple"             // new key-value pair! 🍏
album.topHit = "Apple"                // alternative to above 🍏
```

Exercise: Working with JS Objects 📽️

```
function movieMarathon(movies, movieRuntimes)
```

Suppose you have an object of several movie titles and their runtimes (in minutes), e.g.,

```
movieRuntimes = {👩‍🎤"Wicked": 161, 🚢 "Titanic": 195, ...}
```

Write a function **`movieMarathon`** that also takes an array **`movies`** of movie titles, iterates over them, and **adds up their runtimes, returning that sum total.**

One caveat: For each movie, it should first check if it's in **`movieRuntimes`**, and if not, assume that it's 120 minutes / 2 hours (~ average runtime for a movie).

```
e.g., movies = ["Wicked", "CS109 Lec-3"]
```

Wicked	✅ runtime known	161 min
"CS109 Lec-3"	❌ runtime unknown	120 min

Total **281 min**

Exercise: Working with JS Objects 📽️

```
function movieMarathon(movies, movieRuntimes){
   let totalRuntime = 0;
   for (let movie of movies){
        if (movie in movieRuntimes){
            totalRuntime += movieRuntimes[movie];
        } else{ //if runtime not known, assume 120 min
            totalRuntime += 120;
        }
   }
   return totalRuntime;
}
```

e.g., movies = [“Wicked”, “CS109 Lec-3”]

Wicked	✅ runtime known	161 min	Total **281 min**
“CS109 Lec-3”	❌ runtime unknown	120 min	

Exercise: Working with JS Objects

Syntax for **checking membership**: that a key is currently in an object

```
function movie
    let totalRuntime = 0;
    for (let movie of movies){
        if (movie in movieRuntimes){
            totalRuntime += movieRuntimes[movie];
        } else{ //if runtime not known, assume 120 min
            totalRuntime += 120;
        }
    }
    return t
}
```

Syntax for **accessing a value via its key**, in the object. Also for modifying or creating new key-value pairs, e.g.,

```
movieRuntimes["Big Hero 6"] = 108;
```

Project: Creating wordMap

1. iterate through each training tweet
2. get tweet's classification
3. **for each word in the tweet**
    a. stem the word //var stemmedWord = stemmer(word);
