



NEW YORK SMART.
WORLD-CLASS READY.®

Baruch College | Zicklin School of Business

ZICKLIN BUSINESS

YEARS

Applied Natural Language Processing

CIS/STA 9665

Chaoqun Deng

Assistant Professor in IS

Outline

- Introduction to the Class
 - Getting to know me
 - Getting to know you
 - Introduction to syllabus
- Introduction to NLP
 - What is NLP?
 - NLP Applications
 - Machine Learning for NLP
 - Why is NLP hard?
- Review of Python Programming
- Review of Machine Learning Methods

Outline

- Introduction to the Class
 - Getting to know me
 - Getting to know you
 - Introduction to syllabus
- Introduction to NLP
 - What is NLP?
 - NLP Applications
 - Machine Learning for NLP
 - Why is NLP hard?
- Review of Python Programming
- Review of Machine Learning Methods

About Me

- Ph.D. in Information Systems at Rensselaer Polytechnic Institute, New York



Rensselaer

- Tenure-Track Assistant Professor in IS at University of Missouri



- Tenure-Track Assistant Professor in IS at Baruch College, CUNY

Baruch
COLLEGE CUNY

Teaching and Research

- Teaching Experience
 - Applied Natural Language Processing
 - Data Mining with Business Analytics
 - Introduction to Information Systems
 - Big Data and Hadoop
 - Supply Chain Management
 - IT Security
- Research Area
 - Big Data
 - Machine Learning (Deep Learning) and NLP
 - Sharing Economy

Big Data

Q1. Which of the followings do you think is/are Big Data?

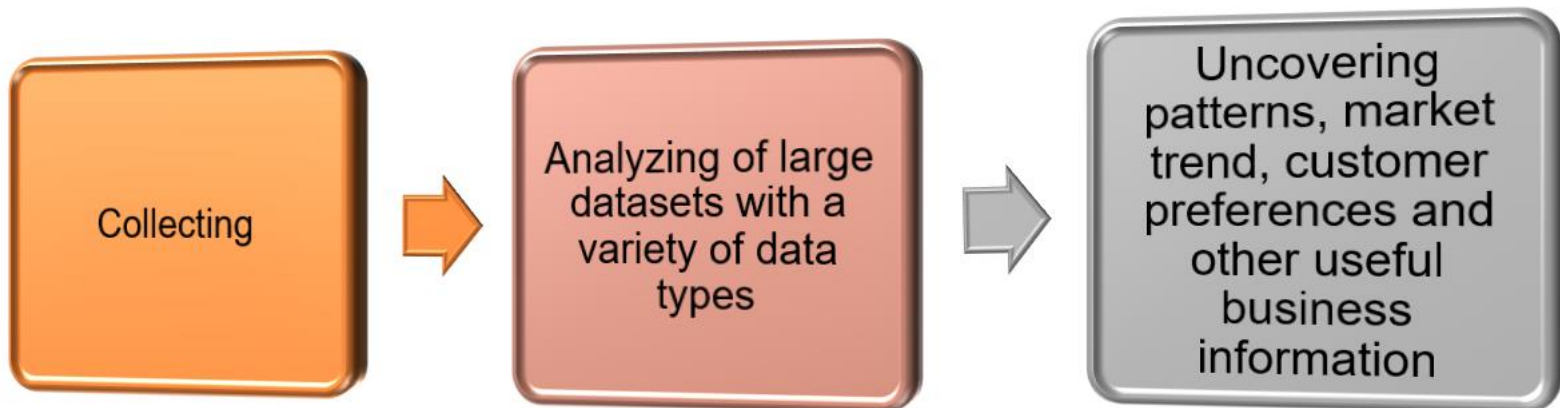
- ☐ Order Details For A Store
- ☒ All Orders Across 100s of Stores
- ☐ A Person's Stock Portfolio
- ☒ All Stock Transactions for the NYSE

- Data that's too big to be processed on a single machine.
- Big data is data that exceeds the processing capacity of conventional database system.
- The data is too big, moves too fast, and does not fit the structures of your database architecture.

Big Data



Big Data Analytics



- Unstructured Data

- Machine-generated data:

- Satellite images
 - Photos and Videos



- Human-generated data:

- Social Media Data
 - Mobile data



- Semi-Structured Data

- XML and JSON



- Structured Data

- Data contained in relational databases and spreadsheets


 Rocio M
 3 2

5.0 Reviewed 3 weeks ago

EXPENSIVE AND POOR SERVICE.

We had booked a double room but could only get us a king and claimed to have a roll away bed that was queen size, but turned out to be twin. That same day as we made our way to dinner we frustrated people in... [More](#)

2 Thank Rocio M

Response from hiltontimes, Manager at Hilton Times Square
 Responded 2 weeks ago

Dear Rocio M, I appreciate your review of your recent experience at our hotel. I'm very sorry you had a disappointing experience with us, and I addressed these issues with our staff. Exceptional guest service is the foundation of our brand promise. Your feedback helps... [More](#)

4.5 9/5/2017

Food was fresh and tasty. I ordered the crab entree and my friend ordered the lobster- we ended up splitting our meals.

Only down fall was it was foggy so our window "view seating" had no view.

This restaurant is Fisherman Warf area so great place given the busy location. Service was great too.

Was this review ...?

