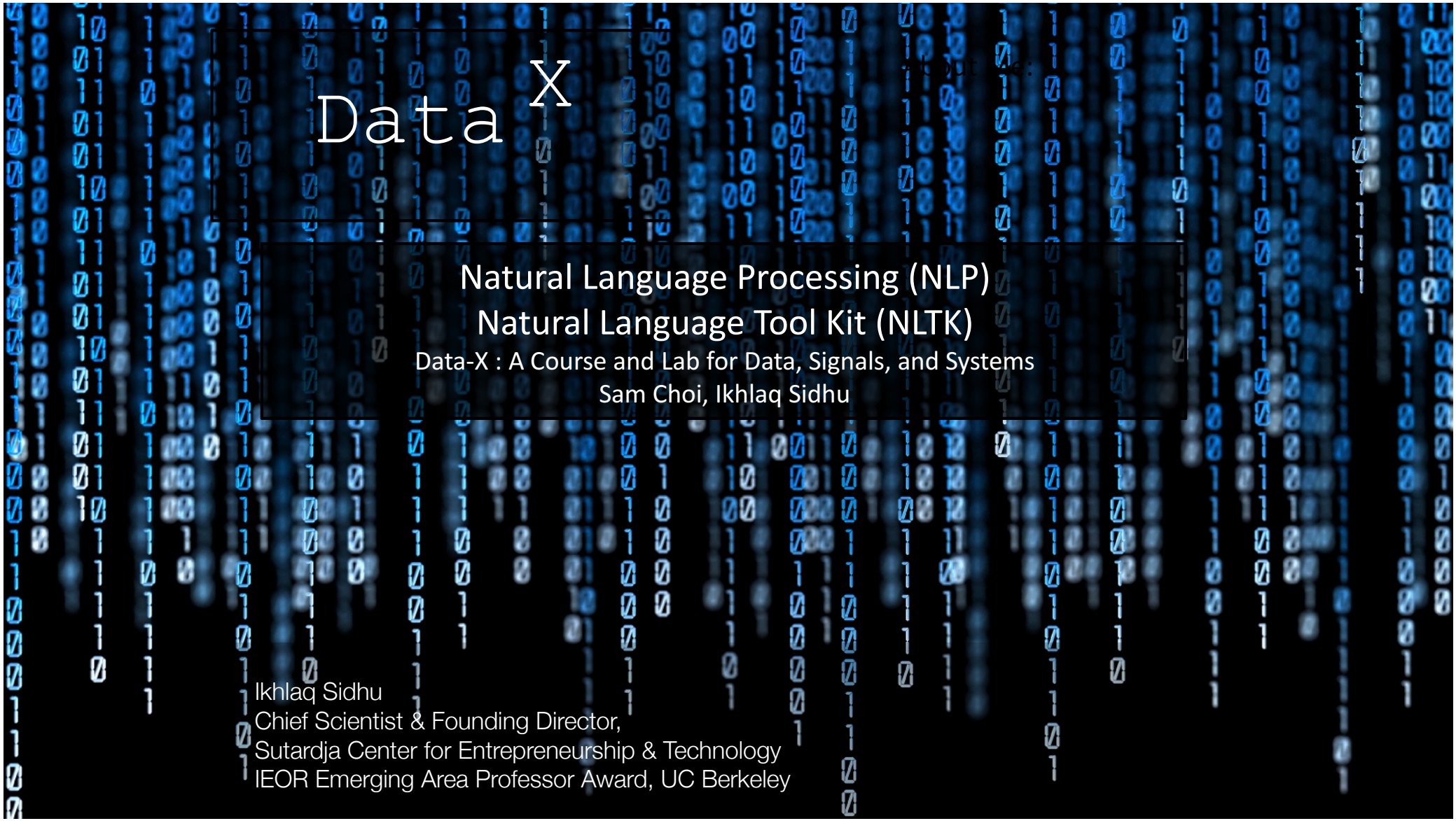
Data X
Natural Language Processing (NLP)
Natural Language Tool Kit (NLTK)
Data-X : A Course and Lab for Data, Signals, and Systems
Sam Choi, Ikhlaq Sidhu
Ikhlaq Sidhu
Chief Scientist & Founding Director,
Sutardja Center for Entrepreneurship & Technology
IEOR Emerging Area Professor Award, UC Berkeley

NLP: Main Idea

- What is NLP?