    b. if tweet is anti-refugee, -1 to word's score in wordMap
    c. if pro-refugee, add +1 to word's Score

+1 +1 +1 +1

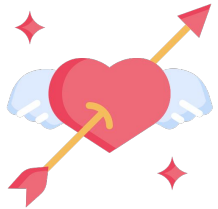
Example: "We love refugees #refugees" Positive (+1)

Creating wordMap

```
let wordMap = {
    “we”: 1,
    “love”: 1,
    “refugees”: 1,
    “#refugees”: 1,
    …
}
```

Stemming

loving



loved

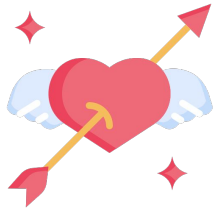
lovely

loves

love

Stemming

loving



loved

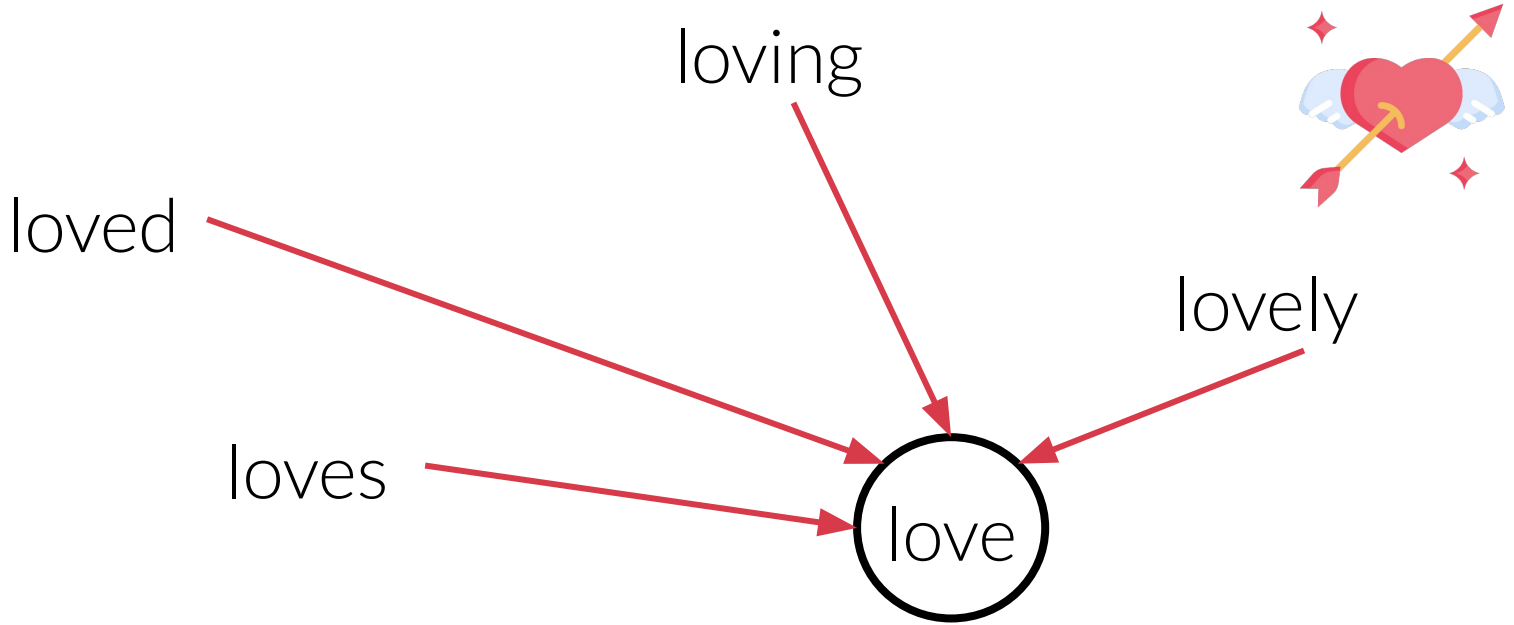
lovely

loves

love

All of these words are inflections of the word “love.”

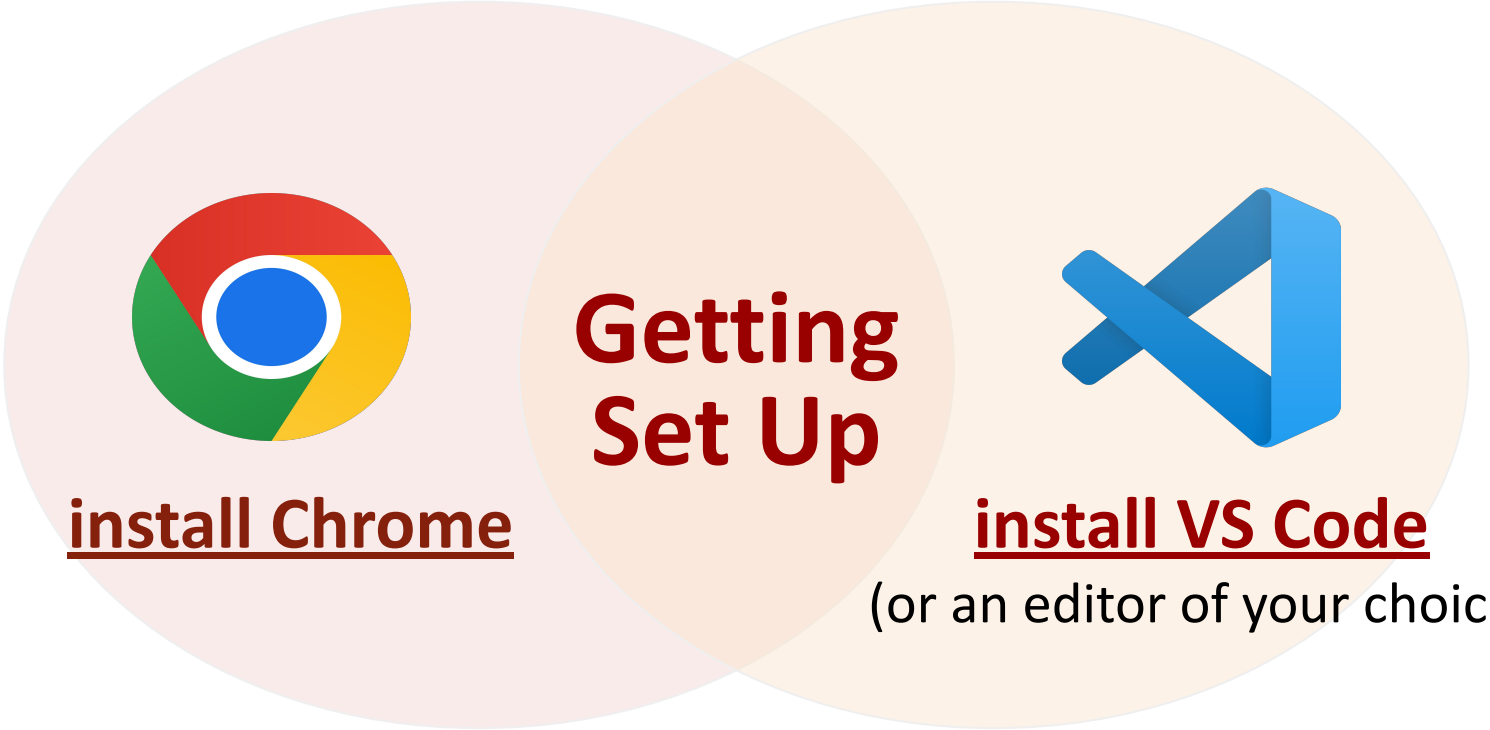
Stemming



let **stemmedWord = stemmer(word);**

Moving on to the Starter Code!

Review: Software Needed

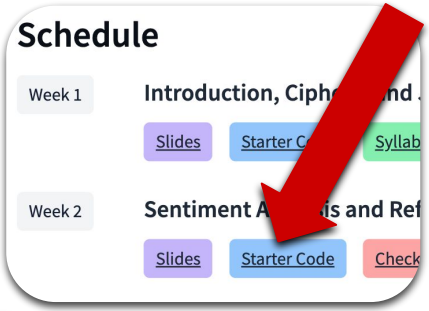


install Chrome

install VS Code

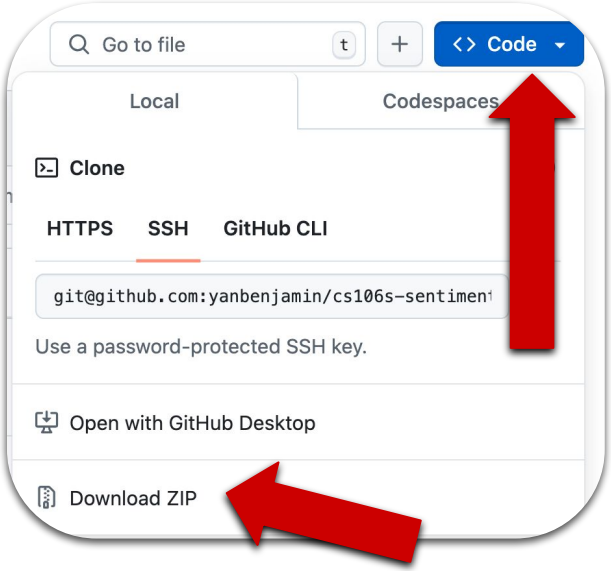
(or an editor of your choice)

Getting Set Up (Downloading Code)

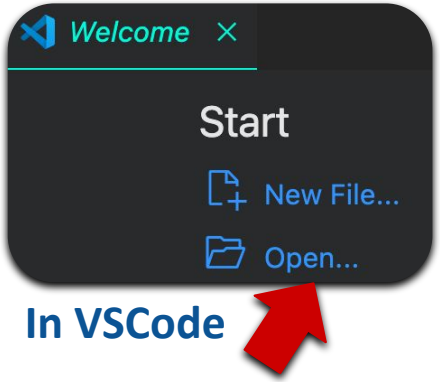


1 Navigate to Week 2 of the Schedule section of **cs106s.stanford.edu**

Also, at this link: **https://github.com/yanbenjamin/cs106s-sentiment**