Useful

Funny

Cool

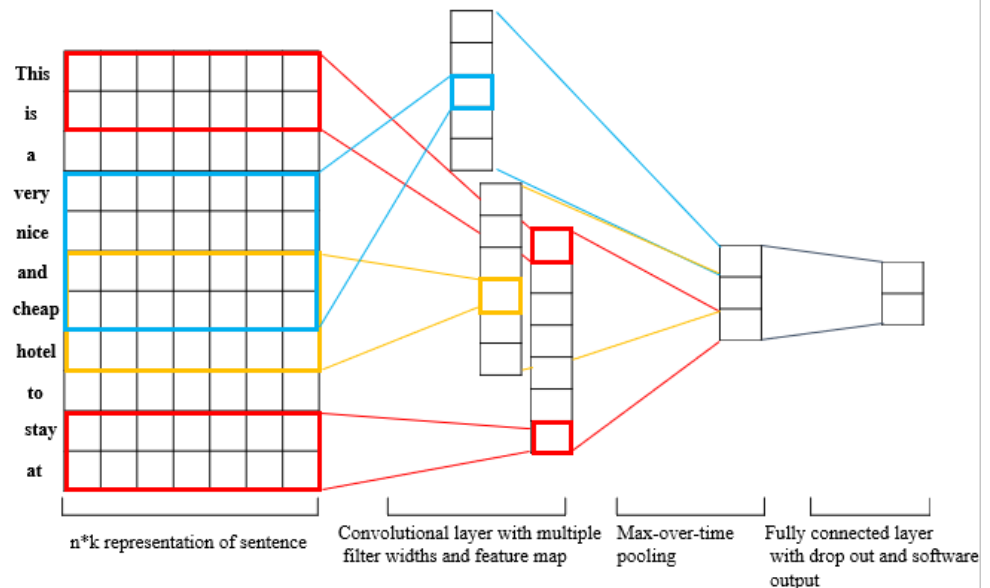


Comment from Bob P. of Fog Harbor Fish House
 Business Manager

9/7/2017 - Danielle

Thank you for the review and for choosing Fog Harbor. I'm thrilled to hear you enjoyed a... [Read more](#)

Convolutional Neural Network-NLP (Natural Language Processing)

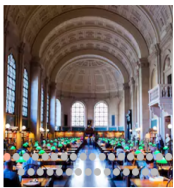




Providers Build Trust in Customers:



How Customers Build Trust in Providers?



Hey, I'm Joe!


Somerville, Massachusetts, United States · Joined in July 2014

[Report this user](#)

1712 Reviews


Reviews (1712)

Reviews From Guests



Staying at Joe's was excellent. The room was exactly as described online. Joe was great at communicating and made our stay as smooth as possible

From Chicago, IL · May 2017 · [✉](#)



The motel 6 of Airbnb. Would not recommend.

From Provincetown, MA · April 2017 · [✉](#)

Verified info


Government ID ☒

Email address ☒

Phone number ☒

[Learn more >](#)

About me



Hey, I'm Roo, Nadine And Neil!

New York, New York, United States · Joined in June 2015

[Report this user](#)

Roo: I'm a yoga teacher, writer, nanny, and bartender living in Brooklyn. Prior to that I lived in Seattle for several years, London for a little bit, and I grew up in a ski resort in Idaho. I travel as much as I can and like to move often. I smile a lot. I'm learning French(slowly). My favorite color is green. I love early 20th-century lit, film noir, and 70s rock and roll.

Neil: I am a Jamaican transplant calling NYC my home for over 10 years now. I spent most of that time in Manhattan but recently moved to Brooklyn with my lovely friend Roo. Music is my passion especially reggae. I love to laugh with and learn from new people - laughing especially! :)

Verified info

Email address ☒

Phone number ☒

[Learn more >](#)


About me

Languages
English

371 Reviews


Reviews (371)

Reviews From Guests



Nici!

From New York, NY · May 2017 · [✉](#)



Joseph

We felt very lucky to find this place on short notice at a decent price. Definitely no frills, but if like us you're just looking for a place to catch a night's rest, it does the trick. One caveat - if you have a bad back, you may find the mattress less than comfortable; I wish I had slept on the futon instead!

From Boston, MA · April 2017 · [✉](#)

Sharing economy or “crowd-based capitalism” is a new way of organizing economic activities. Driven by the Internet and mobile technology, it is a new market model that provides peer-to-peer sharing of access to goods and services (Zhang et al. 2016).

Contact Information

- Email: chaoqun.deng@baruch.cuny.edu
- Office Hour: 9:00-10:00 pm Wednesday
- Zoom meetings

Getting to know each other

- Your name
- Major/level
- Hobby
- Hometown
- Something unique/fun about your hometown
- Share your classmate's information

Read the Syllabus

- 5 Minutes

Course Material

- Textbook
 - Natural Language Processing with Python—
Analyzing Text with the Natural Language Toolkit
(<https://www.nltk.org/book/>)

What will we cover during the course?


- Explore interesting bodies of text using tiny Python programs (Chapters 1-3)
- Structured programming (Chapter 4) that consolidates the programming topics used in NLP
- Fundamental topics in language processing: tagging, classification, and information extraction (Chapters 5-7)
- Parse a sentence, recognize its syntactic structure (Chapters 8)

What you expect of me?

- Be available for extra help and to answer extra questions
- Provide a lesson plan that is practical. Create an environment that allows active learning
- Show up everyday in a good mood and ready to teach
- Provide clear communication of expectations for assignments and how the exams will be structured
- Make the class's content as relevant as possible

What I expect of you?

- Show up for class on time, actively participate in class exercises, lab sessions, and do required assignments and projects.
- Understand what you are reading, listening, watching, etc., and ask questions if you don't asap!
- Collaboration and Coordination in Teamwork
- Complete all the assigned work within time.



NEW YORK SMART. WORLD-CLASS READY.®



**NO LATE
ASSIGNMENTS
ALLOWED**

Academic Integrity

- If you are copying someone else's work (assignment, projects, homework, exams), you will go through academic sanctions.
- Academic sanctions in this class will range from an F on the assignment to an F in this course.
- A report of suspected academic dishonesty will be sent to the Office of the Dean of Students.
- Additional information and definitions can be found at:
http://www.baruch.cuny.edu/academic/academic_honesty.html

Zoom Meetings

- All the class meetings are synchronously
- Come to the class (zoom meetings) on time
- Q and A
 - Unmute yourself
 - Chat
 - I will answer the questions at the end of each section
 - Raise Hand
 - Office hour: after the class 9 pm-10 pm
- Non verbal feedback
 - Participates



- Annotation (viewing option)
 - Draw->arrow

Recording of Remote Classes:

- Students who participate in this class with their camera on or use a profile image are agreeing to have their video or image recorded solely for the purpose of creating a record for students enrolled in the class to refer to, including those enrolled **students who are unable to attend live.**
- If you are unwilling to consent to have your profile or video image recorded, be sure to keep your camera off and do not use a profile image.
- Likewise, students who un-mute during class and participate orally are agreeing to have their voices recorded.
- If you are not willing to consent to have your voice recorded during class, you will need to keep your mute button activated and communicate exclusively using the "chat" feature, which allows students to type questions and comments live.

