Announcements

- Readings (Week 4 due Saturday 2/1)
  - Email: gquer@ucsd.edu
  - Subject line:
  - [DSC96 W20]: Week 04, Sec A/B, YourFirstName YourLastName
  - Email content:
    - Your comments/ questions/ observations on the proposed lectures

- Assignment 2:
  - By end of week 7 (more details next week)

Natural Language Processing

Colin Jemmott & Giorgio Quer

Structured and Unstructured Data

Structured

- In a database
- Sorted and labeled with regular structure
- Proper types

Unstructured

- Just a bunch of stuff on the computer!
- Irregular and had ambiguities
- Difficult to understand using traditional programs

Structured and Unstructured Data

Structured

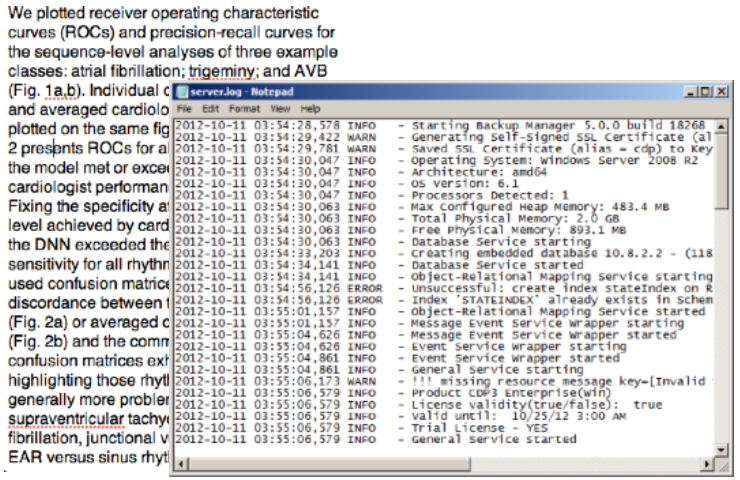
- In a database
- Sorted and labeled with regular structure
- Proper types

Unstructured

- Just a bunch of stuff on the computer!
- Irregular and had ambiguities
- Difficult to understand using traditional programs

We plotted receiver operating characteristic curves (ROCs) and precision-recall curves for the sequence-level analyses of three example classes: atrial fibrillation; trigeminy; and AVB (Fig. 1a,b). Individual cardiologist performance and averaged cardiologist performance are plotted on the same figure. Extended Data Fig. 2 presents ROCs for all classes, showing that the model met or exceeded the averaged cardiologist performance for all rhythm classes. Fixing the specificity at the average specificity level achieved by cardiologists, the sensitivity of the DNN exceeded the average cardiologist sensitivity for all rhythm classes (Table 2). We used confusion matrices to illustrate the discordance between the DNN's predictions (Fig. 2a) or averaged cardiologist predictions (Fig. 2b) and the committee consensus. The two confusion matrices exhibit a similar pattern, highlighting those rhythm classes that were generally more problematic to classify (that is, supraventricular tachycardia (SVT) versus atrial fibrillation, junctional versus sinus rhythm, and EAR versus sinus rhythm).

