CS 584 Natural Language Processing

Introduction

Department of Computer Science
Yue Ning Ph.D.
yue.ning@stevens.edu

COVID19 Logistics

- **Wear a face mask to cover your mouth and nose in all in-person meetings.**
- **If you traveled recently, self-quarantine for 10 days.**
- Contact Student Health Services once you receive a positive or presumed positive result.
- See COVID-19-faqs for school polices.

Course Logistics

- Instructor: Yue Ning
- Meeting time: Wednesday 6:30 PM – 9:00 PM
- Meeting classroom: Babbio 104 and Zoom
- Webpage: https://yue-ning.github.io/cs584-f22.html
- Virtual office hours: Wednesday 4-5pm on Zoom
- Slides uploaded before each lecture

About myself

- Yue Ning

- Assistant Professor at Stevens from 2018.
- PhD in Computer Science and Applications from Virginia Tech.

- Research focus: Machine Learning, Data Mining, Natural Language Processing.

- Personal: **DO NOT call me Madam or ma'am.**
- Research opportunities: **TBA**

Course Assistants

- **Chang Lu** clu13@stevens.edu
  CA office hours: Tuesday 2-4 pm
  Zoom: https://stevens.zoom.us/j/5966521139

- **George Alapatt** galapatt@stevens.edu (not started yet)

- **Qishan Song** qsong4@stevens.edu (not started yet)

Prerequisites

- Python Programming
- Linear Algebra
- Probability and Optimization
- Machine Learning

Self-Introduction

Change your username to your full name and add your pronouns.

What do we learn from this course?

1. Understanding of basic techniques of *processing* human languages and the difficulties in *understanding* languages.
2. Understanding of effective modern methods for NLP
   a. recurrent neural networks, attention, convolutional neural networks, etc
3. Understanding of and ability to build systems for some major tasks in NLP
   a. word semantics, dependency parsing, document classification, language modeling, machine translation, language generation, etc.

Course Outcomes

1. Implement gradient descent (GD) and stochastic gradient descent (SGD) techniques for learning problems and understand the theory behind them.
2. Apply word2vec models in real-world text corpora.
3. Understand the neural networks models and backpropagation optimization. Implement neural network models in TensorFlow or others.
4. Understand dependency parsing, recurrent neural networks and their application in NLP.
5. Understand convolutional neural networks and their application in NLP.
6. Understand sequence to sequence models and attention in NLP deep neural networks.

Recommended Textbooks

- Yoav Goldberg. **Neural Network Methods for Natural Language Processing (NNLP)**.

- Ian Goodfellow and Yoshua Bengio and Aaron Courville, 2016. **Deep Learning (DL)**, MIT Press. We will cover topics including basic neural networks, back propagation, RNN, and CNN.

- Dan Jurafsky and James H. Martin. 2018. **Speech and Language Processing (3rd ed. draft) (SLP)**.

Course work and grading policy

- Assignments (50%): 5 in total
  - Tentative deadlines: 9/28, 10/12, 10/26, 11/30, 12/7
- Final project (20%):
  - Final report and code: Dec. 21
- Midterm exam (20%): **Nov. 2**
  - Scheduled in the middle of the semester
- Participation (5%):
  - Lecture attendance (random two lectures); Canvas discussion
- Quizzes (5%):
  - Pop quizzes; average score of all quizzes;

Late Submission Policy

- **10%** penalty for late submission within **<u>24 hours</u>**.
- **40%** penalty for late submissions within **<u>24-48 hours</u>**.
- After 48 hours, you get **NO** points on the assignment.

Statistics from previous semester

From Spring 2022, CS584

NO cheating/plagiarism

DO **NOT** share your code/writeups anywhere

DO **NOT** use other's code (students in class, students from previous class, strangers online etc)

Stevens Honor System

If cheating/plagiarism found:
You will automatically **FAIL** this course;
And you will be reported to

Stevens Honor System

About this course

Lectures won't always have all the details

- It's up to you to search online OR do some reading to find out more
- This is an active research field. Sometimes there is no clear-cut answer
- Instructors/TAs are happy to discuss with you, but you need to think for yourself

Online lectures and learning:

- Comfortable with learning technology (zoom, canvas, etc)
- Inclusive classroom
- Hardware requirement: computer, microphone, Internet, etc.