Review Policy

- Feel free to ask me why you received a certain grade on an assignment/exam/project within one week after the grade is posted. If you received a grade in error I will correct it. If not, you cannot redo the assignment/project/exam to earn the credit. After one week, your score will stand.

Evaluation Criteria

Assignments	30 pts	5 Assignments* 6 pts	30 pts
Exams	30 pts	Midterm Exam	15 pts
		Final Exam	15 pts
Term Project	30 pts	Project Proposal	5 pts
		Final Presentation	5 pts
		Final Report	15 pts
		Peer Evaluation	5 pts
In-Class Lab Reports	10 pts	8 Labs* 1.25 pts	10 pts
Total	100 pts		100 pts

Term Project

- Team Formation Due –**Sep 7**
 - Self-organize or Randomly assign?
 - Each team should have a similar number of students(4-5 students)
 - Each team should **nominate one contact person (team leader)** for the instructor no later than Sep 07, 2022.
- Project proposal presentation Due-**Oct 26**
- Project proposal Due- **Oct 28**
- Final project presentation- **Dec 7**
- Peer Evaluation Due **Dec 11**
- Final project report-Due **Dec 11**

NEW YORK SMART. WORLD-CLASS READY.®



Q and A

Outline

- Introduction to the Class
 - Getting to know me
 - Getting to know you
 - Introduction to syllabus
- Introduction to NLP
 - What is NLP?
 - NLP Applications
 - Machine Learning for NLP
 - Why is NLP hard?
- Review of Python Programming
- Review of Machine Learning Methods

Introduction to NLP

- What is NLP?
- NLP Applications
- Machine Learning for NLP
- Why is NLP hard?

What is NLP?

- It'd be great if machines could
 - Process our email (usefully)
 - Translate languages accurately
 - Help us manage, summarize, and aggregate information
 - Use speech as a UI (when needed)
 - Talk to us / listen to us
- But it is difficult to realize:
 - Language is complex, ambiguous, flexible, and subtle
 - Good solutions need linguistics and machine learning knowledge

What is NLP?

- What is now difficult for computers (and any other species) to do is effortless for humans



- Natural language processing (NLP)** is a subfield of [linguistics](#), [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-from wikipedia

What is NLP?

- Fundamental goal: deep understand of broad language
 - Not just string processing or keyword matching!



What is NLP?

- Computers use (analyze, understand, generate) natural language
- Text Processing
 - Lexical: tokenization, part of speech, lemmas
 - Parsing and chunking
 - Semantic tagging: semantic role, word sense
 - Certain expressions: named entities
 - Discourse: coreference, discourse segments
- Speech Processing
 - Phonetic transcription
 - Segmentation (punctuations)
 - Prosody

Why is important?

- An enormous amount of knowledge is now available in machine readable form as natural language text
- Conversational agents are becoming an important form of human-computer communication
- Much of human-human communication is now mediated by computers

What is going on?

- Yahoo, Google, Microsoft → Information Retrieval
- Monster.com, HotJobs.com (Job finders) → Information Extraction + Information Retrieval
- Google Translate, Bing Translator → Machine Translation
- Yahoo! Answers → Question Answering
- Myspace, Facebook, Blogspot → Processing of User-Generated Content
- All “Big Guys” have (several) strong NLP research labs:
 - IBM, Microsoft, AT&T, Xerox, Sun, etc.
- Academia: research in a university environment



NEW YORK SMART. WORLD-CLASS READY.®



GOOGLE AI HAS FULL CONVERSATIONS

can I help?

Make me a haircut appointment on Tuesday
morning anytime between 10 and 12.

TECH
INSIDER

NLP Applications

- Simple Applications
 - Word counters
 - Spell checkers, grammar checkers
 - Predictive text on mobile handsets and gmail

Thanks, I'll take a look.

This is great, thank you!

Thanks, I will take a look.

Bigger Application

- Text classification
- Text summarization
- Information retrieval
- Information extraction
- Machine translation
- Speech recognition
- Question answering
- Conversational agent

Google Translate

Chinese - detected ▼



↔



English ▼

我正在学习自然语言处理
Wǒ zhèngzài xuéxí zìrán yǔyán chǔlǐ

×

I am learning natural language processing

Question Answering

- IBM Watson competed on Jeopardy! and won the first place prize!



IBM Watson

- Watson was created as a question answering (QA) computing system that IBM built to apply advanced natural language processing, information retrieval, knowledge representation, automated reasoning, and machine learning technologies to the field of open domain question answering-wikipedia



How Does Watson Fit in?

Systems that think like humans

“The exciting new effort to make computers think... machines with minds, in the full and literal sense.” (Haugeland, 1985)

“[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning...” (Bellman, 1978)

Systems that act like humans

“The art of creating machines that perform functions that require intelligence when performed by people” (Kurzweil, 1990)

“The study of how to make computers do things at which, at the moment, people are better (Rich and Knight, 1991)

Systems that think rationally

“The study of mental faculties through the use of computational models.” (Charniak and McDermott, 1985)

“The study of the computations that make it possible to perceive, reason, and act.”
(Winston, 1972)

Systems that act rationally

“The branch of computer science that is concerned with the automation of intelligent behavior.” (Luger and Stubblefield, 1993)

“Computational intelligence is the study of the design of intelligent agents.”
(Poole et al., 1998)

“AI... is concerned with intelligent behavior in artifacts.” (Nilsson, 1998)

Other Popular NLP tasks

- Data-mining of Weblogs, discussion forums, message boards, user groups, tweets, and other forms of user generated media
 - Product marketing information
 - Political opinion tracking
 - Social network analysis
 - Buzz analysis (what's hot, what topics are people talking about right now)
 - Customer opinions and how firms respond

Machine Learning for NLP

- Machine Learning
 - The field of study that gives computers the ability to learn without being explicitly programmed.
 - It explores the study and construction of algorithms that can learn from and make predictions on data.