Structured and Unstructured Data



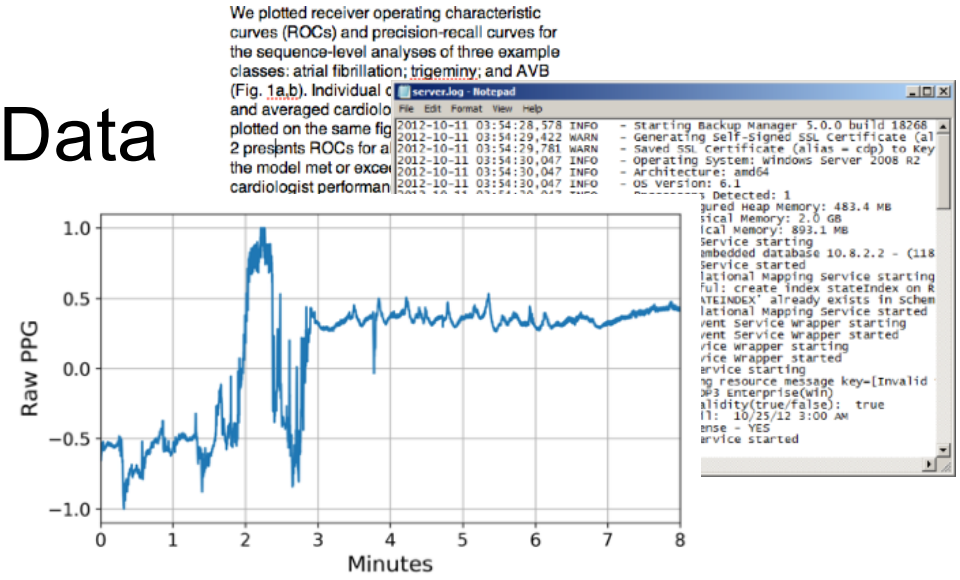
Structured

- In a database
- Sorted and labeled with regular structure
- Proper types

Unstructured

- Just a bunch of stuff on the computer!
- Irregular and had ambiguities
- Difficult to understand using traditional programs

Structured and Unstructured Data



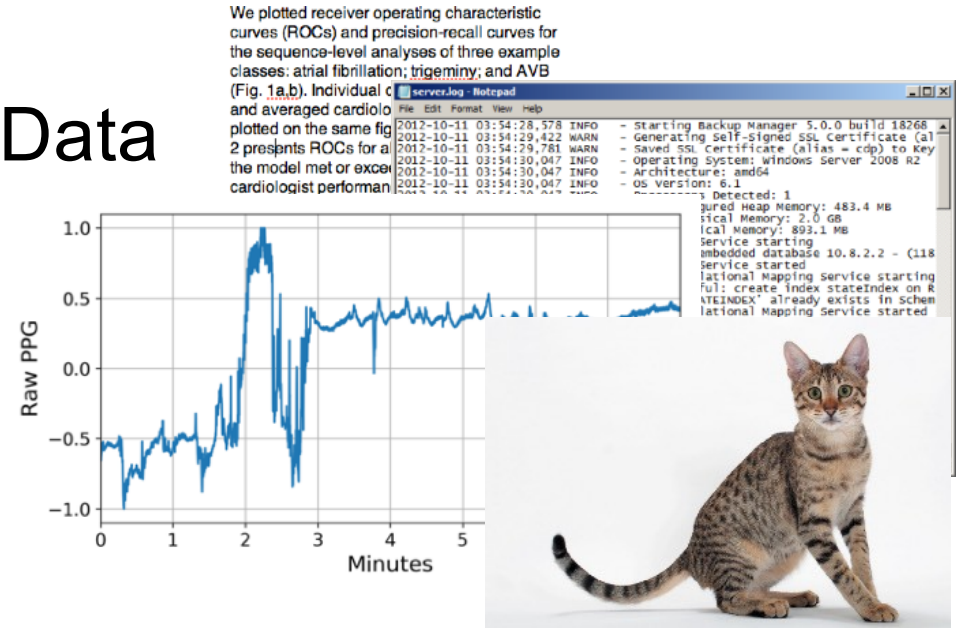
Structured

- In a database
- Sorted and labeled with regular structure
- Proper types

Unstructured

- Just a bunch of stuff on the computer!
- Irregular and had ambiguities
- Difficult to understand using traditional programs

Structured and Unstructured Data



Structured

- In a database
- Sorted and labeled with regular structure
- Proper types

Unstructured

- Just a bunch of stuff on the computer!
- Irregular and had ambiguities
- Difficult to understand using traditional programs