This lecture

- ❑ Course Information
- ❑ **<u>Course Overview</u>**
  - ❑ What is NLP? Why is it important?
  - ❑ What will you learn from this course?
- ❑ What are the challenges?
- ❑ Key NLP components

What is NLP?

Wikipedia: Natural language processing (NLP) is a subfield of **computer science**, **information engineering**, and **artificial intelligence** concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyze large amounts of natural language data.

Go beyond the keyword matching

- Identify the structure and meaning of *words, sentences, texts and conversations*
- Deep understanding of broad language
- NLP is all around us

A recent problem: COVID-19

- Racism is a Virus: Anti-Asian Hate and Counterhate in Social Media during the COVID-19 Crisis [link], see related work [1].
- "The spread of COVID-19 has sparked racism, hate, and xenophobia in social media targeted at the Chinese and the broader Asian communities. However, little is known about how racial hate spreads during a pandemic and the role of counter-hate speech in mitigating the spread. Here we study the evolution and spread of anti-Asian hate speech through the lens of Twitter. To do so, we create COVID-HATE, the largest dataset of anti-Asian hate and counter-hate spanning three months (i.e., from January 15, 2020 to April 17, 2020), containing over 30 million tweets. By creating a novel hand-labeled dataset of 2,400 tweets, we train a text classifier to identify hate and counter-hate tweets and finally we identify 891,204 hate and 200,198 counter-hate tweets in COVID-HATE. Using this data, we conduct a comprehensive overview of anti-Asian hate and counter-hate speech on Twitter during COVID-19."
- NLP is impactful!

[1] Jiaxuan Li, Yue Ning. Anti-Asian Hate Speech Detection via Data Augmented Semantic Relation Inference. In Proceedings of the 16th International AAAI Conference on Web and Social Media (ICWSM). Full Paper. Atlanta, Georgia. June 6-9, 2022.

Spell Check

- Using the Web for Language Independent Spellchecking and Autocorrection. Whitelaw *et al*. 2009.
- Deep Spelling: Rethinking spelling correction in the 21st century. Tal Weiss. 2016.
- How to Write a Spelling Corrector. Peter Norvig. 2007.
- Spelling Correction. Stanford NLP. 2008.

Take-home question: What do you think of the simple spell corrector in this article: How to Write a Spelling Corrector. Peter Norvig. 2007? How would you improve it?

Predictive Text

- Typing Assistant. Varun Kathuria. 2017.
- Bayesian Recurrent Neural Networks. Fortunato et al. 2017.
- Contextual LSTM (CLSTM) models for Large scale NLP tasks. Ghosh et al. 2016.

Machine Translation

- [Language Models are Few-Shot Learners](#) (GPT-3). Brown et al. 2020.

Dialog Systems

- Dialog Systems and Chatbots, Chapter 14, Speech and Language Processing. Daniel Jurafsky & James H. Martin. 2018.

Sentiment/Opinion Analysis

- Sentiment Analysis of Twitter Data. Agarwal et al. 2011.
- An Overview of Sentiment Analysis in Social Media and its Applications in Disaster Relief. Beigi et al. 2016.

Text Classification

- Binary classification (e.g., spam or not, positive or negative)

- Multi-class classification (e.g., based on themes or authorships)

- Large Scale Hierarchical Text Classification (Kaggle competition)

Survey Analysis

Image from [Text iQ](#) by qualtrics

- Understand Datasets
- Statistical Analysis
- Sentiment Analysis
- Visualization
- Discover Relationships
- Summarization

Question Answering

Watson computer wins at Jeopardy

- Efficient and Robust Question Answering from Minimal Context over Documents. Min et al. ACL 2018.
- AILA: A Question Answering System in the Legal Domain. Huang et al. IJCAI 2020.
- Learning to Compose Neural Networks for Question Answering. Andreas et al. NAACL 2016.
- CoQA: A Conversational Question Answering Challenge. Reddy et al. TACL 2019.