Learning Methods for NLP

- Supervised: identify hidden units (concepts) of explicit units
 - Syntactic analysis, word sense disambiguation, name classification
 - Text classification (e.g. Naïve Bayes, Decision Trees etc)
 - Trained from labeled data
- Unsupervised: identify relationships and properties of explicit units (terms, docs)
 - Association detection, topicality (e.g. LDA), similarity (e.g. LDA), clustering
 - Without labeled data
- Semi-supervised: Combinations
 - Combines a small amount of labeled data with a large amount of unlabeled data during training

Deep Learning for NLP

- Recurrent neural networks
- Long short-term memory
- Convolutional neural networks
- Transformers

Recurrent Neural Networks

- Language is inherently sequential
- Thus, a model that can progressively read an input text from one end to another can be very useful for language understanding
- Recurrent neural networks (RNNs) are specially designed to keep such sequential process and learning in mind
- RNNs have neural units that are capable of remembering what they have processed so far
- This memory is temporal and the information is stored and updated with every time step as the RNN reads the next word in the input
- Useful for text classification, name entity recognition and machine translation etc.

Long short-term memory

- RNNs suffer from the problem of forgetful memory—they cannot remember longer contexts and therefore do not perform well when the input text is long
- LSTMs, a type of RNN, circumvent this problem by letting go of the irrelevant context and only remembering the part of the context that is needed to solve the task at hand
- This relieves the load of remembering very long context in one vector representation

Convolutional Neural Network

- CNNs are very popular and used heavily in computer vision tasks like image classification, video recognition and have also seen success in NLP, especially in text classification tasks.
- The main advantage of CNNs have is their ability to look at a group of words together using a context window
- For example, we are doing sentiment classification, “I like this movie very much”
- In order to make sense of this sentence, it is better to look at words and different sets of contiguous words
- CNNs can do exactly this by definitions of their architecture

Transformers

- It models the textual context but not in a sequential manner
- Given a word in the input, it prefers to look at all the words around it and represent each word with respect to its context
- E.g. the word “bank” can have different meanings depending on the context in which it appears
- If the context talks about finance, then “bank” probably denotes a financial institution
- On the other hand, if the context mentions river, then it probably indicates a bank of the river

Why NLP is hard?

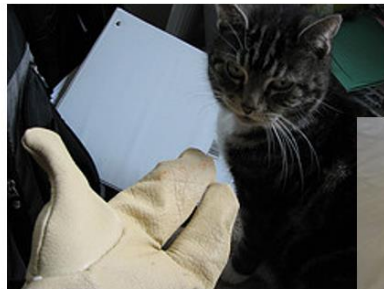
- Ambiguity
- Common Knowledge
- Creativity
- Diversity Across languages

Ambiguity

- Computational linguists are obsessed with ambiguity
- Ambiguity is a fundamental problem of computational linguistics
- Resolving ambiguity is a crucial goal

Ambiguity

- How can a machine understand these differences?
- SENTENCE: Get the cat with the gloves.



Common Knowledge

- A key aspect of any human language is “common knowledge”
- It is the set of all facts that most humans are aware of
- It is assumed that these facts are known, hence they are not explicitly mentioned, but they do have a bearing on the meaning of the sentence

Common Knowledge

- “Man bit dog” or “dog bit man”
- Humans use common knowledge to understand and process any language
- But a computer would find it very difficult to differentiate between the two, as it lacks the common knowledge humans have.
- One of the key challenges in NLP is how to encode all the things that are common knowledge to humans in a computational model

Creativity

- Language is not just rule driven, there is also a creative aspect to it.
- Various styles, dialects, genres, and variation are used in any language
- Poems are a great example of creativity in language
- Making machines understand creativity is a hard problem not just in NLP, but in AI in general

Diversity Across Languages

- For most languages in the world, there is no direct mapping between the vocabularies of any two languages
- This makes porting an NLP solution from one language to another hard
- A solution that works for one language might not work at all for another
- This means that one either builds a solution that is language agnostic or that one needs to build separate solutions for each language
- While the first one is conceptually very hard, the other is laborious and time intensive

NEW YORK SMART. WORLD-CLASS READY.®



Q and A

Outline

- Introduction to the Class
 - Getting to know me
 - Getting to know you
 - Introduction to syllabus
- Introduction to NLP
 - What is NLP?
 - NLP Applications
 - Key NLP Components
 - Machine Learning for NLP
 - Why is NLP hard?
- Review of Python Programming
- Review of Machine Learning

Install Python and Jupyter

- Download Anaconda
- Launch Jupyter Notebook
- Launch Spider (run Python)
- New-Python 3

NEW YORK SMART. WORLD-CLASS READY.®



Data science technology for a better world.

Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine. Start working with thousands of open-source packages and libraries today.

Download 

For Windows

Python 3.9 • 64-Bit Graphical Installer • 594 MB

[Get Additional Installers](#)

Anaconda Installers

Windows

Python 3.9

64-Bit Graphical Installer (594 MB)

32-Bit Graphical Installer (488 MB)

MacOS

Python 3.9

64-Bit Graphical Installer (591 MB)

64-Bit Command Line Installer (584 MB)

64-Bit (M1) Graphical Installer (316 MB)

64-Bit (M1) Command Line Installer (305 MB)

Linux

Python 3.9

64-Bit (x86) Installer (659 MB)

64-Bit (Power8 and Power9) Installer (367 MB)

64-Bit (AWS Graviton2 / ARM64) Installer (568 MB)

64-bit (Linux on IBM Z & LinuxONE) Installer



▼ today (4)

○ Anaconda3 2021.05 (64-bit) Setup



Installing

Please wait while Anaconda3 2021.05 (64-bit) is being installed.

