



Justin Trudeau @JustinTrudeau

Following

To those fleeing persecution, terror & war,  
Canadians will welcome you, regardless of your  
faith. Diversity is our strength #WelcomeToCanada

RETWEETS LIKES  
**165,284 256,250**

12:20 PM - 28 Jan 2017

# CS 106S Week 2

## Sentiment Analysis and Refugee Tweets

Ben Yan, Winter 2025

# Welcome to Week 2 of Class!



Winter Break



Spring “Break”



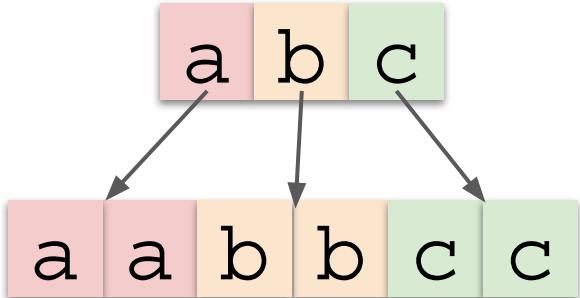
# The Map For Today

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

# Review: General JavaScript Syntax

**Example:** Consider a function **twins** which takes an input string and returns a copy of that string with each character repeated once, e.g.,

- "abc"→"aabbcc"
- minnesota"→"mmiiinnnneesssoottaa".



```
def twins(str): #Python
    result = ""
    for char in str:
        result += char + char
    return result
```

```
function twins(str){ //JavaScript
    let result = "";
    for (let char of str){
        result += char + char;
    }
    return result;
}
```

# Review: General JavaScript Syntax

```
function twins(str){  
    let result = "";  
    for (let char of str){  
        result += char + char;  
    }  
    return result;  
}
```

# Review: General JavaScript Syntax

```
function twins(str) {  
    let result = "";  
    for (let char of str) {  
        result += char + char;  
    }  
    return result;  
}  
  
function twins(str) {  
    let result = "";  
    for (let char of str) {  
        result += char + char;  
    }  
    return result;  
}
```

Unlike Python (blocks of code are defined by indentation), JavaScript uses **curly braces** – similar to C++ / Java.

In Python, having **parentheses** around statements / conditions in **for**, **while**, **if** loops is optional.

In JavaScript, it's mandatory. Here's a **for/of** loop that iterates directly over the characters of a string.

# Review: General JavaScript Syntax

```
function twins(str){  
    let result = "";  
    for (let char of str){  
        result += char + char;  
    }  
    return result;  
}
```

```
function twins(str){  
    let result = "";  
    for (let char of str){  
        result += char + char;  
    }  
    return result;  
}
```

In JavaScript, to declare variables, we use the **let** keyword (or **const**).  
In Python, we could have simply written  
**result = ""**

Unlike C++ or Java, **semi-colons (;)** are **optional** at the end of statements.

But you can include them if you'd like! :)  
It's a style decision.

**JavaScript is a “vibes”-based language,**  
sort of 

# Natural Language Processing

Gemini

Claude



Large Language Models



Sentiment Analysis



The cutest  
cat ever!  
#Pumpkin

Image Captioning

English      ↗      Chinese (Simplified)

I choose you Pikachu

我选择你皮卡丘

Wǒ xuǎnzé nǐ píkāqiū

Language Translation

Work of Art

My Hero Academia is an acclaimed Japanese manga series by Nationality Kōhei Horikoshi Person

Named Entity Recognition

Hello. How are you feeling today?  
Eliza

I'm struggling  
You

Do you enjoy being struggling?  
Eliza

\*surprised pikachu face\*  
You

Let's change focus a bit... Tell me about your family.  
Eliza

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"