Natural Language Instruction

- Grounding Natural Language Instructions to Mobile UI Actions. Yang Li, Google Research et al. ACL 2020. [paper].
- Translating Navigation Instructions in Natural Language to a High-Level Plan for Behavioral Robot Navigation. Zang et al. EMNLP 2018.
- Inferring Maps and Behaviors from Natural Language Instructions. Duvallet et al. ISER 2014.

Digital personal assistant

More on natural language instruction

Cortana. Siri amazon echo Google now Facebook M

Semantic parsing - understand tasks

Entity linking - “my husband’s mom” = “Kelly” in the phone book

Information Extraction

- Unstructured text to database entries

New York Times Co. named Russell T. Lewis, 45, president and general manager of its flagship New York Times newspaper, responsible for all business-side activities. He was executive vice president and deputy general manager. He succeeds Lance R. Primis, who in September was named president and chief operating officer of the parent.

Person	Company	Post	State
Russell T. Lewis	New York Times newspaper	president and general manager	start
Russell T. Lewis	New York Times newspaper	executive vice president	end
Lance R. Primis	New York Times Co.	president and CEO	start

Tokenization, Part of Speech Tagging, Ontology Identification, Named Entity Recognition, Co-reference resolution, Syntax/Semantic Parsing, Semantic Role Labeling, Word Embedding, Concept Discovery.

Language Comprehension

Beyonce Giselle Knowles-Carter (born September 4, 1981) is an American singer, songwriter, record producer and actress. Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in the late 1990s as lead singer of R&B girl-group Destiny's Child. Managed by her father, Mathew Knowles, the group became one of the world's best-selling girl groups of all time. Their hiatus saw the release of Beyonce's debut album, Dangerously in Love (2003), which established her as a solo artist worldwide, earned five Grammy Awards and featured the Billboard Hot 100 number-one singles "Crazy in Love" and "Baby Boy".

- Q: In what city and state did Beyonce grow up?
- A: Houston, Texas

- Q: What areas did Beyonce compete in when she was growing up?
- A: Singing and dancing

- Q: When did Beyonce release Dangerously in love?
- A: 2003

A (wikipedia) passage from the SQuAD 2.0 dataset with 3 sample questions and the labeled answer spans. [Rajpurkar et al. 2018]

Ethics, Bias, and Interpretability in NLP

A man and his son are in a terrible accident and are rushed to the hospital in critical care.

The doctor looks at the boy and exclaims "I can't operate on this boy, he's my son!"

How could this be?

- Biases in Data: Labels, Data representations, Selection bias
- Biases in Interpretation: Confirmation bias, Overgeneralization, Correlation fallacy, Automation bias
- "Bias" can be Good, Bad, Neutral

https://www.bu.edu/articles/2014/bu-research-riddle-reveals-the-depth-of-gender-bias

From
http://web.stanford.edu/class/cs224n/slides/Vinodkumar_Prabhakaran_Socially_Responsible_NLP.pdf

Collaborative language modeling

PEER: A Collaborative Language Model. Schick et al. 2022.

Link to Twitter [GIF]

Learning Map

- Key components for understanding text

- NLP systems/applications
  - Current techniques and limitations

- Build realistic NLP tools

What's not covered by this course

- Speech recognition, signal processing
- Natural language generation (one lecture)
- Text mining/Information retrieval (CS589)
- Programming details

This lecture

- ❑ Course Information
- ❑ Course Overview
  - ❑ What is NLP? Why is it important?
  - ❑ What will you learn from this course?
- ❑ **<u>What are the challenges?</u>**
- ❑ Key NLP components

Why is NLP hard?

- **Complexity** in representing, learning, and using linguistic/situational/world/visual knowledge
- Human languages are **ambiguous** (unlike programming and other formal languages)
- Human language interpretation depends on real world, **common sense**, and **contextual** knowledge.