Setting up the base environment ...



Show details

Anaconda, Inc.

< Back

Next >

Cancel

Launch Jupyter

- Establish a new folder (your destination folder)
- Desktop-Jupyter
- New->python3 → click “Untitled”-change it to “Chapter 1

Jupyter Shortcut

- A to insert a new cell above the current cell
- B to insert a new cell below.
- D + D (press the key twice) to delete the current cell
- Shift+Enter: run the cell
- Enter: continue to write in the current cell

Review of Python Programming

- An Introduction to Python
 - Numbers, Strings, Lists
- More Control Flow Tools
 - If statements
 - for statements
 - Range() function
 - Break and continue statements on loops
- Data Structures
 - List Comprehensions
 - Tuples
 - Sets
 - Dictionaries

1. Using Python as a Calculator

- 1.1 Numbers

```
9/5 # division always returns a floating point numbers
```

```
1.8
```

```
19//3 # floor division discards the fractional part
```

```
6
```

```
19%3 # % operator returns the remainder of the division
```

```
1
```

```
5**3 # the ** operator to calculate powers
```

```
125
```

```
# The equal sign (=) is used to assign a value to a variable. Afterwards, no result is displayed before the next interactive prompt
```

```
width = 20  
height = 5 * 9  
width * height
```

```
900
```

1.2 Strings

- Python can also manipulate strings, which can be expressed in several ways. They can be enclosed in single quotes ('...') or double quotes ("...") with the same result. \ can be used to escape quotes:

```
'spam eggs'
```

```
'spam eggs'
```

```
'doesn\'t' # use \' to escape the single quote...
```

```
"doesn't"
```

```
"doesn't" # ...or use double quotes instead
```

```
"doesn't"
```




- The `print()` function produces a more readable output, by omitting the enclosing quotes and by printing escaped and special characters
- If you don't want characters prefaced by `\` to be interpreted as special characters, you can use raw strings by adding an `r` before the first quote

```
print('C:\some\name')  # here \n means newline!
```

```
C:\some  
ame
```

```
print(r'C:\some\name')  # note the r before the quote
```

```
C:\some\name
```

Strings can be concatenated (glued together) with the `+` operator, and repeated with `*`:

```
# 3 times 'un', followed by 'ium'
```

```
3 * 'un' + 'ium'
```

```
'unununium'
```

Two or more string literals (i.e. the ones enclosed between quotes) next to each other are automatically concatenated. This feature is particularly useful when you want to break long strings:

```
'Py' 'thon'
```

```
'Python'
```

```
# This feature is particularly useful when you want to break long strings:
```

```
text = ('Put several strings within parentheses '  
        'to have them joined together.')
```

```
text
```

```
'Put several strings within parentheses to have them joined together.'
```

Strings can be **indexed** (subscripted), with the first character having index 0. There is no separate character type; a character is simply a string of size one:

```
word = 'Python'
```

```
word[0] # character in position 0
```

```
'P'
```

```
word[5] # character in position 5
```

```
'n'
```

```
# Indices may also be negative numbers, to start counting from the right:
```

```
word[-1] # Last character
```

```
'n'
```

```
word[-2] # second-last character
```

```
'o'
```

```
word[-6]
```

```
'P'
```

While indexing is used to obtain individual characters, slicing allows you to obtain substring:

```
word[0:2] # characters from position 0 (included) to 2 (excluded)
```

```
'Py'
```

```
word[2:5] # characters from position 2 (included) to 5 (excluded)
```

```
'tho'
```

```
# Note how the start is always included, and the end always excluded. This makes sure that s[:i] + s[i:] is always equal to s:
```

```
word[:2] + word[2:]
```

```
'Python'
```

```
word[:4] + word[4:]
```

```
'Python'
```

```
# Slice indices have useful defaults; an omitted first index defaults to zero, an omitted second index defaults to the size of the
```

```
word[:2] # character from the beginning to position 2 (excluded)
```

```
'Py'
```

```
word[4:] # characters from position 4 (included) to the end
```

```
'on'
```

```
word[-2:] # characters from the second-last (included) to the end
```

```
'on'
```

Attempting to use an index that is too large will result in an error:

```
word[42]
```

```
-----  
IndexError                                Traceback (most recent call last)  
<ipython-input-47-4d0f20275732> in <module>  
----> 1 word[42]  
  
IndexError: string index out of range
```

```
# However, out of range slice indexes are handled gracefully when used for slicing:
```

```
word[4:42]
```

```
'on'
```

```
word[42:]
```

```
''
```


Python strings cannot be changed — they are **immutable**. Therefore, assigning to an indexed position in the string results in an error:

```
word[0] = 'j'
```

```
-----  
TypeError                                 Traceback (most recent call last)  
<ipython-input-52-91a956888ca7> in <module>  
----> 1 word[0] = 'j'  
  
TypeError: 'str' object does not support item assignment
```

```
# If you need a different string, you should create a new one:
```

```
'j' + word[1:]
```

```
'Jython'
```

```
word[:2] + 'py'
```

```
'Pypy'
```

```
# The built-in function len() returns the length of a string:
```

```
s = 'supercalifragilisticexpialidocious'  
len(s)
```

1.3 Lists

Like strings (and all other built-in sequence types), lists can be indexed and sliced:

```
squares = [1, 4, 9, 16, 25]
```

```
squares
```

```
[1, 4, 9, 16, 25]
```

```
: squares = [1, 4, 9, 16, 25]
```

```
: squares
```

```
: [1, 4, 9, 16, 25]
```

```
: # Like strings (and all other built-in sequence types), lists can be indexed and sliced:
```

```
: squares[0] # indexing returns the item
```

```
: 1
```

```
: squares[-1]
```

```
: 25
```

```
: squares[-3:] # slicing returns a new list
```

```
: [9, 16, 25]
```

```
: # All slice operations return a new list containing the requested elements.
```

```
: squares[:]
```

```
: [1, 4, 9, 16, 25]
```

