STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

CS 584 Natural Language Processing

Introduction

Ping Wang
Department of Computer Science
Stevens Institute of Technology

This lecture

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

Course Information

- **Meeting**:
  - **Time**: Monday, 3:00-5:30PM
  - **Location**: Burchard 103
- **Canvas**:
  - Link: https://sit.instructure.com/courses/68054
  - Slides uploaded before each lecture
  - Announcements, Assignments, Discussions.
  - Login to myStevens

Instructor and CA Information

- Instructor: Ping Wang
  - Office hours: Tuesday, 11AM-12PM
  - Zoom link: https://stevens.zoom.us/j/97789938972
  - Email: ping.wang@stevens.edu
- Course Assistant:
  - Bharath Beeravelly (bbeerave@stevens.edu) Office hours: Wednesday, 1PM-2PM, Zoom: https://stevens.zoom.us/j/94713156470
  - Chetna Agarwal (cagarwal@stevens.edu) Office hours: Friday, 1PM-2PM. Zoom: https://stevens.zoom.us/j/93947263945, passcode: 251118

Prerequisites

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

Self-Introduction

What do we learn from this course?

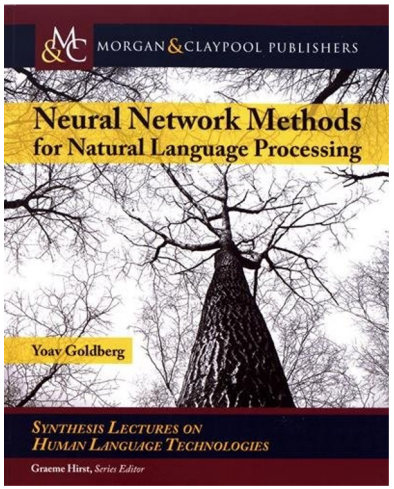
1. Understanding of basic techniques of *processing* human languages and the difficulties in *understanding* languages.
2. Understanding of effective modern methods for NLP
   - recurrent neural networks, attention, convolutional neural networks, etc
3. Understanding of and ability to build systems for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

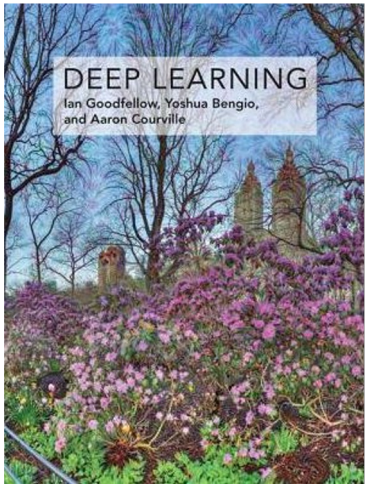
Course Outcomes

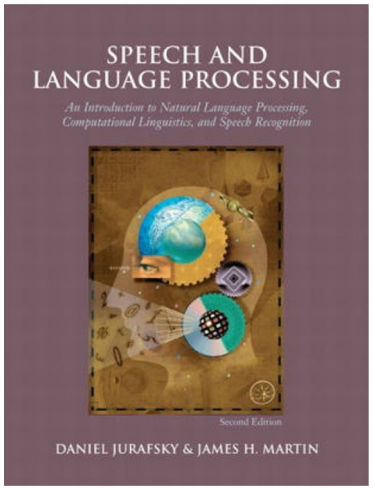
1. Implement gradient descent (GD) and stochastic gradient descent (SGD) techniques for learning problems and understand the theory behind them.
2. Apply word2vec models in real-world text corpora.
3. Understand the neural networks models and backpropagation optimization. Implement neural network models 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.
7. Understand the recent advances in NLP, such as transformers, pre-training, fine-tuning, prompting, etc.

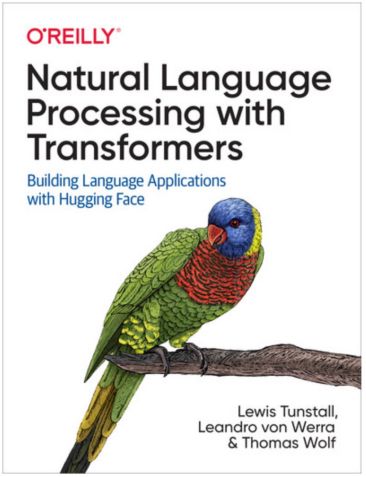
Recommended Textbooks

- 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)**.
- Lewis Tunstall, Leandro von Werra, and Thomas Wolf. **Natural Language Processing with Transformers (NLPT), revised edition**, O'Reilly Media, Inc., 2022.









Course work and grading policy

- Assignments (30%): 4 in total
- Midterm exam (30%): 11/20/2023, in person
- Course project (35%):
  - Project proposal (5%)
  - Project midterm report (10%)
  - Final report and code (15%)
  - Presentation (5%)
- Participation (5%):
  - Lecture attendance (random 3-4 lectures)
  - Canvas discussion

Assignments

- **Goal:** test your ability to understand knowledge of NLP
- Mix of written and programming problems.
- Submission: Upload your e-copies (PDF file and codes files) on Canvas
- Jupyter Notebook is recommended to use for your programming assignments (with code and output after execution). Make sure to include detailed comments and analysis in your code.

Course project

- Project topic
- Team: can be a group project
- Requirements
- More details will be provided in the project guidelines.

Grading

Distribution of (0-100):

- A (90-100)
- A- (85-90)
- B+ (80-85)
- B (75-80)
- B- (70-75)
- C+ (65-70)
- C (60-65)
- F (<60)

Late Submission Policy

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

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

Use generative AI technologies with care

- You may use to help generate ideas and brainstorm
  - But material generated by these programs may be inaccurate, incomplete, or otherwise problematic.
  - May also stifle your own independent thinking and creativity.
- If you include material generated by an AI program, it should be **cited like any other reference material** (with due consideration for the quality of the reference, which may be poor).
- Any plagiarism or other form of cheating will be dealt with under relevant Stevens policies.