Structured and Unstructured Data

Structured

- In a database
- Sorted and labeled with regular structure
- Proper types

Unstructured

- Just a bunch of stuff on the computer!
- Irregular and had ambiguities
- Difficult to understand using traditional programs

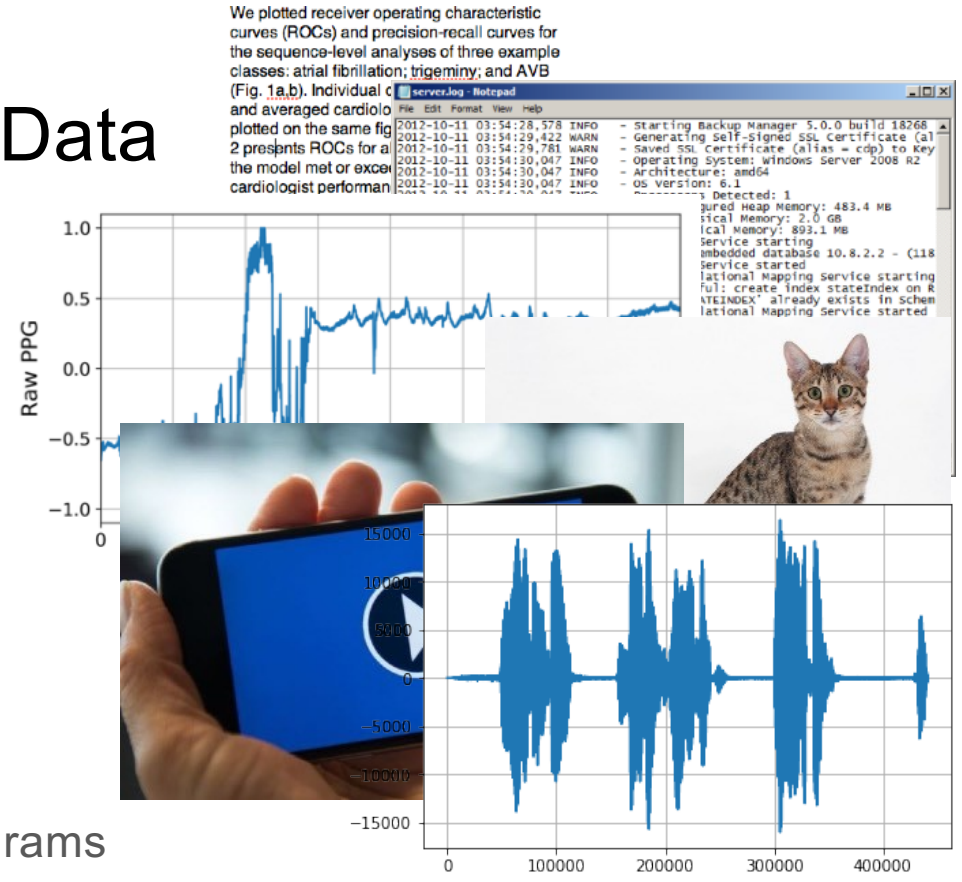
Structured and Unstructured Data

Structured

- In a database
- Sorted and labeled with regular structure
- Proper types

Unstructured

- Just a bunch of stuff on the computer!
- Irregular and had ambiguities
- Difficult to understand using traditional programs