# 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"



# 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



### 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."

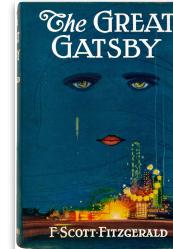
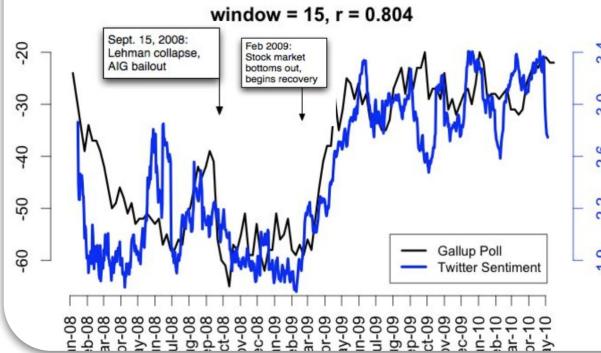


There was one thing I really liked about *The Great Gatsby*.

It was short.

## 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



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



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



“over 75% of this movie is just memes”



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



“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*

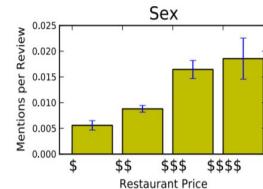
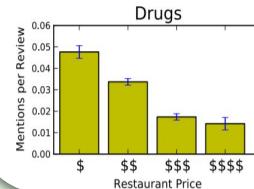
*garlic noodles... my drug of choice*

*the fries are like crack*

*orgasmic pastry*

*sexy food*

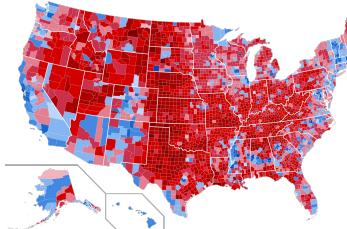
*seductively seared fois gras*



Prof. Dan Jurafsky.  
Check out more [here](#)!

# Uses of Sentiment Analysis

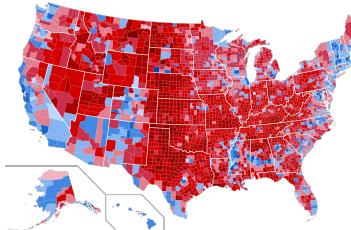
- ★ **movie reviews:** what does the public think of a movie?
- ★ **products:** what do people think of the new iPhone?
- ★ **public sentiment:** reaction to news? current events?
- ★ **politics:** what do people think of x issue?
- ★ **prediction:** what will the outcome of an election be based on social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





# In 'Recent' News

US elections 2024

## 'They're eating the cats': Trump rambles falsely about immigrants in debate

Ex-president's repetition of unsubstantiated claims prompts ABC moderator to fact-check him on air

- [Fact check: Harris and Trump's debate claims at a glance](#)
- [Harris-Trump debate: key takeaways](#)

Alice Herman

Tue 10 Sep 2024 22.28 EDT

 Share



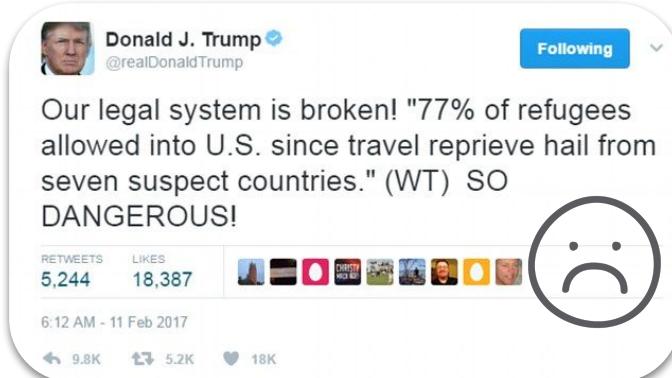
The Guardian, 2024

# In-Class Project

Build a sentiment classifier  
for refugee tweets.

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

Why Twitter? 