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

Stevens Honor System

DO NOT share course material outside the classroom

COPYRIGHT

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

Emergency Resources

Stevens Campus Police
Kidde Building, Ground Floor
201-216-3911 (Emergency Line, 24/7)
201-216-5105 (Non-Emergency Line, 24/7)

National Suicide Prevention Lifeline
1-800-273-8255

Crisis Textline
Text HOME to 741-741 (24/7)

Please save these numbers in your phone.

Emergency Communication

Sign up or update your information in Stevens Alerts, the emergency alert system used at Stevens. Be in the know about snow days and campus emergencies.

Be sure to provide:

- Your Stevens email
- Your personal email
- Your cell phone number

Go to MyStevens; click on 'Stevens Alerts' to register.

Report a Concern

If you have a concern about another student that is ***NOT time sensitive***, use the '*Report a Concern*' link on MyStevens or email care@stevens.edu to inform a team of professionals who can assist.

Wellness Resources

Counseling and Psychological Services (CAPS)
Wellness Center, 2nd Floor; 201-216-5177
Free personal and group counseling.
Call for an appointment.
Counselors are available 24/7.

Student Health Services
Wellness Center, 1st Floor; 201-216-5678
Call in advance to be seen by a clinician.
A nurse is available 24/7.

Disability Services
Wellness Center, 2nd Floor
Assists students with disabilities to fully participate in campus services and programs with equal access.

Acknowledgements

- Course Textbooks
- Dr. Yue Ning's NLP at Stevens
- Dr. Christopher Manning's NLP with Deep Learning at Stanford
- Dr. Greg Durrett's NLP at UT Austin
- Other online courses and materials

Acknowledgements

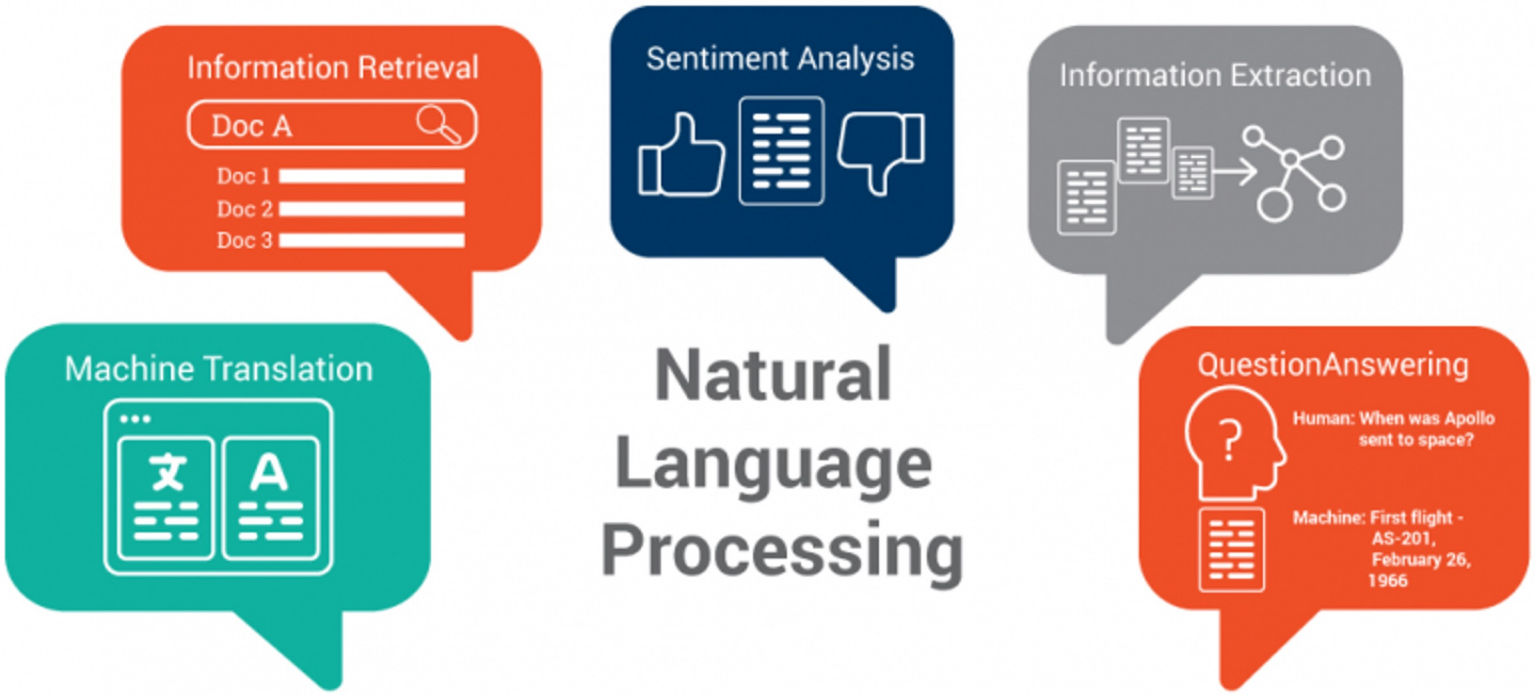
Slides adapted from Dr. Christopher Manning's Natural Language Processing with Deep Learning at Stanford.

This lecture

- ❑ Course Information
- ❑ **Course Overview**
  - ❑ 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.