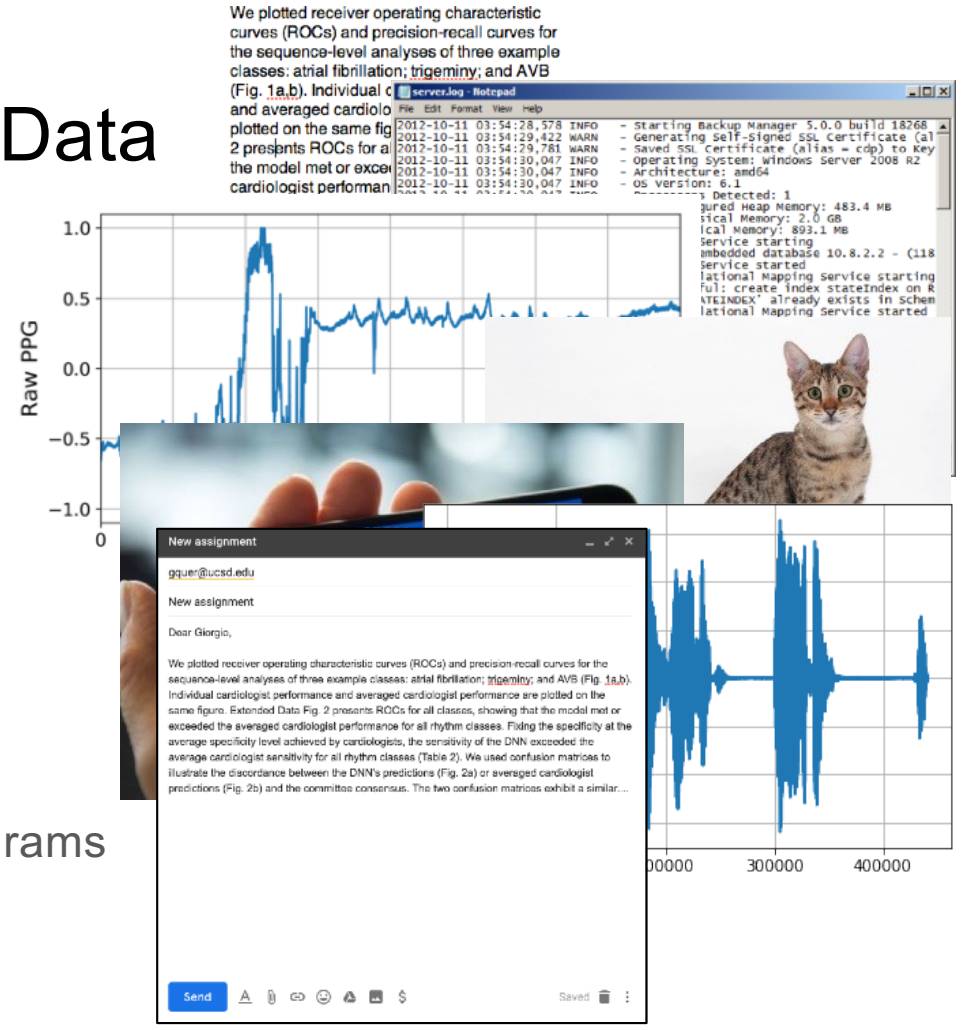
Structured and Unstructured Data

Structured

- In a database
- Sorted and labeled with regular structure
- Proper types

Unstructured

- Just a bunch of stuff on the computer!
- Irregular and had ambiguities
- Difficult to understand using traditional programs