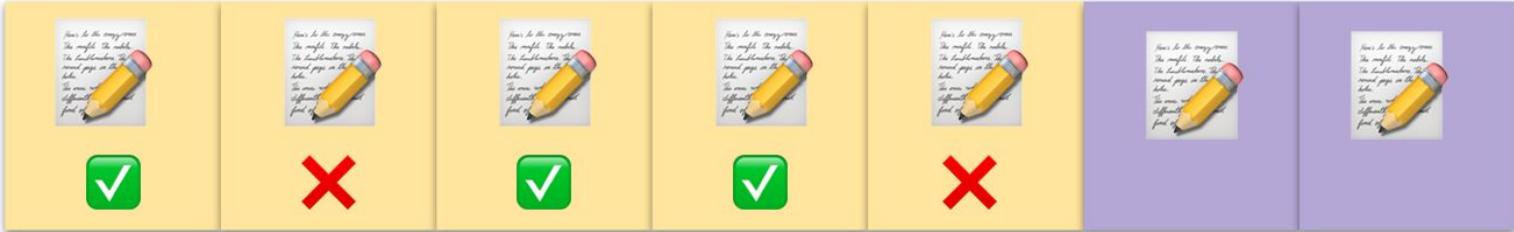


# Motivation / Why?

- ❑ Bots & automated tools that detect hateful tweets and respond with educational resources
  - ❑ Sway public opinion, combat misinformation
- ❑ Understanding general opinion in the US
  - ❑ 2024 – Presidential Election Year, alongside critical Senate and House races in battleground states
  - ❑ Geo-tagging of tweets for mapping sentiment

# Training & Testing Splits

Training 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  
word 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);
```

**phonebook**

**"Mary"**

**3141592653**

• • •

# JavaScript Objects

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

# JavaScript Objects

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

Key

Value

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

# JavaScript Objects

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

brat and it's  
the same but  
there's three  
more songs  
so it's not

```
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"  
};
```

brat and it's  
the same but  
there's three  
more songs  
so it's not

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

// true  
// true