NLP: Main Idea

- What is NLP?
  - Natural Language Processing

NLP: Main Idea

- What is NLP?
  - Natural Language Processing
- Core Questions

NLP: Main Idea

- What is NLP?
  - Natural Language Processing
- Core Questions
  - How can we get a computer to understand speech and writing?
  - How can we get a computer to speak/write like a person?

NLP: Main Idea

- What is NLP?
  - Natural Language Processing
- Core Questions
  - How can we get a computer to understand speech and writing?
  - How can we get a computer to speak/write like a person?

NLP: Main Idea

- Natural language understanding

NLP: Main Idea

- Natural language understanding
  - How can a computer understand the meaning and nuances of human language

NLP: Main Idea

- Natural language understanding
  - How can a computer understand the meaning and nuances of human language
- Natural language generation

NLP: Main Idea

- Natural language understanding
  - How can a computer understand the meaning and nuances of human language
- Natural language generation
  - Respond to language queries
  - Convert data stored into readable human language
    - Chat/Email bots
    - Siri/Alexa

NLP: Implications

- Why is Natural Language Processing Important?

NLP: Implications

- Why is Natural Language Processing Important?
  - Short Answer: Because natural language is important

NLP: Implications

- Why is Natural Language Processing Important?
  - Short Answer: Because natural language is important
  - Data is not only numerical, but also textual
  - Deriving strategies to extrapolate information from this data is difficult

NLP Applications

- How can we use NLP to our advantage?

NLP Applications

- How can we use NLP to our advantage?
- High-Level Applications

NLP Applications

- How can we use NLP to our advantage?
- High-Level Applications
  - Language Translation (Google Translate)

NLP Applications

- How can we use NLP to our advantage?
- High-Level Applications
  - Language Translation (Google Translate)
  - Speech detection (Siri, SoundHound)

NLP Applications

- How can we use NLP to our advantage?
- High-Level Applications
  - Language Translation (Google Translate)
  - Speech detection (Siri, SoundHound)
  - Sentiment Analysis (Kensho)

NLP Applications

- How can we use NLP to our advantage?
- High-Level Applications
  - Language Translation (Google Translate)
  - Speech detection (Siri, SoundHound)
  - Sentiment Analysis (Kensho)
  - Plagiarism detector (turnitin)

NLP Applications

- How can we use NLP to our advantage?
- High-Level Applications
  - Language Translation (Google Translate)
  - Speech detection (Siri, SoundHound)
  - Sentiment Analysis (Kensho)
  - Plagiarism detector (turnitin)
  - Grammar/Spelling checking (gmail, microsoft word)

NLP Applications

- How can we use NLP to our advantage?
- High-Level Applications
  - Language Translation (Google Translate)
  - Speech detection (Siri, SoundHound)
  - Sentiment Analysis (Kensho)
  - Plagiarism detector (turnitin)
  - Grammar/Spelling checking (gmail, microsoft word)
  - Construction/Generation (chat bots)

Case Study

- Kensho
  - Financial data analysis

Case Study

- Kensho
  - Financial data analysis
- How do they use NLP?

Case Study

- Kensho
  - Financial data analysis
- How do they use NLP?
  - Natural language understanding

Case Study

- Kensho
  - Financial data analysis
- How do they use NLP?
  - Natural language understanding
    - organizes corpora of a variety of textual data
    - economic reports, financial policies, political reports, drug approvals, etc

Case Study

- Kensho
  - Financial data analysis
- How do they use NLP?
  - Natural language understanding
    - organizes corpora of a variety of textual data
    - economic reports, financial policies, political reports, drug approvals, etc
  - Natural language generation

Case Study

- Kensho
  - Financial data analysis
- How do they use NLP?
  - Natural language understanding
    - organizes corpora of a variety of textual data
    - economic reports, financial policies, political reports, drug approvals, etc
  - Natural language generation
    - Generate answers to questions and produce financial reports

NLP Subproblems

- Lower Level Problems

NLP Subproblems

- Lower Level Problems
  - Co-reference
    - Multiple words refer to the same subject
    - Ex: Ikhlaq, professor, he

NLP Subproblems

- Lower Level Problems
  - Co-reference
    - Multiple words refer to the same subject
    - Ex: Ikhlaq, professor, he
  - Classification
    - Labeling input based on type/class

NLP Subproblems

- Lower Level Problems
  - Co-reference
    - Multiple words refer to the same subject
    - Ex: Ikhlaq, professor, he
  - Classification
    - Labeling input based on type/class
  - Morphological
    - Identifying different forms of a word
    - Ex: open, opened, opens, opening