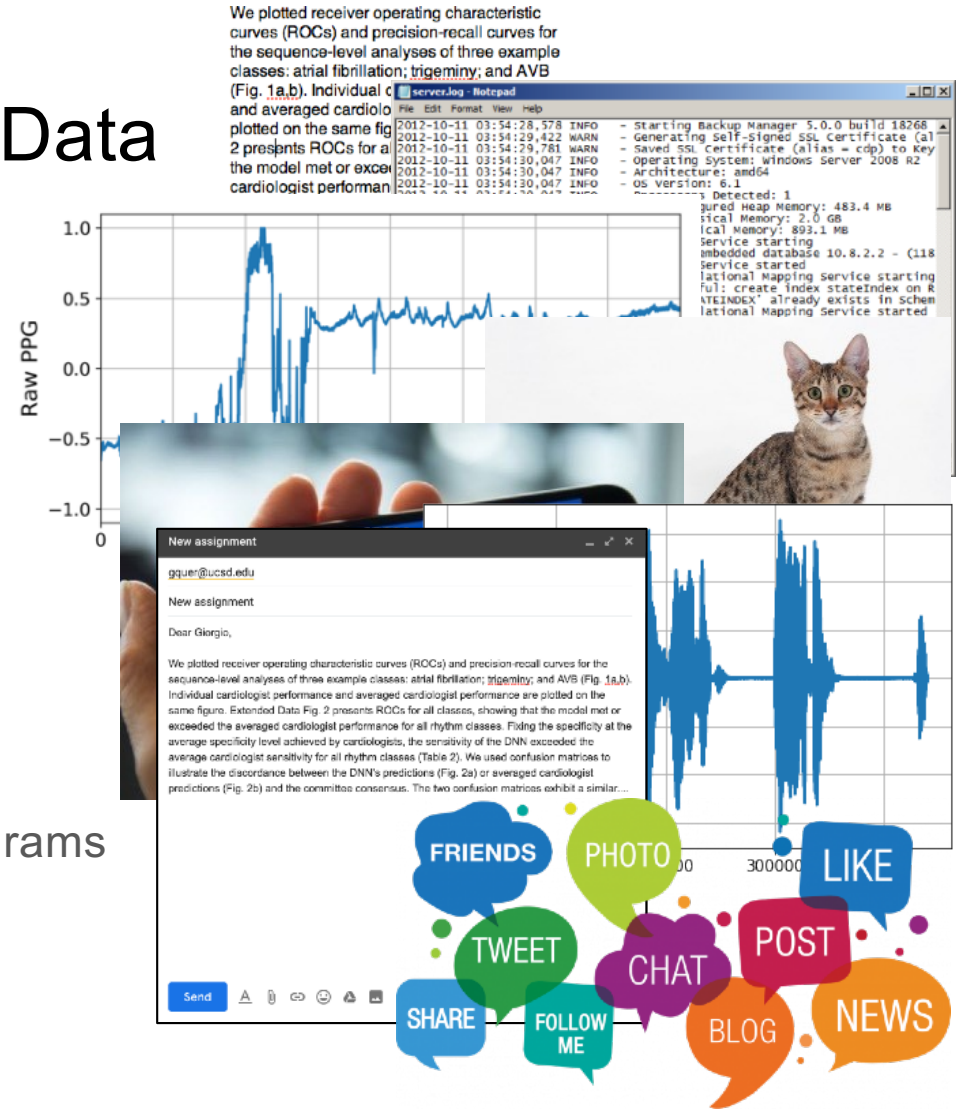
Structured and Unstructured Data

Structured

- In a database
- Sorted and labeled with regular structure
- Proper types

Unstructured

- Just a bunch of stuff on the computer!
- Irregular and had ambiguities
- Difficult to understand using traditional programs



Hutzler 571 Banana Slicer by Hutzler Manufacturing Co.

"What can I say about the 571B Banana Slicer that hasn't already been said about the wheel, penicillin, or the iPhone?"
Mrs Toledo

"Gone are the days of biting off slice-sized chunks of banana and spitting them onto a serving tray…. Next on my wish list: a kitchen tool for dividing frozen water into cube-sized chunks."
N. Krumpe

"As shown in the picture, the slices is curved from left to right. All of my bananas are bent the other way."
J. Anderson

80-90% of data is unstructured, and much of it is text. What can we do with it?

Syntax

Word segmentation

- This might be easy - or it “isn’t.”

Lemmatization and Stemming

- Reducing the inflectional forms of each word into a common base or root

Part-of-speech tagging

- Example: noun ("the book on the table") or verb ("to book a flight");

Semantics

Named entity recognition (NER)

- Which items in text map to proper names? What type (e.g. person, location)?