```
album["topHit"] = "Apple"  
album.topHit = "Apple"
```

// new key-value pair!  
// alternative to above

# 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

loved

loves

loving

lovely

love



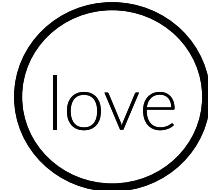
# Stemming

loved

loves

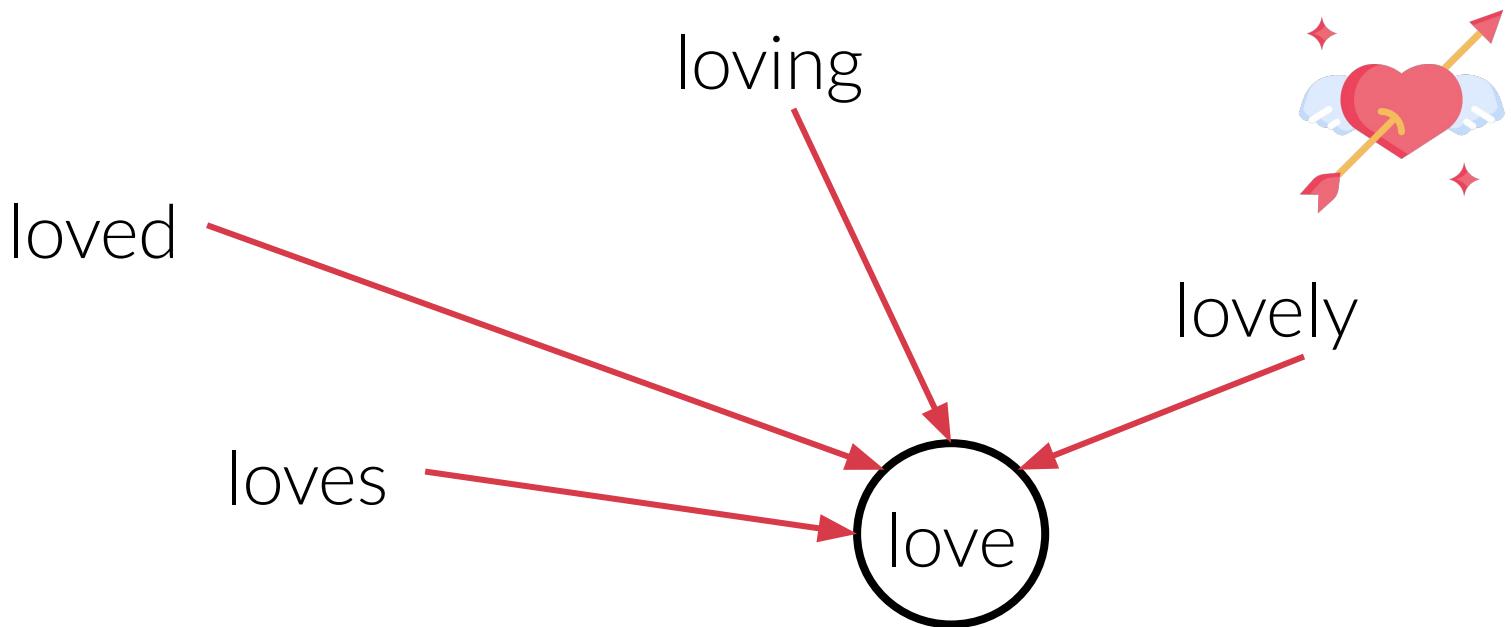
loving

lovely



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

# Stemming



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

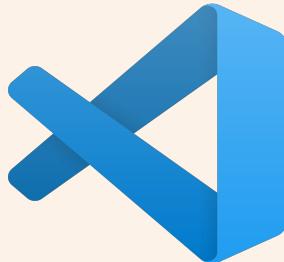
# Moving on to the Starter Code!

# Review: Software Needed



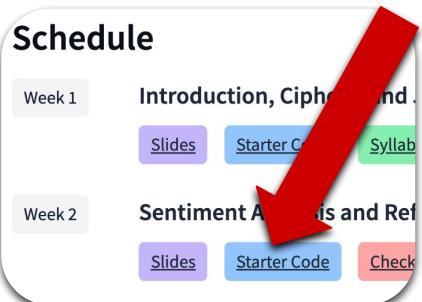