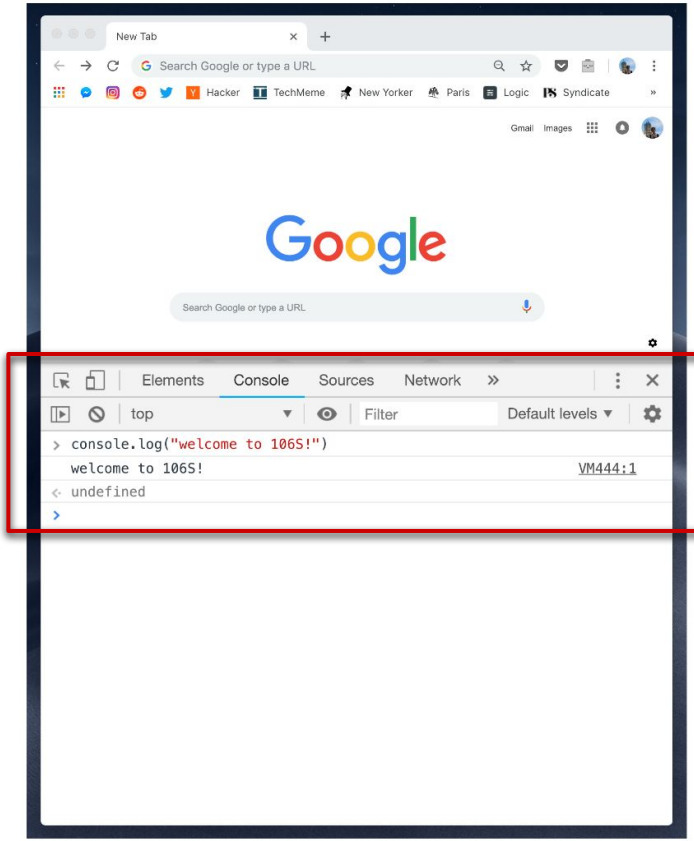
2 Click the bright **"Code"** button, then click "Download ZIP"



In VSCode

3 Unzip the download (clicking .zip file should do the trick) and open the folder / files in your editor

Review: Using Browser to Code



1. Open **index.html** in Chrome
2. On Mac: Press **cmd** + **option** + **j**

   On Windows: Press **ctrl** + **shift** + **j**

Don't let go of the previous key while pressing the next.

Here, **in the console that pops up**, we can input and run JavaScript code!

Classifying Test Tweets

1. initialize a variable to store our “score” of the tweet.
2. for each word in the tweet:
    a. stem the word
    b. if word is in wordMap, add its score to our variable
3. once we go through all words in a tweet, classify!
    a. if score is > 0, classify as pro-refugee (1)
    b. if score is <= 0, classify as anti-refugee (-1)

Classifying Test Tweets

```
let wordMap = {
    "hate": -5,
    "love": +4,
    "welcome": +3,
    "refugees": +1,
    "#refugees": +1,
    ...