Lists also support operations like concatenation:

```
squares + [36, 49, 64, 81, 100]
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

Unlike strings, which are immutable, lists are a mutable type, i.e. it is possible to change their content:

```
cubes = [1, 8, 27, 65, 125] # something's wrong here
```

```
cubes[3] = 64 # replace the wrong value
```

```
cubes
```

```
[1, 8, 27, 64, 125]
```

You can also add new items at the end of the list, by using the append() method (we will see more about methods later):

```
cubes.append(216) # add the cube of 6
```

```
cubes.append(7 ** 3) # and the cube of 7
```

```
cubes
```

```
[1, 8, 27, 64, 125, 216, 343]
```

Lists also support operations like concatenation:

```
letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g']
```

```
letters
```

```
['a', 'b', 'c', 'd', 'e', 'f', 'g']
```

```
letters[2:5] = ['C', 'D', 'E'] # replace some values
```

```
letters
```

```
['a', 'b', 'C', 'D', 'E', 'f', 'g']
```

```
letters[2:5] = [] # now remove some values
```

```
letters
```

```
['a', 'b', 'f', 'g']
```

```
letters[:] = [] # clear the list by replacing all the elements with an empty list
```

```
letters[:] = []
```

```
letters
```

```
[]
```

The built-in function `len()` also applies to lists:

```
letters = ['a', 'b', 'c', 'd']
```

```
len(letters)
```

```
4
```

It is possible to nest lists (create lists containing other lists), for example:

```
a = ['a', 'b', 'c']  
n = [1, 2, 3]  
x = [a, n]
```

```
x
```

```
[['a', 'b', 'c'], [1, 2, 3]]
```

```
x[0]
```

```
['a', 'b', 'c']
```

```
x[0][1]
```

```
'b'
```


2. More Control Flow Tools

- 2.1 if Statements

```
x = int(input("Please enter an integer: "))
```

Please enter an integer: 42

```
if x < 0:
    x = 0
    print('Negative changed to zero')
elif x == 0:
    print('Zero')
elif x == 1:
    print('Single')
else:
    print('More')
```

More

2.2. for Statements

- Python's for statement iterates over the items of any sequence (a list or a string), in the order that they appear in the sequence. For example (no pun intended):

```
words = ['cat', 'window', 'defenestrate']  
for w in words:  
    print(w, len(w))
```

```
cat 3  
window 6  
defenestrate 12
```

2.3 The Range() Function

- If you do need to iterate over a sequence of numbers, the built-in function `range()` comes in handy. It generates arithmetic progressions:

```
for i in range(5):  
    print(i)
```

```
0  
1  
2  
3  
4
```

The given end point is never part of the generated sequence; `range(10)` generates 10 values, the legal indices for items of a sequence of length 10. It is possible to let the range start at another number, or to specify a different increment (even negative; sometimes this is called the ‘step’):

```
for i in range(5,10):  
    print(i)
```

5
6
7
8
9

```
for i in range(0, 10, 3):  
    print(i)
```

0
3
6
9

```
for i in range(-10, -100, -30):  
    print(i)
```

-10
-40
-70

To iterate over the indices of a sequence, you can combine `range()` and `len()` as follows:

```
a = ['Mary', 'had', 'a', 'little', 'lamb']  
for i in range(len(a)):  
    print(i, a[i])
```

```
0 Mary  
1 had  
2 a  
3 little  
4 lamb
```

In most such cases, however, it is convenient to use the `enumerate()` function

```
list(enumerate(a))
```

```
[(0, 'Mary'), (1, 'had'), (2, 'a'), (3, 'little'), (4, 'lamb')]
```


2.4. break and continue Statements on Loops

```
for letter in 'Python':  
    if letter == 'h':  
        break  
    print ('Current Letter :', letter)
```

```
Current Letter : P  
Current Letter : y  
Current Letter : t
```

The continue statement, also borrowed from C, continues with the next iteration of the loop:

```
for num in range(2, 10):  
    if num % 2 == 0:  
        print("Found an even number", num)  
        continue  
    print("Found a number", num)
```

```
Found an even number 2  
Found a number 3  
Found an even number 4  
Found a number 5  
Found an even number 6  
Found a number 7  
Found an even number 8  
Found a number 9
```

3. Data Structure

- **3.1 List Comprehensions**
- List comprehensions provide a concise way to create lists. Common applications are to make new lists where each element is the result of some operations applied to each member of another sequence or iterable, or to create a subsequence of those elements that satisfy a certain condition.
- For example, assume we want to create a list of squares, like:

```
squares = []
```

```
for x in range(10):  
    squares.append(x**2)
```

```
squares
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

```
# list comprehension
```

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

```
squares
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

```
squares = [x**2 for x in range(10) if x%2==0]
```

```
squares
```

```
[0, 4, 16, 36, 64]
```

3.2. Tuples

- We saw that lists and strings have many common properties, such as indexing and slicing operations. They are two examples of sequence data types (see Sequence Types — list, tuple, range). Since Python is an evolving language, other sequence data types may be added. There is also another standard sequence data type: the tuple.
- A tuple consists of a number of values separated by commas, for instance:

3.2. Tuples

- We saw that lists and strings have many common properties, such as indexing and slicing operations. They are two examples of sequence data types (see Sequence Types — list, tuple, range). Since Python is an evolving language, other sequence data types may be added. There is also another standard sequence data type: the tuple.
- A tuple consists of a number of values separated by commas, for instance:



```
t = 12345, 54321, 'hello!'
```

```
t[0]
```

```
12345
```

```
t
```

```
(12345, 54321, 'hello!')
```

```
#Tuples may be nested:
```

```
u = t, (1, 2, 3, 4, 5)
```

```
u
```

```
((12345, 54321, 'hello!'), (1, 2, 3, 4, 5))
```

```
# Tuples are immutable:
```

```
t[0] = 88888
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-137-d739abe3b757> in <module>  
----> 1 t[0] = 88888
```