Machine translation

Sentiment Analysis

Natural language understanding, Question answering, Relationship extraction, Topic segmentation and recognition, Word sense disambiguation

NLTK: natural language toolkit

Tokenize and tag some text:

```
>>> import nltk
>>> sentence = """At eight o'clock on Thursday morning
... Arthur didn't feel very good."""
>>> tokens = nltk.word_tokenize(sentence)
>>> tokens
['At', 'eight', "o'clock", 'on', 'Thursday', 'morning',
'Arthur', 'did', "n't", 'feel', 'very', 'good', '.']
>>> tagged = nltk.pos_tag(tokens)
>>> tagged[0:6]
[('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'), ('on', 'IN'),
('Thursday', 'NNP'), ('morning', 'NN')]
```

https://pythonprogramming.net/natural-language-toolkit-nltk-part-speech-tagging/

NLTK

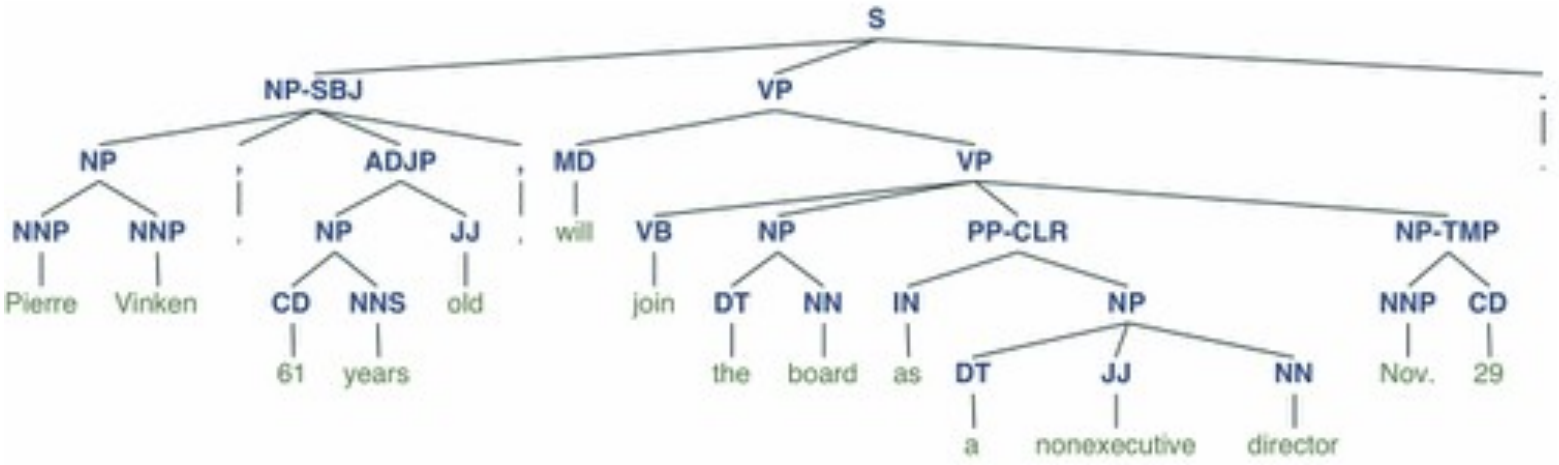
Identify named entities:

```
>>> entities = nltk.chunk.ne_chunk(tagged)
>>> entities
Tree('S', [('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'),
           ('on', 'IN'), ('Thursday', 'NNP'), ('morning', 'NN'),
       Tree('PERSON', [('Arthur', 'NNP')]),
           ('did', 'VBD'), ("n't", 'RB'), ('feel', 'VB'),
           ('very', 'RB'), ('good', 'JJ'), ('.', '.')])
```

NLTK

Display a parse tree:

```
>>> from nltk.corpus import treebank
>>> t = treebank.parsed_sents('wsj_0001.mrg')[0]
>>> t.draw()
```



Other NLP Tools

Commercial solutions (Google, Microsoft, Amazon, IBM, etc)

- Translation: don’t DIY

SpaCy

- Similar performance to NLTK
- Many fewer options
- ~500x faster