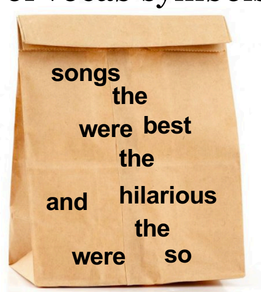
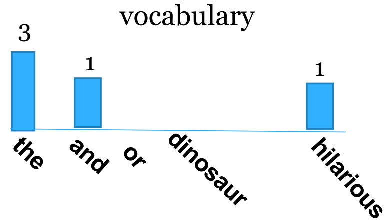
Text Representation
Bag-of-Words and Word Embeddings

unordered "bag"
of vocab symbols



count vector
over large (fixed-size)
vocabulary



Verb tense

Prof. Mike Hughes

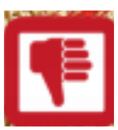
PROJECT 2: Text Sentiment Classification

Example Text Reviews + Labels

Food was so gooodd.

I could eat their bruschetta all day it is devine.

The Songs Were The Best And The Muppets Were So Hilarious.



VERY DISAPPOINTING. there was NO SPEAKERPHONE!!!!

Issues our representation might need to handle

Misspellings?

Food was so gooodd.

Misspellings?

I could eat their bruschetta all day it is devine.

Unfamiliar Words

The Songs Were The Best And The Muppets Were So Hilarious.

VERY DISAPPOINTING. there was NO SPEAKERPHONE!!!!

Capitalization? **Punctuation?**

Sentiment Analysis

- Question: How to represent text reviews?

Friendly staff, **good tacos**, and **fast service**. What more can you **look** for at **taco bell**?

Raw sentences vary in length and content.

$$\phi(x_n)?$$

Need to produce a feature vector of same length for every sentence, whether it has 2 words or 200 words.

Proposal:
1) Define a fixed vocabulary (size F)
2) Feature representation: Count how often each term in vocabulary appears in each review

Bag-of-words representation

$\phi(x_n)$

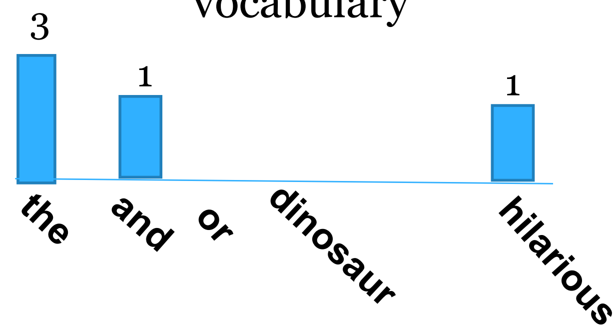
original data

The Songs Were The Best And The Muppets Were So Hilarious.

unordered "bag" of vocab symbols

count vector

over large (fixed-size) vocabulary



Predefined vocabulary

0: the
1: and
2: or
3: dinosaur
...
5005: hilarious

Excludes out of vocabulary words

muppets

Bag of words example

Food was so gooodd.
I could eat their bruschetta all day it is devine.
The Songs Were The Best And The Muppets Were So Hilarious.
So Hilarious were the Muppets and the songs were the best
VERY DISAPPOINTING. there was NO SPEAKERPHONE!!!!

food	the	eat	was/were	best/good	disappoint	no	so
1	0	0	1	1	0	0	1
0	0	1	0	0	0	0	0
0	3	0	2	1	0	0	1
0	3	0	2	1	0	0	1
0	0	0	1	0	1	1	0

Most entries in BoW features will be zero. Can use sparse matrices to store/process efficiently.
Each column of BoW feature array is **interpretable**

BoW: Key Design Decisions for Project B

- how did you "clean" and "standardize" the data? (punctuation, upper vs. lower case, etc)
- how did you determine the final vocabulary set? did you exclude words, and if so how?
- what was your final vocabulary size (or ballpark size(s), if size varies across folds because
- did you use unigrams or bigrams?
- did you use counts or binary values or something else?
- how did you handle out-of-vocabulary words in the test set?

Sentiment Analysis

- Question: How to represent text reviews?

Friendly staff, **good tacos**, and **fast service**. What more can you **look** for at **taco bell**?