install Chrome

Getting  
Set Up



install VS Code  
(or an editor of your  
choice)

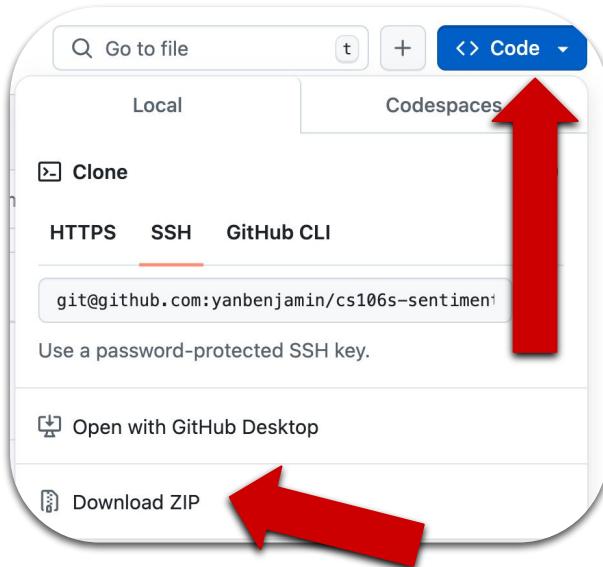
# Getting Set Up (Downloading Code)



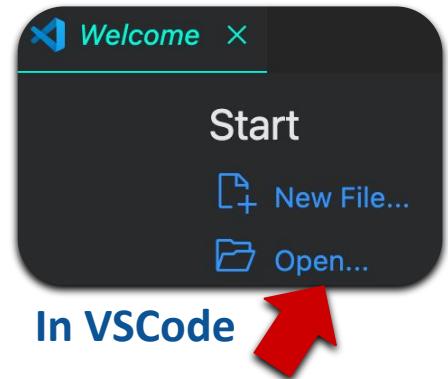
- 1 Navigate to Week 2 of the Schedule section of **cs106s.stanford.edu**

Also, at this link:

<https://github.com/yanbenjamin/cs106s-sentiment>

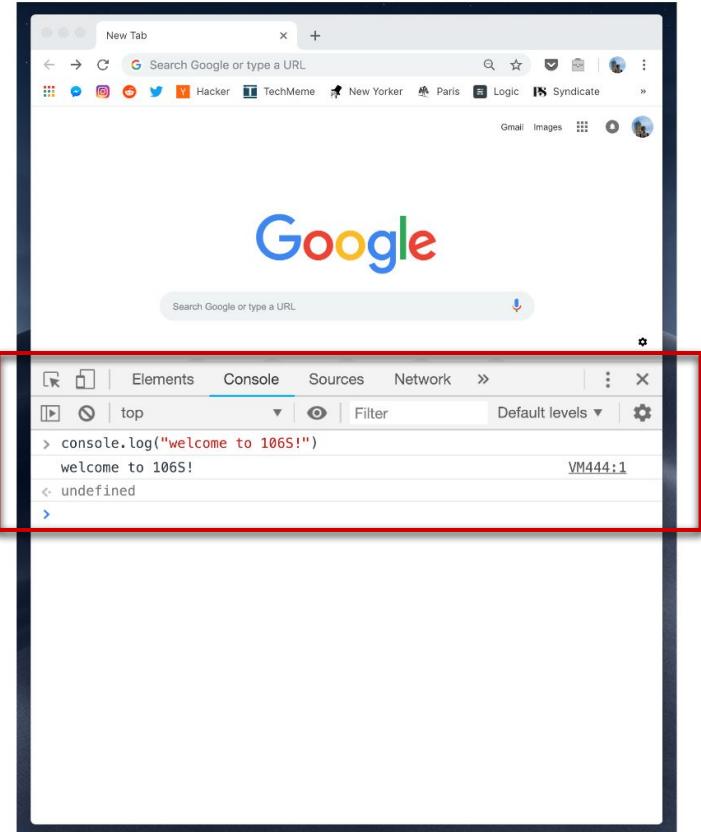


- 2 Click the bright “Code” button, then click “Download ZIP”



- 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  $\leq 0$ , 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!**