NLP Subproblems

- Subproblems

NLP Subproblems

- Subproblems
  - Part-of-speech tagging

NLP Subproblems

- Subproblems
  - Part-of-speech tagging
  - Parsing

NLP Subproblems

- Subproblems
  - Part-of-speech tagging
  - Parsing
  - Sentence breaking (finding sentence boundaries)

NLP Subproblems

- Subproblems
  - Part-of-speech tagging
  - Parsing
  - Sentence breaking (finding sentence boundaries)
  - Word segmentation (separating text by word)

NLP Subproblems

- Subproblems
  - Part-of-speech tagging
  - Parsing
  - Sentence breaking (finding sentence boundaries)
  - Word segmentation (separating text by word)
- What tools are available to simply these problems?

NLP Subproblems

- Subproblems
  - Part-of-speech tagging
  - Parsing
  - Sentence breaking (finding sentence boundaries)
  - Word segmentation (separating text by word)
- What tools are available to simply these problems?
  - NLTK

NLTK: Introduction

- What is NLTK?

NLTK: Introduction

- What is NLTK?
  - The Natural Language Toolkit

NLTK: Introduction

- What is NLTK?
  - The Natural Language Toolkit
  - Platform created for working with textual data
  - Libraries for NLP development in Python

NLTK: Introduction

- What is NLTK?
  - The Natural Language Toolkit
  - Platform created for working with textual data
  - Libraries for NLP development in Python
- Similar Resources
  - Stanford's Core NLP Suite

NLTK

- Features

NLTK

- Features
  - Sentence & word tokenization

NLTK

- Features
  - Sentence & word tokenization
  - Part of speech tagging

NLTK

- Features
  - Sentence & word tokenization
  - Part of speech tagging
  - Chunking & named entity recognition

NLTK

- Features
  - Sentence & word tokenization
  - Part of speech tagging
  - Chunking & named entity recognition
  - Text classification

NLTK

- Features
  - Sentence & word tokenization
  - Part of speech tagging
  - Chunking & named entity recognition
  - Text classification
- Resources

NLTK

- Features
  - Sentence & word tokenization
  - Part of speech tagging
  - Chunking & named entity recognition
  - Text classification
- Resources
  - Corpora, large sets of organized data
  - Sources include: WSJ, twitter, Project Gutenberg, etc.

NLTK: Getting Started

- Install Python
  - https://www.python.org/downloads/
- Install NLTK
  - http://www.nltk.org/install.html
- Download Corpora (NLTK Data)
  - http://www.nltk.org/data.html

Using NLTK

- Basic Functions

Using NLTK

- Basic Functions
  - words()
    - Partitions a text file into a list where each element in a word

Using NLTK

- Basic Functions
  - words()
    - Partitions a text file into a list where each element in a word
  - sents()
    - Partitions a text file into lists of words – each list a sentence

Using NLTK

- Basic Functions
  - words()
    - Partitions a text file into a list where each element in a word
  - sents()
    - Partitions a text file into lists of words – each list a sentence
  - sent_tokenize
    - Organize text into a list of sentences

Using NLTK

- Basic Functions
  - words()
    - Partitions a text file into a list where each element in a word
  - sents()
    - Partitions a text file into lists of words – each list a sentence
  - sent_tokenize
    - Organize text into a list of sentences
  - word_tokenize
    - organize text into a list of words

Using NLTK

- Basic Functions
  - words()
    - Partitions a text file into a list where each element in a word
  - sents()
    - Partitions a text file into lists of words – each list a sentence
  - sent_tokenize
    - Organize text into a list of sentences
  - word_tokenize
    - organize text into a list of words
  - pos_tag
    - tag part of speech for each word in a list

sent_tokenize & word_tokenize

- sent_tokenize

sent_tokenize & word_tokenize

- sent_tokenize
  - Takes a single string as input
  - Returns the string as a list of sentences

sent_tokenize & word_tokenize

- sent_tokenize
  - Takes a single string as input
  - Returns the string as a list of sentences
- word_tokenize

sent_tokenize & word_tokenize

- sent_tokenize
  - Takes a single string as input
  - Returns the string as a list of sentences
- word_tokenize
  - Takes a single string as input
  - Returns the string as a list of words

Sentence Tokenization

```
>>> from nltk.tokenize import sent_tokenize

>>> sent_tokenize("Hello Data-X. This is NLTK.")
```