```
TypeError: 'tuple' object does not support item assignment
```

```
# but they can contain mutable objects:
```

```
v = ([1, 2, 3], [3, 2, 1])
```

```
v
```

```
([1, 2, 3], [3, 2, 1])
```

3.3 Sets

- Python also includes a data type for sets. A set is an unordered collection with no duplicate elements. Basic uses include membership testing and eliminating duplicate entries. Set objects also support mathematical operations like union, intersection, difference, and symmetric difference.
- Curly braces or the `set()` function can be used to create sets. Note: to create an empty set you have to use `set()`, not `{}`; the latter creates an empty dictionary, a data structure that we discuss in the next section.
- Here is a brief demonstration:

```
: basket = {'apple', 'orange', 'apple', 'pear', 'orange', 'banana'}
```

```
: print(basket)
```

```
{'apple', 'banana', 'pear', 'orange'}
```

```
: 'orange' in basket
```

```
: True
```

```
: 'crabgrass' in basket
```

```
: False
```

```
a = set('abracadabra')
```

```
b = set('alacazam')
```

```
a
```

```
{'a', 'b', 'c', 'd', 'r'}
```

```
b
```

```
{'a', 'c', 'l', 'm', 'z'}
```

```
a - b
```

```
# Letters in a but not in b
```

```
{'b', 'd', 'r'}
```

```
a | b
```

```
# Letters in a or b or both
```

```
{'a', 'b', 'c', 'd', 'l', 'm', 'r', 'z'}
```

```
a & b
```

```
# Letters in both a and b
```

```
{'a', 'c'}
```

```
a ^ b
```

```
# Letters in a or b but not both
```

```
{'b', 'd', 'l', 'm', 'r', 'z'}
```

3.4 Dictionaries

- It is best to think of a dictionary as a set of key: value pairs, with the requirement that the keys are unique (within one dictionary). A pair of braces creates an empty dictionary: {}. Placing a comma-separated list of key:value pairs within the braces adds initial key:value pairs to the dictionary; this is also the way dictionaries are written on output.
- The main operations on a dictionary are storing a value with some key and extracting the value given the key. It is also possible to delete a key:value pair with del. If you store using a key that is already in use, the old value associated with that key is forgotten. It is an error to extract a value using a non-existent key.
- Performing list(d) on a dictionary returns a list of all the keys used in the dictionary, in insertion order (if you want it sorted, just use sorted(d) instead). To check whether a single key is in the dictionary, use the in keyword.

```
tel = {'jack': 4098, 'sape': 4139}
tel['guido'] = 4127
```

```
tel
```

```
{'jack': 4098, 'sape': 4139, 'guido': 4127}
```

```
tel['jack']
```

```
4098
```

```
del tel['sape']
```

```
tel['irv'] = 4127
```

```
tel
```

```
{'jack': 4098, 'guido': 4127, 'irv': 4127}
```

```
list(tel)
```

```
['jack', 'guido', 'irv']
```

```
sorted(tel)
```

```
['guido', 'irv', 'jack']
```

```
'guido' in tel
```

```
True
```

```
'jack' not in tel
```

```
False
```

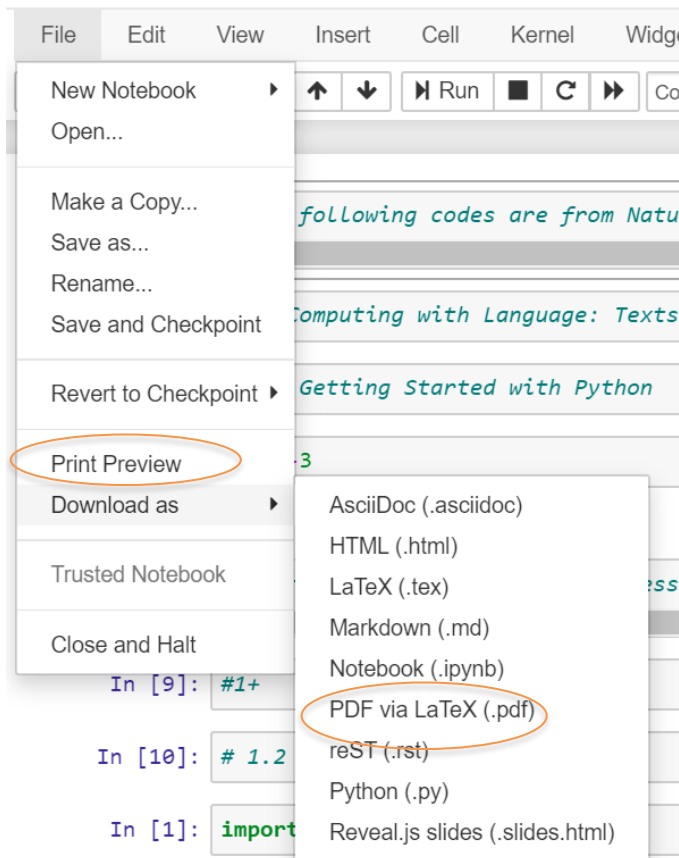
```
{x: x**2 for x in (2, 4, 6)}
```

```
{2: 4, 4: 16, 6: 36}
```

When the keys are simple strings, it is sometimes easier to specify pairs using keyword arguments:

```
dict(sape=4139, guido=4127, jack=4098)
```


Save your lab report and assignments



Option 1 (recommended):
Print preview → print as “pdf”

Option 2:
Download as PDF and upload .pdf file to BB
If you have installed Latex

Install LaTeX

- <https://nbconvert.readthedocs.io/en/latest/install.html#installing-tex>

Installing TeX

For converting notebooks to PDF (with `--to pdf`), nbconvert makes use of LaTeX and the XeTeX as the rendering engine.

New in version 5.0: We use XeTeX as the rendering engine rather than pdfTeX (as in earlier versions). XeTeX can access fonts through native operating system libraries, it has better support for OpenType formatted fonts and Unicode characters.

To install a complete TeX environment (including XeLaTeX and the necessary supporting packages) by hand can be tricky. Fortunately, there are packages that make this much easier. These packages are specific to different operating systems:

- Linux: [TeX Live](#)
 - E.g. on Debian or Ubuntu:

```
sudo apt-get install texlive-xetex texlive-fonts-recommended texlive-generic-recommended
```

- macOS (OS X): [MacTeX](#).
- Windows: [MikTeX](#)

Windows Version

[Home](#) [About](#) [Docs](#) [Downloads](#) [Give Back](#) [Help](#)

Getting MiKTeX

MiKTeX is available for selected operating systems. Please check the [prerequisites](#) in order to find out whether your system is supported.

If your system is not (yet) supported: it is not too difficult to [build MiKTeX](#).

[Windows](#) [Mac](#) [Linux](#) [Docker](#) [All downloads](#)

Install on Windows

[Installer](#) [Portable Edition](#) [Command-line installer](#)

Installer

To install a basic TeX/LaTeX system on Windows, download and run this installer.

Please read the [tutorial](#), if you want step-by-step guidance.

Date: 7/3/2020

File name: `basic-miktex-20.6.29-x64.exe`

Size: 234.43 MB

SHA-256: `b69f96e56a2a7c49f9cac106d743284ede86d18f986a6a202e9628ed30447885`

[Download](#)

© 2020 Christian Schenk

[Packages](#)
[A-Z](#)
[Browse](#)
[Packaging](#)
[Repositories](#)

[Developers](#)
[Build MiKTeX](#)

[Legal](#)
[License](#)
[Privacy Policy](#)

[Datenschutzerklärung](#)
[Widerrufsbelehrung](#)
[Impressum](#)

Download All
Driver Updates
- 100% Free -

File Name:
WinZip
System Suite

Download Size:
45 MB

Details:
Update all drivers
- fast and free

OS:
Windows
10, 8, 7, XP, Vista



Get Started

Getting MiKTeX

MiKTeX is available for selected operating systems. Please check the [prerequisites](#) in order to find out whether your system is supported.

If your system is not (yet) supported: it is not too difficult to [build MiKTeX](#).

Windows

Mac

Linux

Docker

All downloads

Install on Windows

Installer

Portable Edition

Command-line installer

Installer

To install a basic TeX/LaTeX system on Windows, download and run this installer.

Please read the [tutorial](#), if you want step-by-step guidance.

Date: 7/3/2020

File name: `basic-miktex-20.6.29-x64.exe`

Size: 234.43 MB

SHA-256: `b69f96e56a2a7c49f9cac106d743284ede86d18f986a6a202e9628ed30447885`

Download

© 2020 Christian Schenk

Packages

A-Z
Browse
Packaging
Repositories

Developers

Build MiKTeX

Basic MiKTeX Installer (20.6.29, 7487, 64-bit)

Executing

The main task is being executed.

Installing:

Overall progress

this is MiKTeX Setup Service 4.0 (MiKTeX 20.6.29)
starting installer...
Loading package database...
starting package maintenance...
installation directory: C:\Users\Chaoqun\AppData\Local\Programs\MiKTeX
package repository: C:\Users\Chaoqun\AppData\Local\Temp\mik29781

< Back

Next >

Cancel

NEW YORK SMART. WORLD-CLASS READY.®



Q and A

Outline

- Introduction to the Class
 - Getting to know me
 - Getting to know you
 - Introduction to syllabus
- Introduction to NLP
 - What is NLP?
 - NLP Applications
 - Key NLP Components
 - Machine Learning for NLP
 - Why is NLP hard?
- Review of Python Programming
- Review of Machine Learning

Review of Machine Learning Methods

- Supervised Learning
- We are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output.
 - Regression: predict single value outcome
 - Classification: predict a class of values
- Unsupervised Learning
- Allows us to approach problems with little or no idea what our results should look like (only have X, no Y)
 - Clustering: given the data, find the structure in the data
 - Objects in the same group (called a cluster) are more similar to each other than those in other groups (clusters).

Regression Problems vs Classification Problems

- Regression
 - Whether the response (target or DV) take on numerical values
 - Quantitative Response
 - E.g. a person's age, height, or income, the value of a house, the price of a stock
- Classification
 - Whether the response (target or DV) take on values in one of K different classes or categories
 - Qualitative response
 - E.g. gender (male/female), whether a person defaults on a debt (Yes/No)
- Which statistical learning method is best suited to a problem tends to depend on whether the response is qualitative or quantitative

Exercises

- 1) Which of the following is not one of the supervised learning problems?
- A. Linear Regression
- B. Logistic Regression
- C. Classification
- D. Clustering

Exercises

- 1) Which of the following is not one of the supervised learning problems?
- A. Linear Regression
- B. Logistic Regression
- C. Classification
- D. Clustering

Exercise

- 2) Which of the following problems can be referred to as classification problems ?
- A. Predict a student's grade for this course
- B. Predict the stock price
- C. Predict the sales for a product
- D. Predict whether the student will pass or fail in this course

Exercise

- 2) Which of the following problems can be referred to as classification problems ?
- A. Predict a student's grade for this course
- B. Predict the stock price
- C. Predict the sales for a product
- D. Predict whether the student will pass or fail in this course

Machine Learning Methods

Supervised Learning Methods


- Regression: (Linear, Multilinear Regression)
- Classification: (Logistic Regression, KNN)
- Regression and Classification: Decision Tree Methods: (Regression Tree and Classification Tree)
- Classification: Naïve Bayes

Unsupervised Learning Methods

- Principle Component Analysis (PCA)-Dimensionality Reduction
- Clustering (K-Mean, Hierarchical Clustering)
- Association Rule Mining

Assignments

- Read Chapter 1 of Book
- Team Formation by Sep 7



NEW YORK SMART. WORLD-CLASS READY.®



Thank You

Questions or Comments?