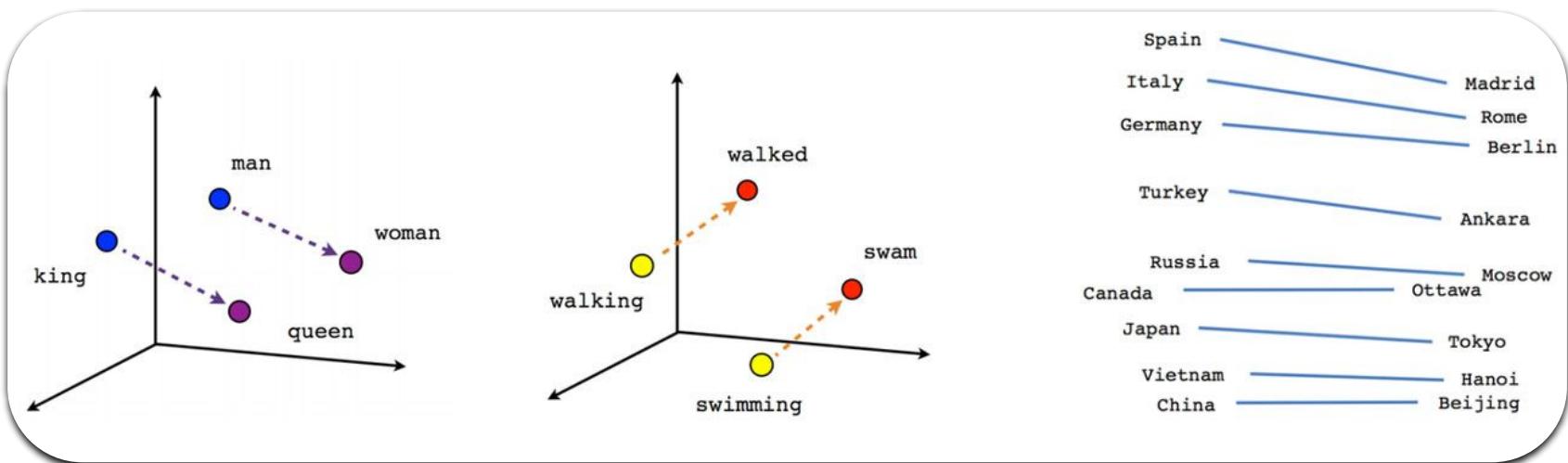
# Model Pitfalls

- 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)



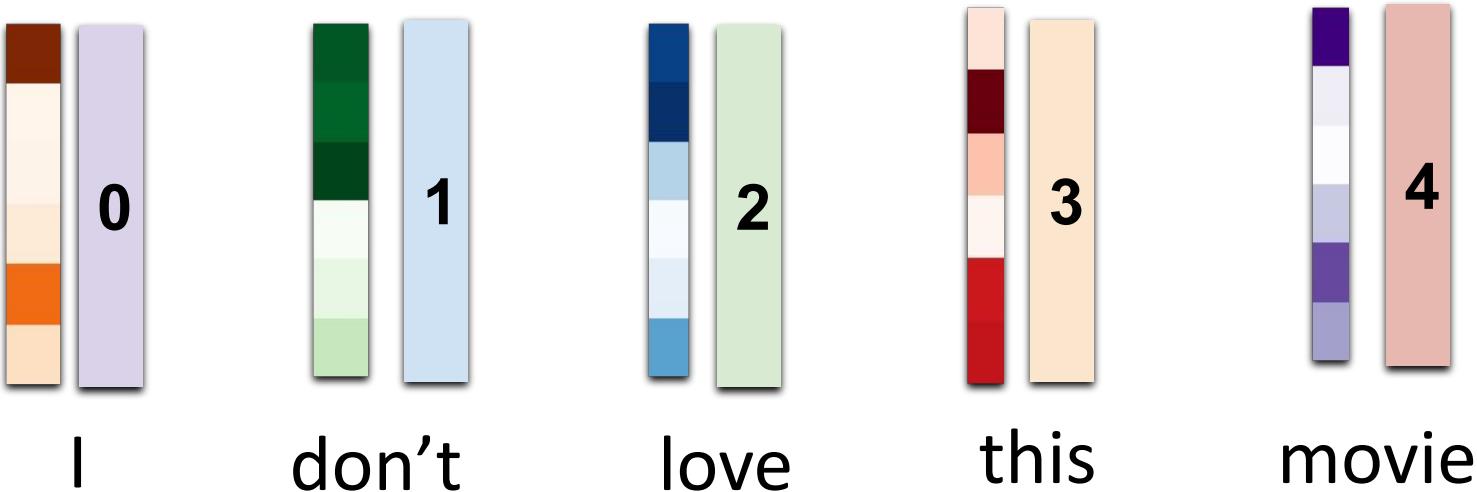
If you're interested in learning more!

**Extra**

- 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





# Topical Classes @ Stanford

**CS 124:** From Languages  
to Information (Jurafsky)

**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://cs106s.stanford.edu).



<https://tinyurl.com/cs106s-win25-w2-checkoff> (case sensitive!)



**Have an awesome Week 2!**