Sentence Tokenization

```
>>> from nltk.tokenize import sent_tokenize

>>> sent_tokenize("Hello Data-X. This is NLTK.")
['Hello Data-X.', 'This is NLTK.']
```

Sentence Tokenization

```
>>> from nltk.tokenize import sent_tokenize

>>> sent_tokenize("Hello Data-X. This is NLTK.")
['Hello Data-X.', 'This is NLTK.']

>>> sent_tokenize("Hello, Sam. Welcome to Data-X!")
```

Sentence Tokenization

```
>>> from nltk.tokenize import sent_tokenize

>>> sent_tokenize("Hello Data-X. This is NLTK.")
['Hello Data-X.', 'This is NLTK.']

>>> sent_tokenize("Hello, Sam. Welcome to Data-X!")
['Hello Sam.', ’Welcome to Data-X!']
```

Word Tokenization

```
>>> from nltk.tokenize import word_tokenize

>>> word_tokenize("This is NLTK.")
```

Word Tokenization

```
>>> from nltk.tokenize import word_tokenize

>>> word_tokenize("This is NLTK.")
['This', 'is' , 'NLTK' , '.' ]
```

Word Tokenization

```
>>> from nltk.tokenize import word_tokenize

>>> word_tokenize("This is NLTK.")
['This', 'is' , 'NLTK' , '.' ]

>>> bio = "Hi, everyone. My name is Sam.”

>>> word_tokenize(bio)
```

Word Tokenization

```
>>> from nltk.tokenize import word_tokenize

>>> word_tokenize("This is NLTK.")
['This', 'is' , 'NLTK' , '.' ]

>>> bio = "Hi, everyone. My name is Sam.”

>>> word_tokenize(bio)
['Hi', ',', 'everyone', '.', 'My', 'name', 'is', 'Sam', '.']
```

pos_tag

- pos_tag

pos_tag

- pos_tag
  - Takes a single string as input
  - Returns the string as a list of tuples

pos_tag

- pos_tag
  - Takes a single string as input
  - Returns the string as a list of tuples
    - Pairs of words and their respective part-of-speech tags
      - (Sam, NNP)

Part-of-Speech Tagging

```
>>> words = word_tokenize("Hi, everyone. My name is Sam.")
>>> from nltk import pos_tag
```

Part-of-Speech Tagging

```
>>> words = word_tokenize("Hi, everyone. My name is Sam.")
>>> from nltk import pos_tag

>>> pos_tag(words)
```

Part-of-Speech Tagging

```
>>> words = word_tokenize("Hi, everyone. My name is Sam.")
>>> from nltk import pos_tag

>>> pos_tag(words)
[('Hi', 'NNP'), (',', ','), ('everyone', 'NN'), ('.', '.'), ('My', 'PRP$'),
('name', 'NN'), ('is', 'VBZ'), ('Sam', 'NNP'), ('.', '.')]
```

Part-of-Speech Tagging

```
>>> words = word_tokenize("Hi, everyone. My name is Sam.")
>>> from nltk import pos_tag

>>> pos_tag(words)
[('Hi', 'NNP'), (',', ','), ('everyone', 'NN'), ('.', '.'), ('My', 'PRP$'),
('name', 'NN'), ('is', 'VBZ'), ('Sam', 'NNP'), ('.', '.')]
```

- NNP -> Proper Noun, singular
- NN -> Noun, singular or mass
- PRP$ -> Possessive pronoun
- VBZ -> Verb, 3rd person singular present

Part-of-Speech Tagging

```
>>> words = word_tokenize("Hi, everyone. My name is Sam.")
>>> from nltk import pos_tag

>>> pos_tag(words)
[('Hi', 'NNP'), (',', ','), ('everyone', 'NN'), ('.', '.'), ('My', 'PRP$'),
('name', 'NN'), ('is', 'VBZ'), ('Sam', 'NNP'), ('.', '.')]
```

- NNP -> Proper Noun, singular
- NN -> Noun, singular or mass
- PRP$ -> Possessive pronoun
- VBZ -> Verb, 3rd person singular present

List of tags: http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

NLTK Data

- Organized into collections of written texts (corpora)
- Examples of NLTK Corpora
  - gutenberg (Project Gutenberg selections)
  - shakespeare (selection of Shakespeare's plays)
  - twitter_samples (samples of tweets)
  - brown (Brown University's collection of published works)
  - cmudict (Carnegie Mellon's dictionary of words/pronunciations)

End of Section