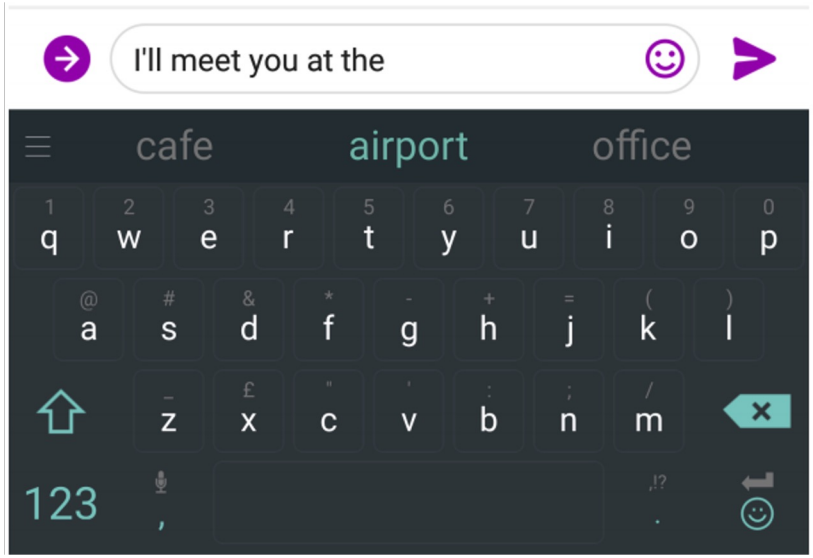
Go beyond the keyword matching

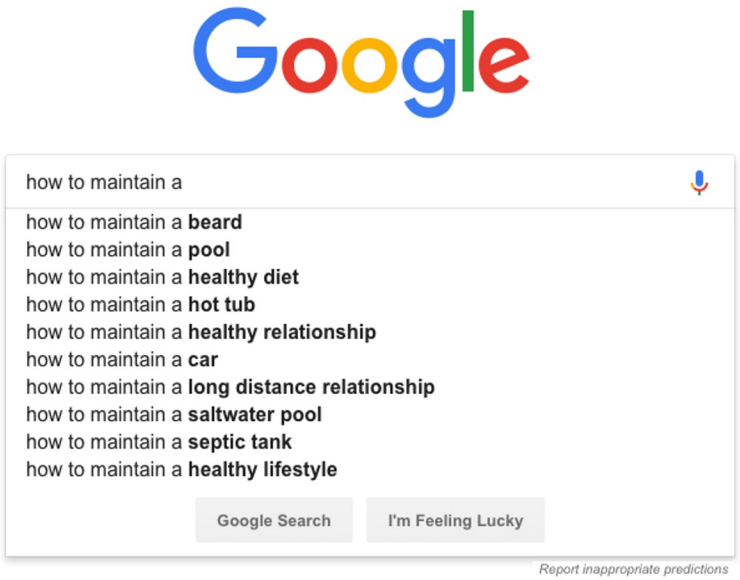
- Identify the structure and meaning of *words, sentences, texts and conversations*
- Deep understanding of broad language
- NLP is all around us

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.