}
```

"Don't hate. Love and welcome refugees #refugees" – Positive

score = -5 + 4 + 3 + 1 + 1 = +5

prediction = positive (+1)

That’s the JavaScript model – let’s code it up!

Model Pitfalls

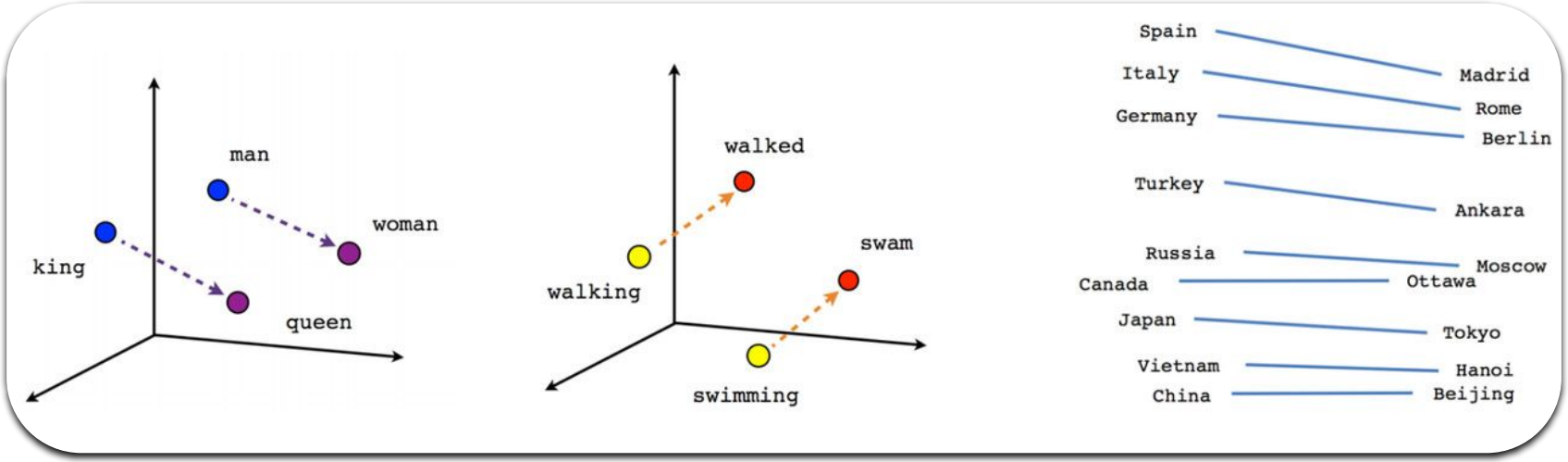
- Negation
  - Use a phrase-based algorithm like bigrams?
  - Can lead to more problems: **overfitting** (doesn't generalize well to new data)

- No weighting
  - Should **"the"** really be treated the same way we treat **"love"** or **"hate"**?

Far From a Solved Problem

1. There was an earthquake in California.
2. The team failed to complete the challenge. (We win/lose!)
3. They said it would be great.
4. They said it would be great, and they were right.
5. They said it would be great, and they were wrong.
6. The party fat-cats are sipping their expensive imported wines.
7. Oh, you're terrible!
8. Here's to ya, ya bastard!
9. Of 2001, "Many consider the masterpiece bewildering, boring, slow-moving or annoying, . . . "
10. long-suffering fans, bittersweet memories, hilariously embarrassing moments, . . .