Option 1: Bag-of-words count vectors

Option 2: Word embedding vectors

Word Embeddings (word2vec)

Goal: map each word in vocabulary to high-dimensional vector

- Preserve semantic meaning in this new vector space

Verb tense

Ability to make an embedding is implemented as a simple lookup table

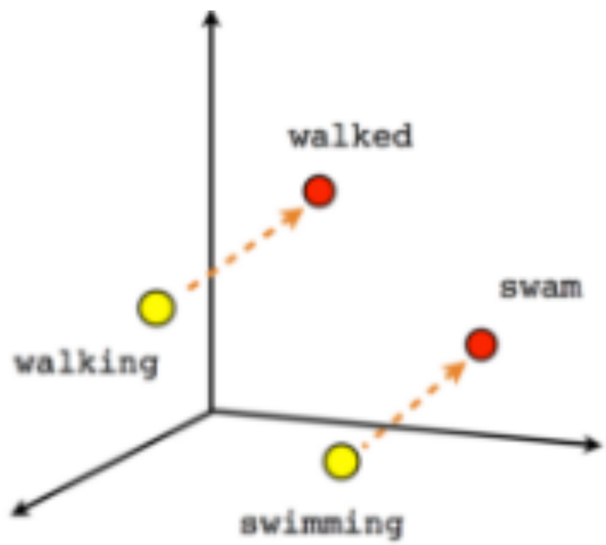
- In: vocabulary word as string (“walked”)
- Out: 50-dimensional vector of reals

Only words in the predefined vocabulary can be mapped to a vector.

Word Embeddings (word2vec)

Goal: map each word in vocabulary to high-dimensional vector

- Preserve semantic meaning in this new vector space



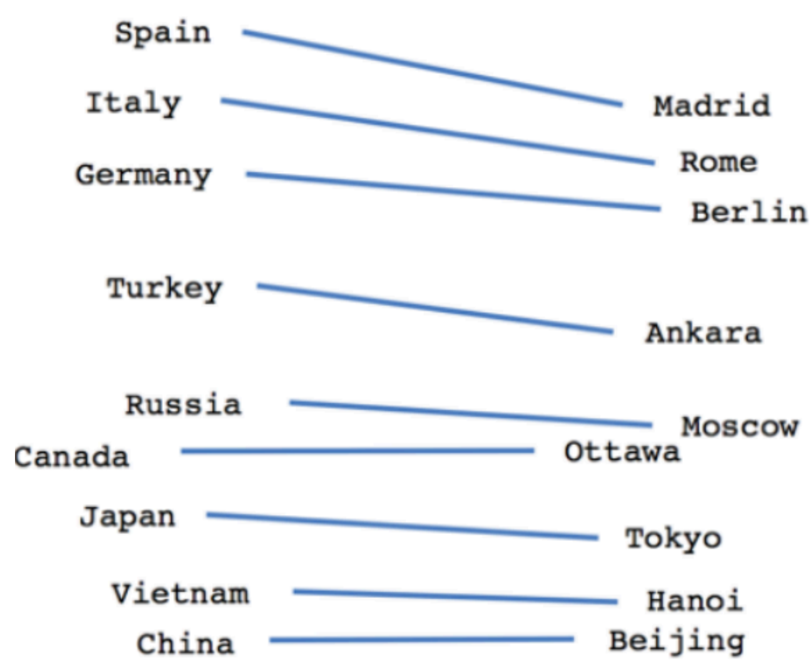
Verb tense

vec(swimming) – vec(swim) + vec(walk) = vec(walking)

Word Embeddings (word2vec)

Goal: map each word in vocabulary to high-dimensional vector

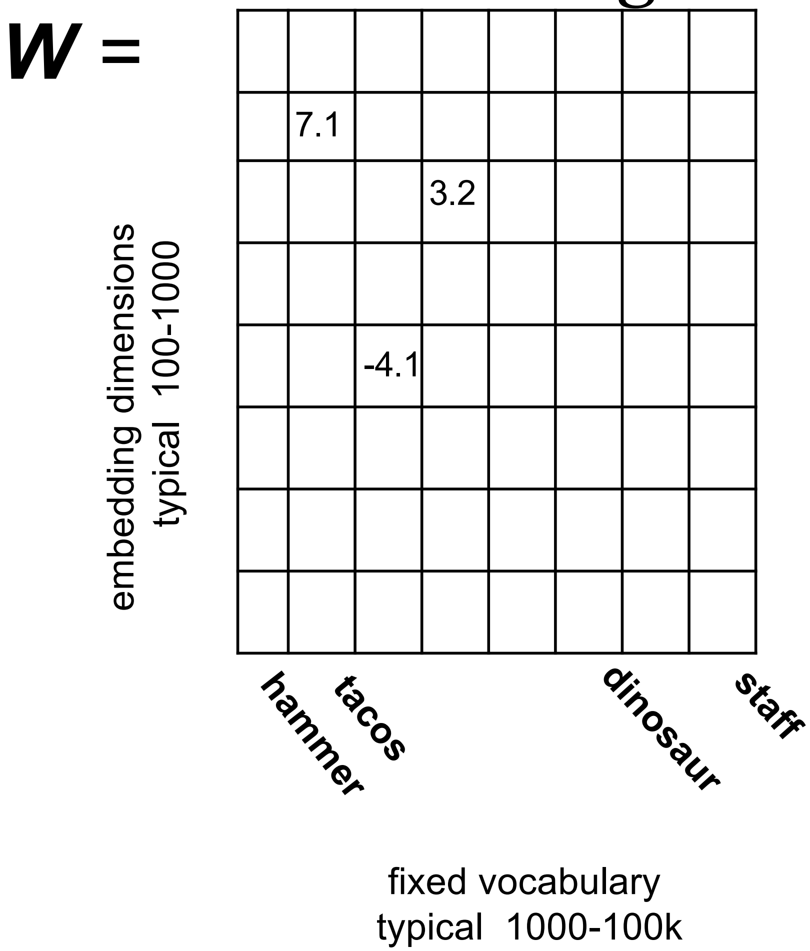
- Preserve semantic meaning in this new vector space