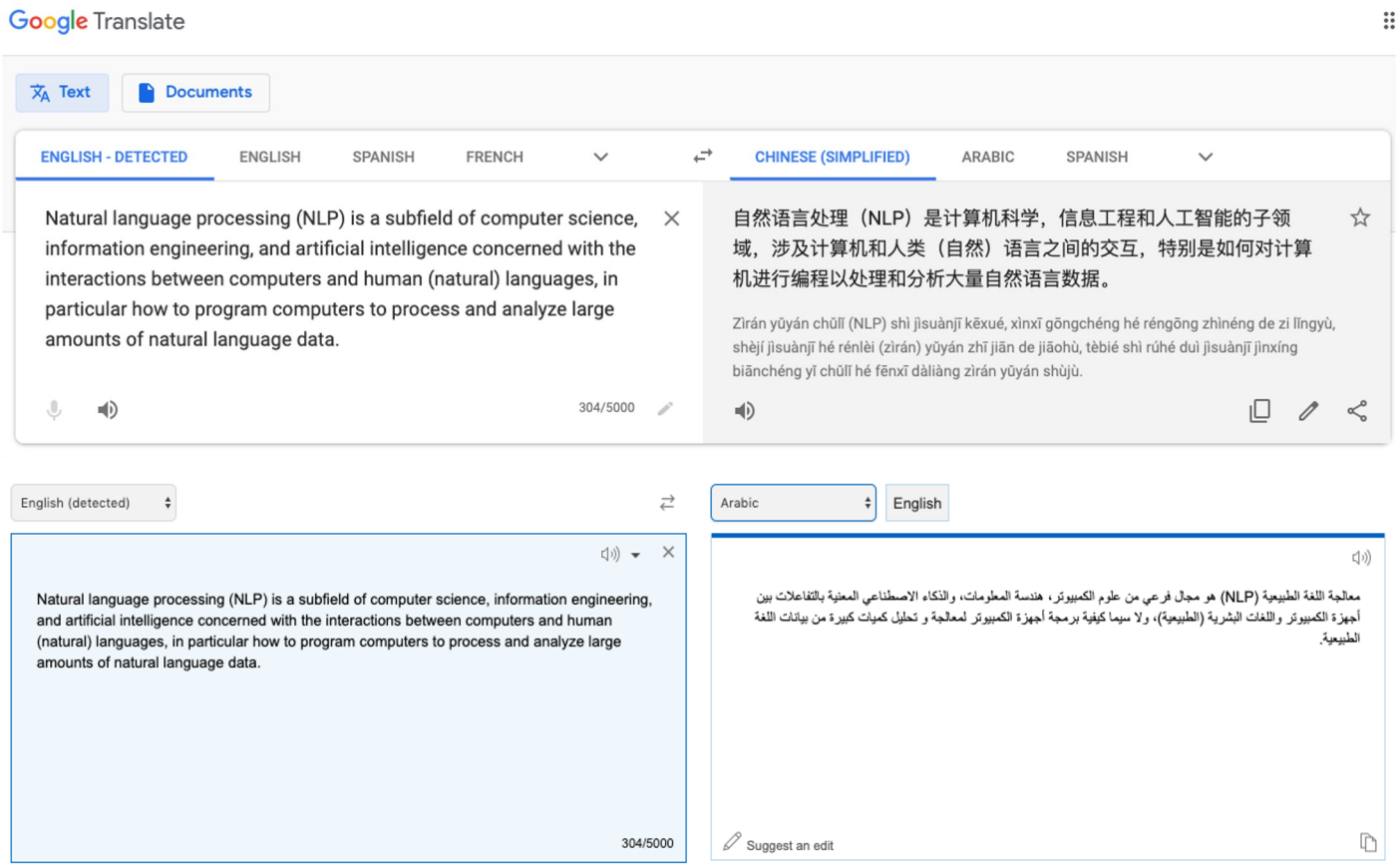
Predictive Text





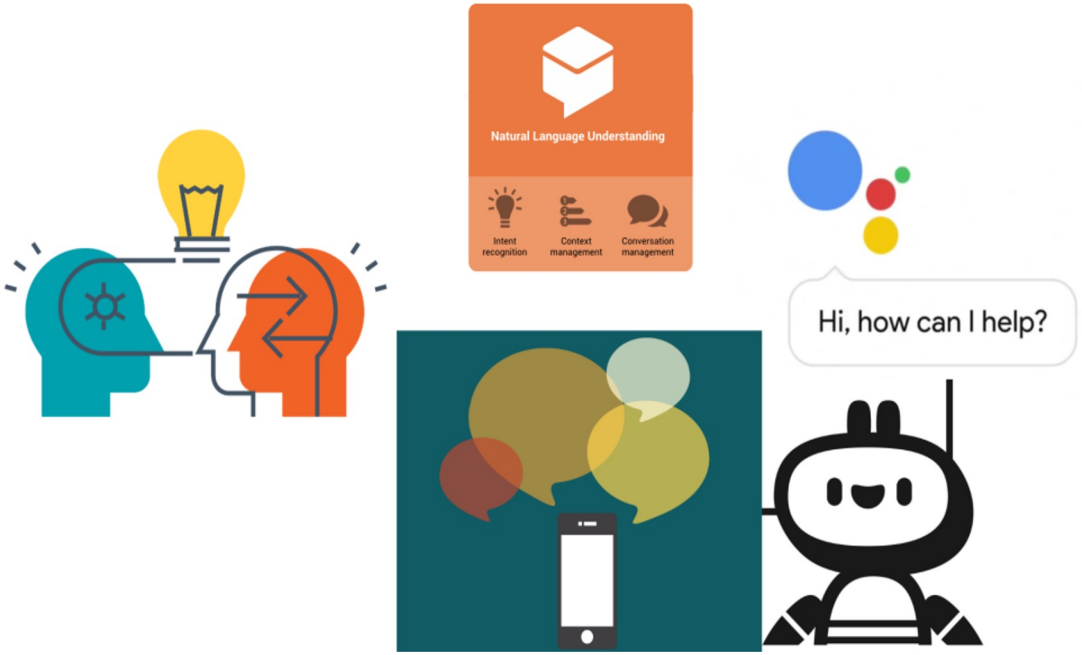
- Typing Assistant. Varun Kathuria. 2017.
- Bayesian Recurrent Neural Networks. Fortunato et al. 2017.
- Contextual LSTM (CLSTM) models for Large scale NLP tasks. Ghosh et al. 2016.

Machine Translation



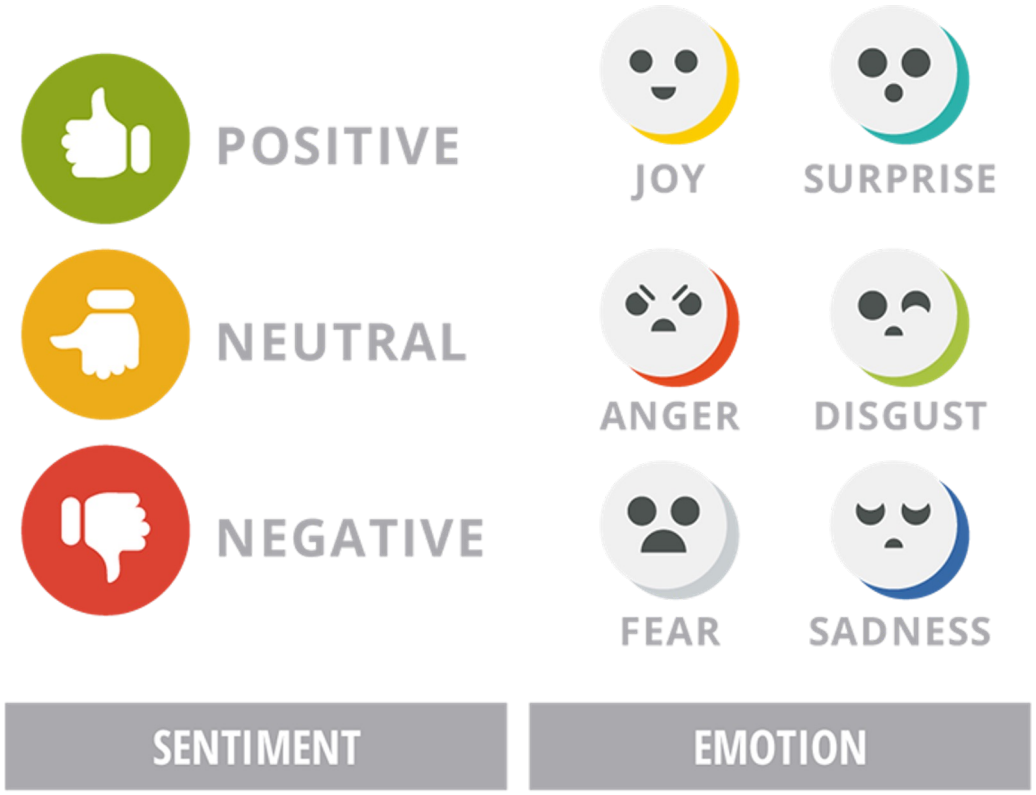
- Language Models are Few-Shot Learners (GPT-3). Brown et al. 2020.