Challenges - Ambiguity

- Word sense ambiguity: semantic ambiguity, when a word has more than one meanings.
  - *John killed the wolf*
  - *Bill killed the project*
  - *Mary killed Jane (at tennis or murdered her?)*

credit: A. Zwicky

Challenges - Ambiguity

- Word sense/meaning ambiguity

Challenges - Ambiguity

❖ Prepositional Phrase (PP) attachment ambiguity

San Jose cops kill man with knife

Close

Text Paper Translate Listen

San Jose cops kill man with knife

Ex-college football player, 23, shot 9 times allegedly charged police at fiancee's home

By Hamed Aleaziz and Vivian Ho

A man fatally shot by San Jose police officers while allegedly charging at them with a knife was a 23-year-old former football player at De Anza College in Cupertino who was distraught and depressed, his family said Thursday.

Police officials said two officers opened fire Wednesday afternoon on Phillip Watkins outside his fiancee's home because they feared for their lives. The officers had been drawn to the home, officials said, by a 911 call reporting an armed home invasion that, it turned out, had been made by Watkins himself.

But the mother of Watkins' fiancee, who also lives in the home on the 1300 block of Sherman Street, said she witnessed the shooting and described it as excessive. Faye Buchanan said the confrontation happened shortly after she called a suicide intervention hotline in hopes of getting Watkins medical help.

Watkins' 911 call came in at 5:01 p.m., said Sgt. Heather Randol, a San Jose police spokeswoman. "The caller stated there was a male breaking into his home armed with a knife," Randol said. "The caller also stated he was locked in an upstairs bedroom with his children and requested help from police."

She said Watkins was on the sidewalk in front of the home when two officers got there. He was holding a knife with a 4-inch blade and ran toward the officers in a threatening manner, Randol said.

"Both officers ordered the suspect to stop and drop the knife," Randol said. "The suspect continued to charge the officers with the knife in his hand. Both officers, fearing for their safety and defense of their life, fired at the suspect."

On the police radio, one officer said, "We have a male with a knife. He's walking toward us."

"Shots fired! Shots fired!" an officer said moments later.

A short time later, an officer reported, "Male is down. Knife's still in hand."

Buchanan said she had been prompted to call the

Shoot continues on D8

Back Continue

Is the prepositional phrase attached high, modifying the verb "kill"?

Or is it attached low, to the object NP "man"?

Challenges - Ambiguity

Ambiguous headlines

- Include your children when baking cookies----------
  (in the activity of baking? or inside the cookies?)
- Local High School Dropouts Cut in Half----------
  (dropout students? or dropout rates?)
- Hospitals are Sued by 7 Foot Doctors----------
  (7 foot tall? Or 7 doctors?)
- Iraqi Head Seeks Arms----------
  (human arms? or weapon arms?)
- Safety Experts Say School Bus Passengers Should Be Belted----------
  (safety/seat belted? Or strike with a belt?)
- Teacher Strikes Idle Kids----------
  (go on strikes? Or hit idle kids?)

Challenges - Ambiguity

Pronoun reference ambiguity

Dr. Macklin often brings his dog Champion to visit with the patients. He just loves to give big, wet, sloppy kisses!

Challenges - Language is dynamic

Language evolves and changes, e.g. cyber lingo

LOL	Laugh out loud
G2G	Got to go
BFN	Bye for now
B4N	Bye for now
Idk	I don’t know
FWIW	For what it’s worth
LUWAMH	Love you with all my heart

Challenges - Scale

Bible (King James version)	~700K words
Penn Tree bank	~4.5M words from wall street journal
Newswire collection	~500M+ words
Wikipedia	~3 billion words
Web	several billions of words

This lecture

- ❑ Course Information
- ❑ Course Overview
  - ❑ What is NLP? Why is it important?
  - ❑ What will you learn from this course?
- ❑ What are the challenges?
- ❑ **<u>Key NLP components</u>**

Part of speech tagging

Common closed-classes:

- prepositions: on, under, over, near, by, at, from, to, with
- particles: up, down, on, off, in, out, at, by
- determiners: a, an, the
- conjunctions: and, but, or, as, if, when
- pronouns: she, who, I, others
- auxiliary verbs: can, may, should, are
- numerals: one, two, three, first, second, third

Common open-classes:

- adjectives
- adverbs
- nouns
- verbs (except auxiliary verbs)
- interjections

Chapter 8: https://web.stanford.edu/~jurafsky/slp3/8.pdf

Syntactic (Constituency) tagging

Syntactic structure -> meaning

Syntactic categories

Lexical and phrasal categories together are called syntactic categories:

Dependency parsing

Dependencies are asymmetrical relations between two lexical items (words):

Hearing –-ATT → A [*A* is an attribute of *hearing*]
Is – SBJ → hearing [*hearing* is a subject of *is*]

Semantic analysis

- Word sense disambiguation
- Semantic role labeling

Co-reference resolution

Beyonce Giselle Knowles-Carter (born September 4, 1981) is an American singer, songwriter, record producer and actress. Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in the late 1990s as lead singer of R&B girl-group Destiny's Child. Managed by her father, Mathew Knowles, the group became one of the world's best-selling girl groups of all time. Their hiatus saw the release of Beyonce's debut album, Dangerously in Love (2003), which established her as a solo artist worldwide, earned five Grammy Awards and featured the Billboard Hot 100 number-one singles "Crazy in Love and "Baby Boy"

Q: Is Beyonce's father Mathew Knowles?

Learning Representations

Quiz 0
and
Break

Linear Algebra Review

- s = $\mathbf{a}^T\mathbf{b}$ where a and b are both column vectors with dimension of d.
  - what is $\frac{\partial s}{\partial \mathbf{a}}$ and what is the dimension of $\frac{\partial s}{\partial \mathbf{a}}$?

- **y** = **Ax** where **y** is a mx1 column vector, **A** is a mxn matrix, **x** is a nx1 column vector, **A** does not depend on **x**.
  - what is $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$?

- s = $\mathbf{y}^T\mathbf{A}\mathbf{x}$ where **y** is a mx1 column vector, **A** is a mxn matrix, **x** is a nx1 column vector, **A** and **y** do not depend on **x**.
  - what is $\frac{\partial s}{\partial \mathbf{x}}$?

Reference: https://en.wikipedia.org/wiki/Matrix_calculus

Python Review

Why Python?

+ Python is a widely used, general purpose programming language.

+ Easy to start working with.

+ Scientific computation functionality similar to Matlab and Octave.

+ Used by major deep learning frameworks such as PyTorch and TensorFlow.

Language basics

Python is a strongly-typed and dynamically-typed language.

- Strongly-typed: Interpreter always “respects” the types of each variable.[1]
- Dynamically-typed: “A variable is simply a value bound to a name.” [1]
- Execution: Python is first interpreted into bytecode (.pyc) and then compiled by a VM implementation into machine instructions. (Most commonly using C.)
  - Python is “slower”, but it can run highly optimized C/C++ subroutines which make scientific computing (e.g. matrix multiplication) really fast.

https://wiki.python.org/moin/Why%20is%20Python%20a%20dynamic%20language%20and%20also%20a%20strongly%20typed%20language

Language basics

- Strongly-typed: `1+'1' -> Error`

- Dynamically-typed: `foo = [1,2,3]` and later, `foo = "hello"`

- Execution: `np.dot(x, W) + b` -> fast!

Python Review

Language Basics

Anyone wants to guess what this function does?

```
def function(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) // 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]
    return function(left) + middle + function(right)
```

Example code from Andrej Karpathy's tutorial: http://cs231n.github.io/python-numpy-tutorial/

Python Review - Common operations

```
x = 10                          # Declaring two integer variables
y = 3                           # Comments starts with the hash symbol

x + y >> 13                     # addition
x - y >> 7                      # subtraction
x ** y >> 1000                  # exponentiation
x/y >> 3                        # dividing two integers
x / float(y) >> 3.333..         # type casting for float division
str(x) +"+" + str(y) >>"10+3"   # casting and string concatenation
```

Example code from Andrej Karpathy's tutorial: http://cs231n.github.io/python-numpy-tutorial/

Build-in values

```
True, False              # usual boolean values
None                     # represents the null

x = None                 # variables can be None
array=[1,2,None]         # lists can contain None

def func():
        return None      # functions can return None

if [1,2]!=[3,4]:         # can check for equality
        print 'error!'
```