Country-Capital

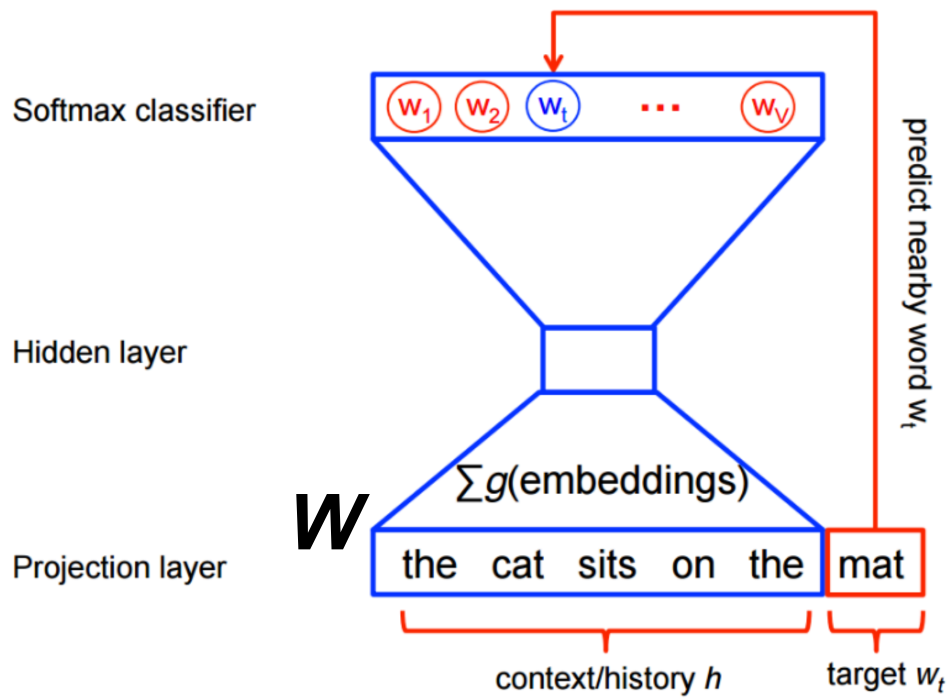
How to learn the embedding?

Goal: learn weights



Training

Reward embeddings that predict nearby words in the sentence.



Credit:
https://www.tensorflow.org/tutorials/representation/word2vec

Example: Word embedding features

Food was so gooodd.
I could eat their bruschetta all day it is devine.
The Songs Were The Best And The Muppets Were So Hilarious.
VERY DISAPPOINTING. there was NO SPEAKERPHONE!!!!

dim1	dim2	dim3	dim4	…	dim49	dim50
+1.2	+1.2	+3.1	-3.2	..	+20.1	-6.8
+5.8	-22.5	+4.4	+4.3		+3.1	-111.1
-8.3	-3.1	-40.8	-4.3		+6.9	-10.8
+3.2	+4.7	-9.6	+5.5		-7.7	+1.8

Entries will be **dense** and **real-valued (negative or positive).**

Each column of feature array might be difficult to interpret.

GloVe: Key Design Decisions for Project B

- how did you "clean" and "standardize" the data? (punctuation, upper vs. lower case, etc)
- how did you determine the final vocabulary set? did you exclude words, and if so how?
- what is the size of your final vocabulary (roughly)?
- how was each vocabulary word represented as an embedding vector?
- how did you combine the embedding vectors for each word in a sentence to produce one vector representation for your sentence? how large is each sentence's feature vector?
- how did you handle out of vocabulary words in the test set?

PROJECT 2:
Text Sentiment Classification

What features are best?
What classifier is best?
What hyperparameters are best?

Lab: Bag of Words

- Part 1-3 : Pure python to build BoW features
- Part 4: How to use with classifier

- Part 5: sklearn CountVectorizer
- Part 6: Doing grid search with sklearn pipelines