Dialog Systems



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

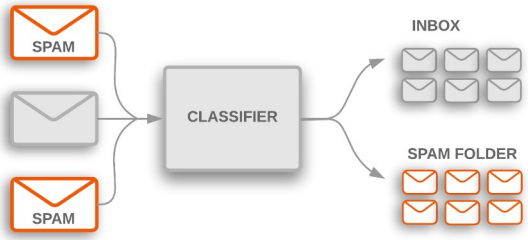
Sentiment/Opinion Analysis



- 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

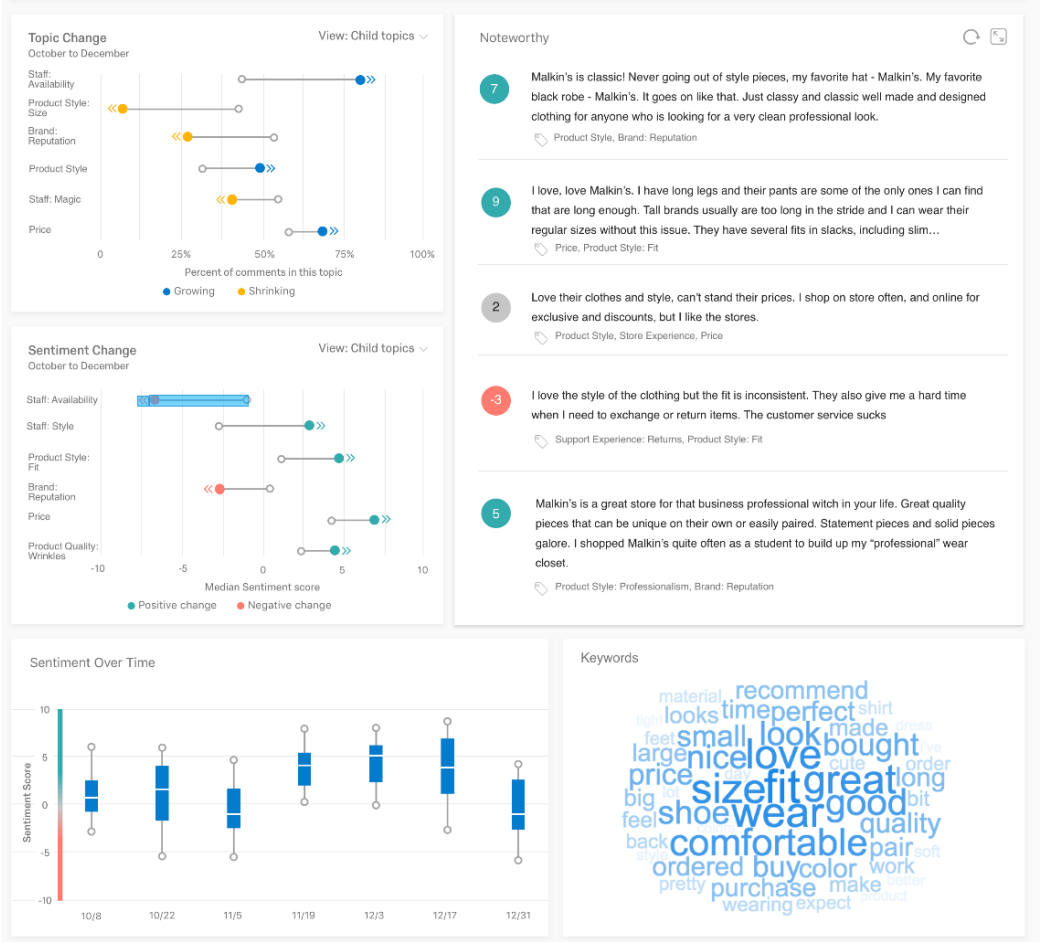
- Binary classification (e.g., spam or not, positive or negative)



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

- Large Scale Hierarchical Text Classification (Kaggle competition)

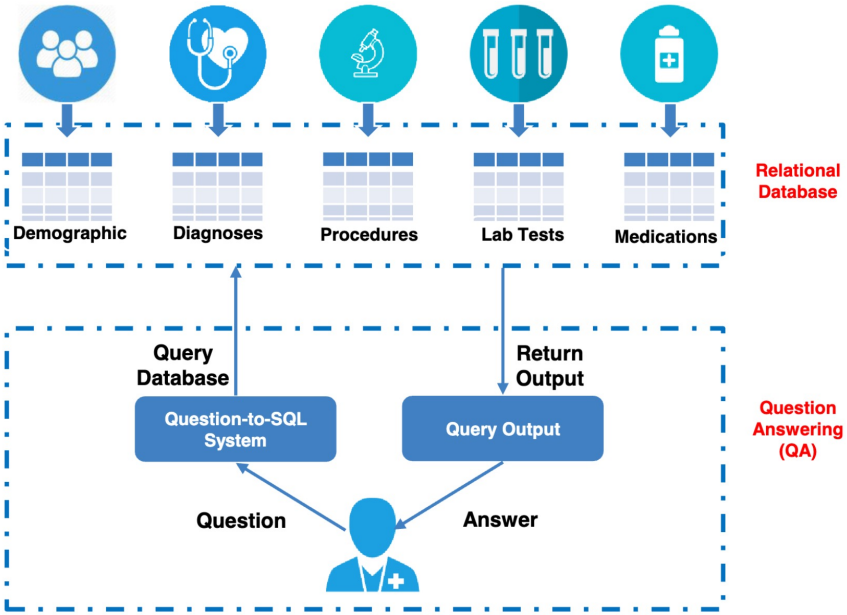
Survey Analysis



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

Image from [Text iQ](#) by qualtrics

Question Answering



Text-to-SQL generation for QA

- 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

Use **natural language** to provide instructions or guidance to machines, computers, robots, or other automated systems. Instead of **using formal programming languages or complex commands**, NLI allows users to communicate with technology in a more intuitive and human-friendly manner.