List

```
names = ['zach', 'emily']

names[0] == 'zach'

names.append('alex')

len(names) == 3

print names >> ['zach','emily','alex']

names.extend(['aba','kevin'])

print names >> ['zach','emily','alex','aba','kevin']

names = [] # creates an empty list

names = list() # also creates an empty list

stuff=[1,['hello','bye'], 1.3, None] # can mix types
```

List slicing

List elements can be accessed in convenient ways

basic format: mylist[start_index:end_index]

```
numbers = [0,1,2,3,4,5,6]
numbers[0:3] == numbers[:3] >> [0,1,2]
numbers[5:] == numbers[5:7] >>[5,6]

numbers[-1] == 6 #negative index wraps around
number[-3:] == [4,5,6]
numbers[3:-2] == [3,4] # can mix and match
```

Tuple

Tuple are immutable arrays

```
names = ('zach', 'emily')
names[0] == 'zach'
len(names) == 2
print names >> ('zach', 'emily')
names[0] = 'richard'
>>typeerror:  'tuple' object does not support item assignment

empty = tuple() # empty tuple
single = (10,) # single-element tuple, comma matters!
```

Dictionary

Dictionaries are hash maps

```
phonebook = dict()                          #empty dictionary
phonebook = {'zach': '12-68'}               # dictionary with one item
phonebook['emily'] = '35-65'                #add another item
print('zach' in phonebook)                  >> True
print('john' in phonebook)                  >>False
del phonebook['zach']                       #delete an item
print (phonebook)                           >> {'emily':'36-65'}
for name, number in phonebook.iteritems():
        print name, number                  >> emily 35-66
```

Classes

```
class Animal(object):
    def __init__(self, species, age): # Constructor `a = Animal('bird', 10)`
        self.species = species            # Refer to instance with `self`
        self.age = age                    # All instance variables are public

    def isPerson(self):
            return self.species == "Homo Sapiens"
    def ageOneYear(self):
            self.age += 1
class Dog(Animal):                        # Inherits Animal's methods
    def ageOneYear(self):                 # Override for dog years
        self.age += 7
```

Numpy - ndarray operations

```
import numpy as np

z=np.array([[6,7],[8,9]])


print(np.max(z, axis=1))       >> [7,9]

print(np.max(z, axis=1, keepdims=True))  >> [[7][9]]

np.dot(x, W) or x.dot(W)
```

Note: shapes (N,) !=(1,N)

```
print (np.array([1,2,3]).T)            >> [1  2  3]

np.sum(np.array([1,2,3]), axis = 1) >> Error!
```

Indexing

```
x = np.random.random((3, 4))

x[:]

x[np.array([0, 2]), :]

x[1, 1:3]


x[x > 0.5]
```

Indexing

```
x = np.random.random((3, 4))   # Random (3,4) matrix
x[:]                           # Selects everything in x
x[np.array([0, 2]), :]         # Selects the 0th and 2nd rows
x[1, 1:3]                      # Selects 1st row as 1-D vector
                               # and 1st through 2nd elements
x[x > 0.5]                     # Boolean indexing
```

Broadcasting

```
x = np.random.random((3, 4))

y = np.random.random((3, 1))

z = np.random.random((1, 4))

x + y

x * z

print((y + y.T).shape)
```

Broadcasting

```
x = np.random.random((3, 4))  # Random (3, 4) matrix
y = np.random.random((3, 1))  # Random (3, 1) matrix
z = np.random.random((1, 4))  # Random (1,4) vector
x + y                         # Adds y to each column of x
x * z                   # Multiplies z element-wise with each row of x
print((y + y.T).shape)        # Can give unexpected results!
```

Efficiency

For loops will dramatically slow down your code (~10-100x)

```
for i in range(x.shape[0]):
    for j in range(x.shape[1]):
            x[i,j] **= 2
x **=2
for i in range(100, 1000):
    for j in range(x.shape[1]):
            x[i, j] += 5
x[np.arange(100,1000), :] += 5
```