Glimpse of Modern ML approaches: word2Vec (word vectors)



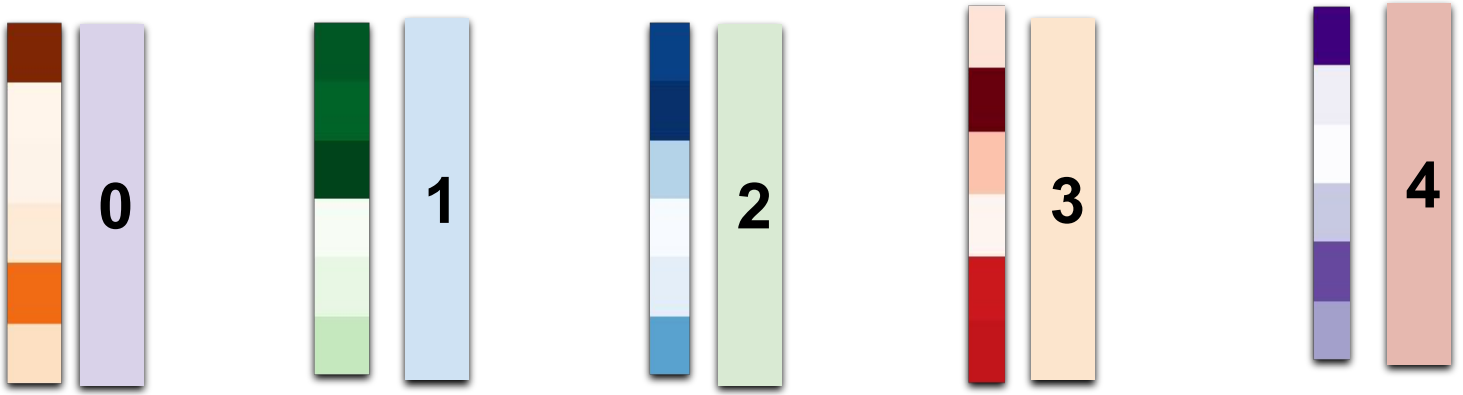
Extra

If you're interested in learning more!

- word2Vec: https://code.google.com/archive/p/word2vec/
- GloVe Vectors Project (Stanford NLP): https://nlp.stanford.edu/projects/glove/

Glimpse of Modern ML approaches: LLM embedding layers

- Generally a sum of a word + position embeddings / vectors



Extra

If you’re interested in learning more!

➢ 🤖 Transformers: https://en.wikipedia.org/wiki/Attention_Is_All_You_Need

Topical Classes @ Stanford

CS 109: Probability for CS (Classification Models)

CS 221: Artificial Intelligence (Sentiment Analysis)

CS 124: From Languages to Information

CS 224N: Natural Language Processing with Deep Learning

CS 224C: NLP for Computational Social Science

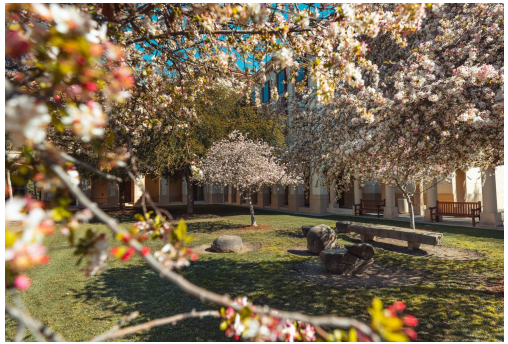
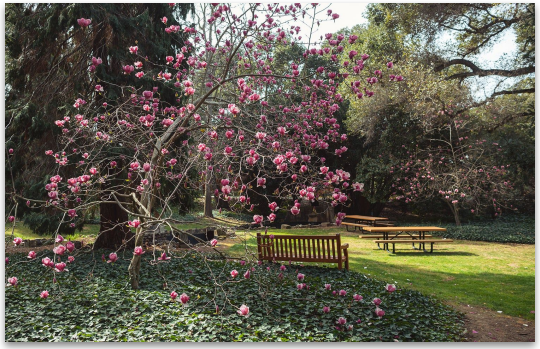
CS 152: Trust & Safety

✅ Check-Off Form!

Another **brief check-off form** (< 5 min to complete) for checking attendance!

For today, click the “Check-Off Form” link in the **Week 2** section of **cs106s.stanford.edu**.

https://tinyurl.com/cs106s-spr25-w2-checkoff (case sensitive!)

Have an awesome Week 2!