- 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

Challenges:

Understanding the context of the instruction, dealing with ambiguities in language, and ensuring accurate interpretation

Potential:

Making technology more accessible and user-friendly, reducing the barrier for non-technical users to interact with and control sophisticated systems.

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

Related tasks: 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

Gender bias is one of the most prominent issues in NLP. Some researchers discovered that many advanced word-embedding language models, such as GPT-3, tend to connect males with occupations that require higher level of education.

- When the model is asked “what is the gender of a doctor?”, it would have much higher possibility to respond, “It’s male.”
- In contrast, if the model is asked “what is the gender of a nurse?”, it would be more likely to respond, “It’s female.”

- Furthermore, other researchers realized that when they translated “He is a nurse. She is a doctor.” to Hungarian and then translated it back to English, the sentence became “She is a nurse. He is a doctor.”

This evidence clearly demonstrates how gender bias has truly appeared in NLP systems.

Source link

Collaborative language modeling

Instead of generate only the final texts, the collaborative language modeling allows to modify and refine the generated texts to obtain more accurate and high-quality final texts.

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

Link to Twitter [GIF]

ChatGPT: Optimizing Language Models for Dialogue. OpenAI. Nov. 2022.

Learning Map

- Key components for understanding text

- NLP systems/applications
  - Current techniques and limitations

- Build realistic NLP tools

This lecture

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

Challenges in NLP

- **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

  - Get you an ambulance
  - or tell you you are an ambulance

Challenges - Ambiguity

- Prepositional Phrase (PP) attachment ambiguity

San Jose cops kill man with knife

Text Paper Translate Listen Close

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.

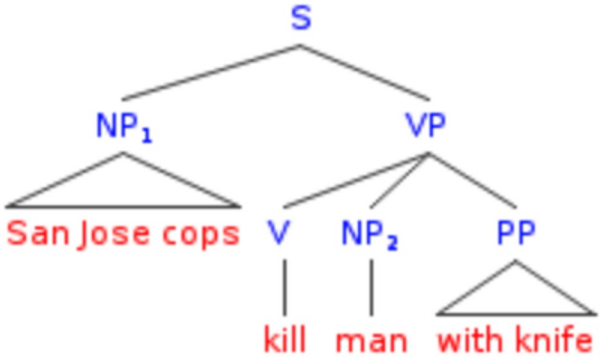
A short time later, an officer reported, "Male is down. Knife's still in hand."

Buchanan said she had been prompted to call the

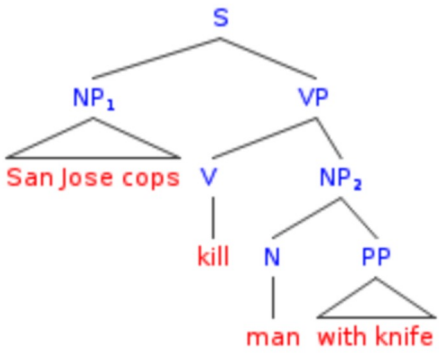
Shoot continues on D8

Back Continue

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



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



Challenges - Ambiguity

Headlines Don't Always Tell the Story

Ambiguous headlines:

- Include your children when baking cookies

  (in the activity of baking? or inside the cookies?)

- Hospitals are Sued by 7 Foot Doctors
  (7 foot tall? Or 7 doctors?)

- Eye drops off shelf
  (eye drops are taken off the shelf or an eye drops off a shelf)

- Teacher Strikes Idle Kids
  (go on strikes? Or hit idle kids?)

Challenges - Ambiguity

Pronoun reference ambiguity

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

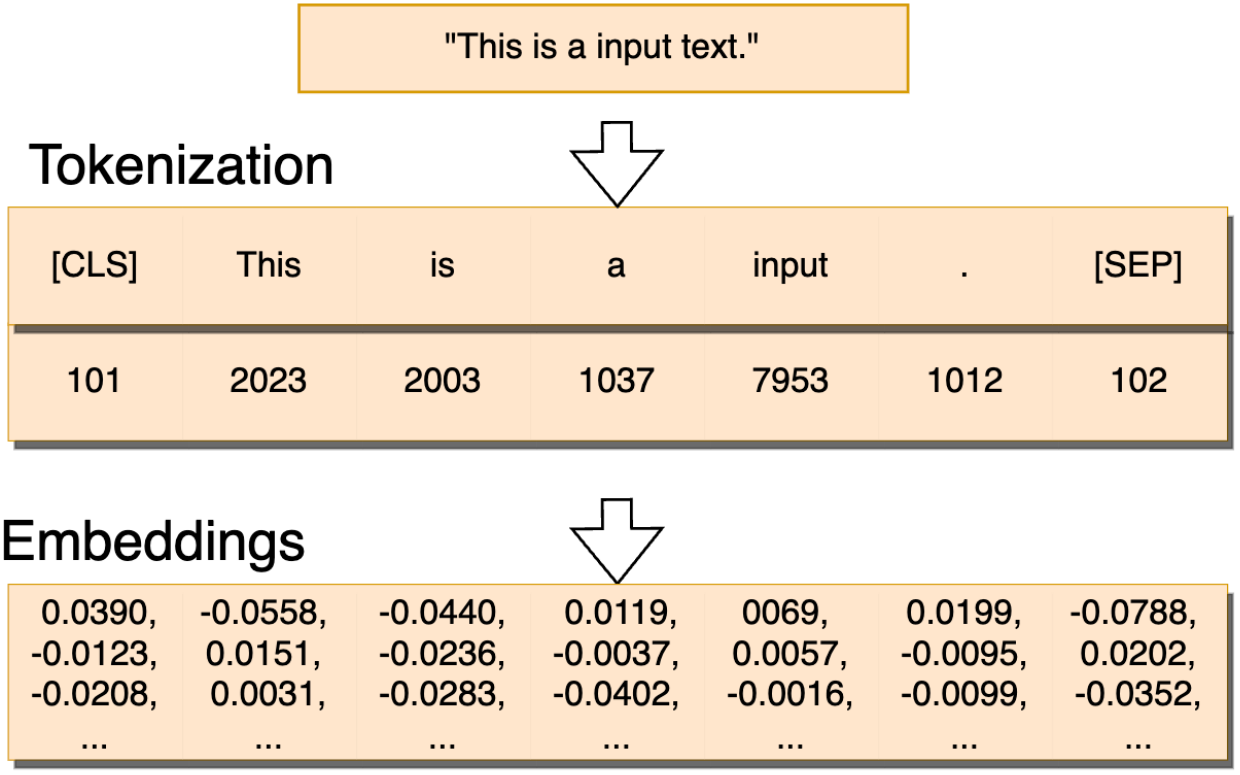
Bible (King James version)	~700K words
Penn Tree bank	~4.5M words from wall street journal
Newswire collection	~500M+ words
Wikipedia	~3 billion words
Web	several billions of words