List comprehension

- Similar to map() from functional programming languages.
- Can improve readability & make the code succinct.
- Format: `[func(x) for x in some_list]`
- Following are equivalent:

```
squares = []

for i in range(10):

    squares.append(i**2)

squares = [i**2 for i in range(10)]
```

- Can be conditional:

```
odds = [i**2 for i in range(10) if i%2 == 1]
```

Convenient syntax

- Multiple assignment / unpacking iterables

```
x, y, z = ['Tensorflow', 'PyTorch', 'Chainer']
age, name, pets = 20, 'Joy', ['cat']
```

- Returning multiple items from a function

```
def some_func():
    return 10, 1
ten, one = some_func()
```

- Joining list of strings with a delimiter

```
", ".join(['1', '2', '3']) == '1, 2, 3'
```

- String literals with both single and double quotes

```
message = 'I like "single" quotes.'
reply = "I prefer 'double' quotes."
```

Debugging tips

- Python has an interactive shell where you can execute arbitrary code
  - Great replacement for TI-84 (no integer overflow!)
  - Confused by syntax? Just try it in the shell!

```
$ python

Python 2.7.10 (default, Jul 15 2017, 17:16:57)

>>> 2 ** 5 / 2

16

>>> 2 ** (5 / 2)

4
```

  - Can import any module (even custom ones in the current directory)
  - Try small test cases in the shell

Debugging tips

Unsure of what you can do with an object?**Use type() and dir()!!**

```
>>> class Duck(object):
 ...     def quack(self): pass
 ...
 >>> bird = Duck()
 >>> type(bird)
 <class '__main__.Duck'>
 >>> dir(bird)
 ['__class__', '__delattr__', '__dict__', '__doc__', '__format__',
 '__getattribute__', '__hash__', '__init__', '__module__',
 '__new__', '__reduce__', '__reduce_ex__', '__repr__',
 '__setattr__', '__sizeof__',
 '__str__', '__subclasshook__', '__weakref__', 'quack'] >>>
```

Numpy debugging

- Print shapes to see if they match what you expect: print x.shape
- Print shapes!! Make sure broadcasting is done properly.
- Print types and values.
- Checking if two float arrays are approximately equal (element-wise)
  - `np.allclose(x, y) # Can also specify tolerance`
- Checking if an array is close to zero (e.g. gradient)
  - `np.allclose(x, 0) # Broadcasting`
- Selecting all elements less than 0 from an array
  - `x[x < 0] # Returns 1-dim array`

Environment management

- Problem:
  - Python 3 is not backward-compatible with Python 2
  - Countless Python packages and their dependencies
  - Different projects require different packages
    - Even worse, different versions of the same package!

- Solution:
  - Keep multiple Python environments that are isolated from each other
  - Each environment…
    - can use different Python versions
    - keeps its own set of packages
    - can be easily replicated (e.g. on a VM, friend's laptop, etc.)

Anaconda

- Anaconda is a popular Python environment/package manager
  - Install from https://www.anaconda.com/download/
  - Supports Windows, Linux, macOS
  - Basic workflow

```
$ source activate <environment_name>

    <... do stuff ...>

    $ deactivate
```

  - Other environments won't be affected by anything you do
  - Allows you to run a different version of Python for each environment

Virtualenv

- Virtualenv is another popular Python environment manager
  - Only specifies different packages per environment
  - Doesn't help run different Python version
  - Installation from
    - https://virtualenv.pypa.io/en/stable/installation/
  - Basic workflow

```
$ mkdir <environment_directory>

$ virtualenv <environment_directory>

$ source <env_dir>/bin/activate

$ pip install <package>
```

Other Python Resources

1. Official Python 2 documentation: https://docs.python.org/2/
2. Official Python 2 tutorial: https://docs.python.org/2.7/tutorial/index.html
3. Numpy Quickstart: https://docs.scipy.org/doc/numpy-dev/user/quickstart.html
4. Python Tutorial from CS231N at Stanford: http://cs231n.github.io/python-numpy-tutorial/
5. Stanford Python course (CS41): http://stanfordpython.com/

Acknowledgements

Slides adapted from Dr. Christopher Manning's Natural Language Processing with Deep Learning at Stanford.

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

stevens.edu

Thank You