This lecture

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

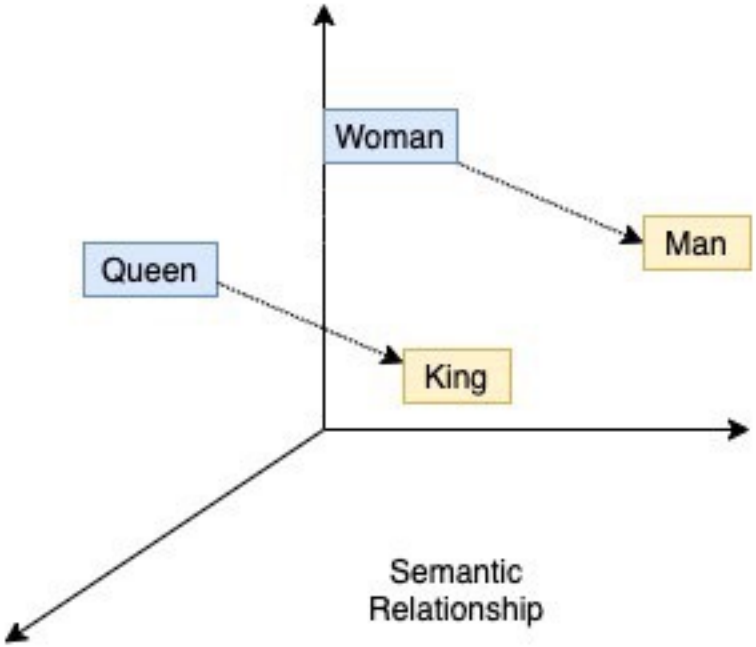
Tokenization in NLP

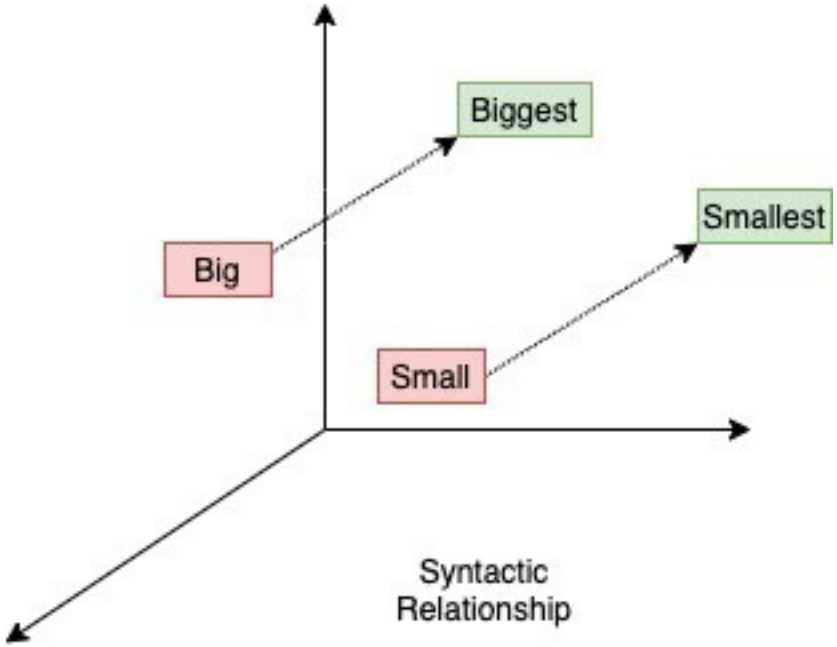
Tokenization is cutting input data into parts (symbols) that can be mapped (embedded) into a vector space.



https://vaclavkosar.com/ml/Tokenization-in-Machine-Learning-Explained

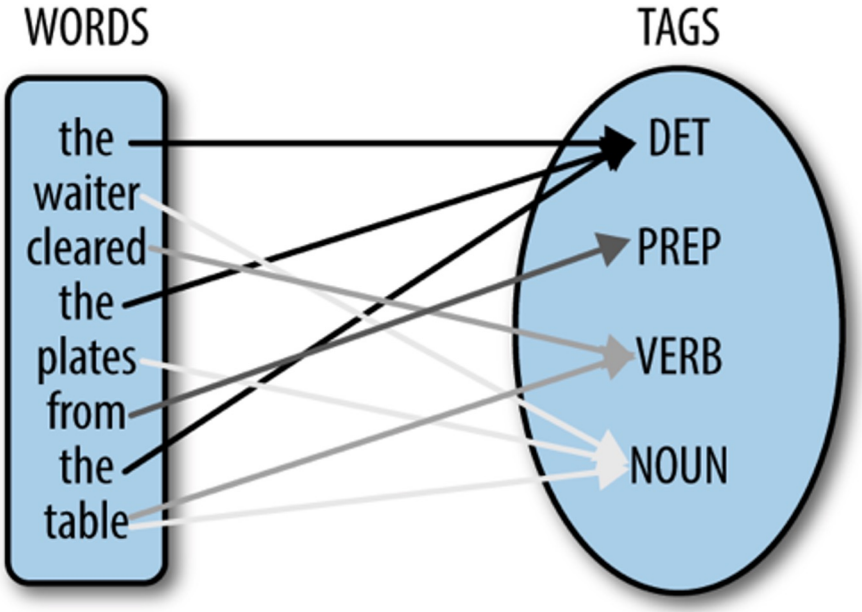
Learning Representations





Part of speech tagging

Assigning grammatical parts of speech (e.g., noun, verb, adjective) to each word in a sentence.



Common closed-classes:

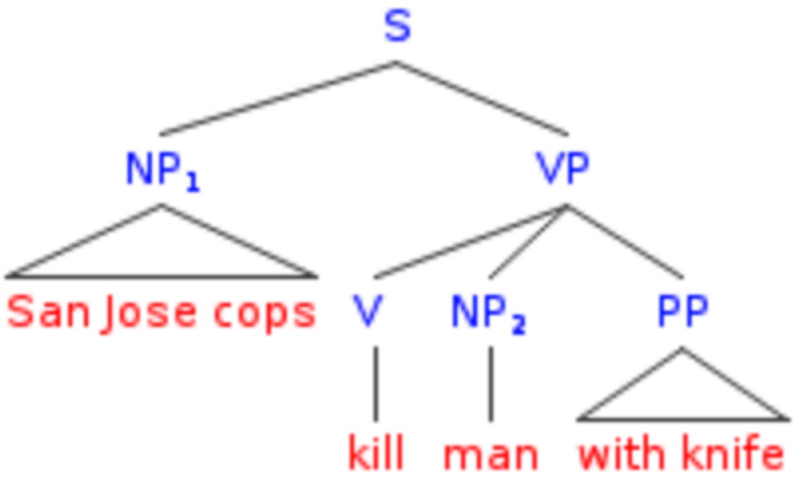
- prepositions: on, under, over, near, by, at, from, to, with
- particles: up, down, on, off, in, out, at, by
- determiners: a, an, the
- conjunctions: and, but, or, as, if, when
- pronouns: she, who, I, others
- auxiliary verbs: can, may, should, are
- numerals: one, two, three, first, second, third

Common open-classes:

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

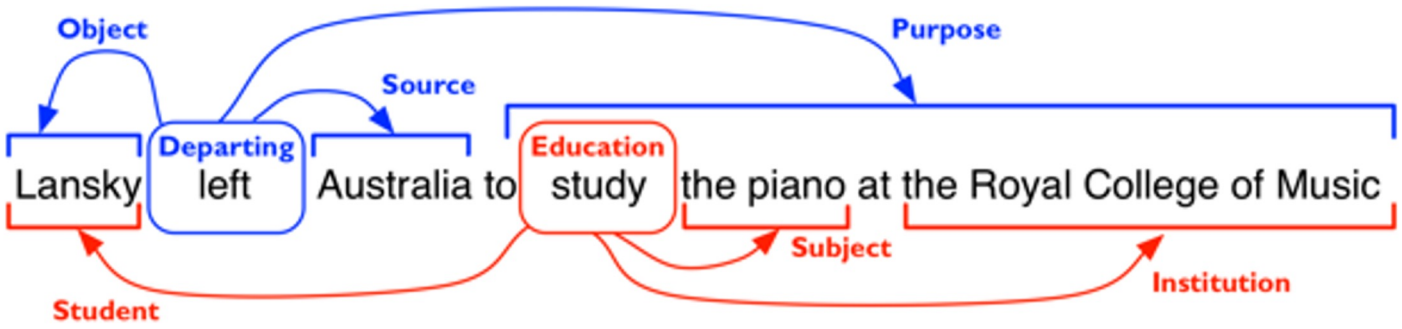
Chapter 8: https://web.stanford.edu/~jurafsky/slp3/8.pdf

Syntactic tagging



Semantic analysis

- Word sense disambiguation
- Semantic role labeling



Named Entity Recognition

RECORD #961115

895226877 | YMC | 33406917 | | 2170940 | 9/12/2005 12:00:00 AM | Upper Respiratory Infection | | DIS | Admission Date: 6/27/2005 Report Status:

Discharge Date: 1/26/2005

****** DISCHARGE ORDERS ******

DISCHARGE MEDICATIONS:

IBUPROFEN Medication | 600-800 MG Dosage | PO Mode | TID PRN Frequency | Pain Reason

Food/Drug Interaction Instruction Take with food

DIET: No Restrictions

albuterol Medication | prn Frequency | wheezing Reason | while in hospital Duration but did not affect patient's sob. Neg D-dimer but due to acute onset SOB and concominant Right lower extremity pain , LENI's done which were negative for DVT. Patient felt better with IVF Medication given in ER and did not take any nebulizers while on the floor.

2.HTN: BP well controlled , pt has never taken HCTZ or any other Rx HTN

Co-reference resolution

Finding all expressions that refer to the same entity in a text.

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?

Readings

- CH1-3 SLP
- CH1 NNLP

STEVENS

INSTITUTE of TECHNOLOGY

THE INNOVATION UNIVERSITY®

stevens.edu

Thank You

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 - 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: axis 1 is out of bounds
for array of dimension 1
```

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))  # 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/