

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

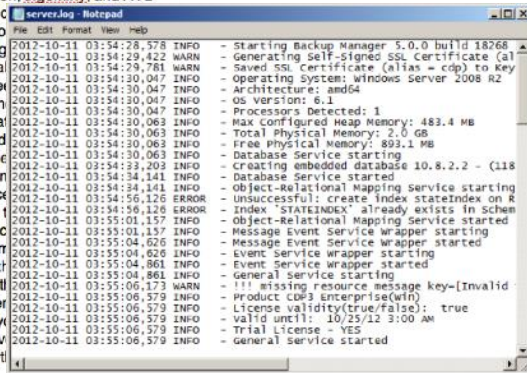
- In a database
- Sorted and labeled with regular structure
- Proper types

Unstructured

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

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 c

and averaged cardiolo
plotted on the same fig
2 presnts ROCs for al
the model met or exce
cardiologist perform
Fixing the specificity a
level achieved by card
the DNN exceeded the
sensitivity for all rhyth
used confusion matrix
discordance between t
(Fig. 2a) or averaged c
(Fig. 2b) and the comm
confusion matrices ext
highlighting those rhyt
generally more proble
supraventricular tachy
fibrillation, junctional v
EAR versus sinus rhyt



```
server.log - Notepad
File Edit Format View Help
2012-10-11 03:54:28,578 INFO - Starting Backup Manager 5.0.0 build 18268
2012-10-11 03:54:29,422 WARN - Generating self-signed SSL Certificate (al
2012-10-11 03:54:29,781 WARN - Saved SSL Certificate (alias = cdp) to Key
2012-10-11 03:54:30,047 INFO - Operating System: Windows Server 2008 R2
2012-10-11 03:54:30,047 INFO - Architecture: amd64
2012-10-11 03:54:30,047 INFO - OS version: 6.1
2012-10-11 03:54:30,047 INFO - Processors Detected: 1
2012-10-11 03:54:30,063 INFO - Max configured Heap Memory: 483.4 MB
2012-10-11 03:54:30,063 INFO - Total Physical Memory: 2.0 GB
2012-10-11 03:54:30,063 INFO - Free Physical Memory: 893.1 MB
2012-10-11 03:54:30,063 INFO - Database Service starting
2012-10-11 02:54:33,203 INFO - Creating embedded database 10.8.2.2 - (118
2012-10-11 03:54:34,141 INFO - Database Service started
2012-10-11 03:54:34,141 INFO - Object-Relational Mapping Service starting
2012-10-11 03:54:56,126 ERROR - Unsuccessful: create index stateIndex on R
2012-10-11 03:55:01,157 INFO - Index 'STATEINDEX' already exists in schem
2012-10-11 03:55:01,157 INFO - Object-Relational Mapping Service started
2012-10-11 03:55:04,626 INFO - Message Event Service wrapper starting
2012-10-11 03:55:04,626 INFO - Message Event Service wrapper started
2012-10-11 03:55:04,626 INFO - Event Service wrapper starting
2012-10-11 03:55:04,861 INFO - Event Service wrapper started
2012-10-11 03:55:06,173 WARN - !!! missing resource message key-[Invalid
2012-10-11 03:55:06,579 INFO - Product CDWS enterprise(win)
2012-10-11 03:55:06,579 INFO - License validity(true/false): true
2012-10-11 03:55:06,579 INFO - valid until: 10/25/12 3:00 AM
2012-10-11 03:55:06,579 INFO - Trial License = YES
2012-10-11 03:55:06,579 INFO - General Service started
```

Structured and Unstructured Data

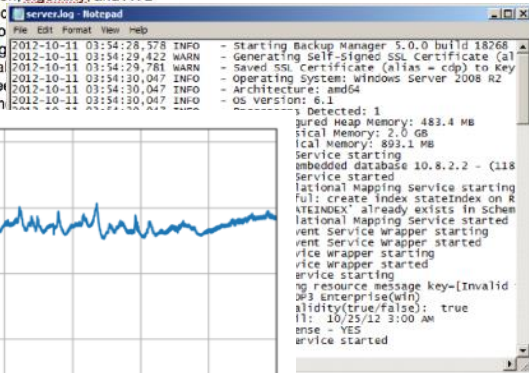
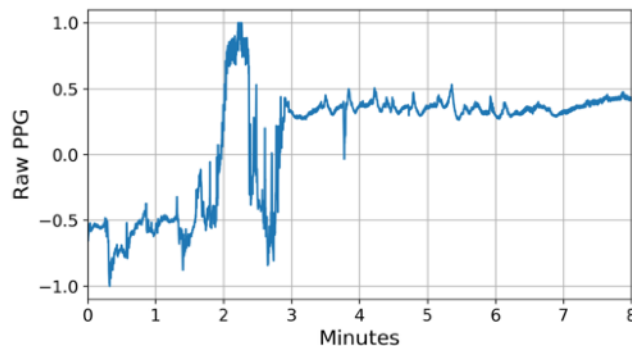
Structured

- 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. Figure 2 presents ROCs for all the model met or exceeded the cardiologist performance.



Structured and Unstructured Data

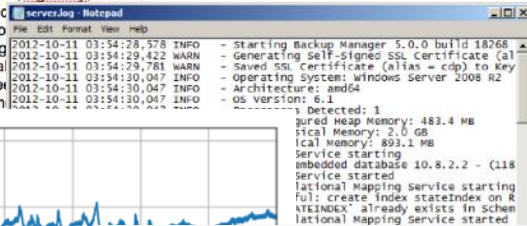
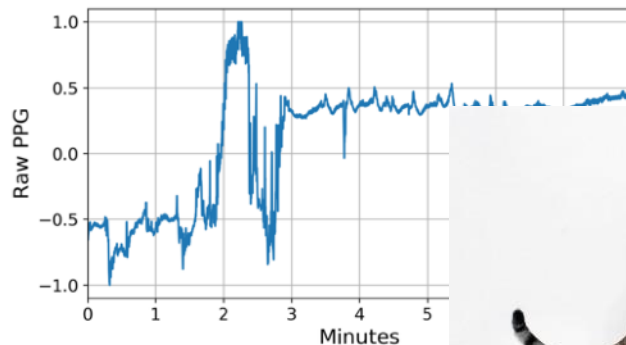
Structured

- 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 were plotted on the same figure. Figure 2 presents ROCs for all the model met or exceeded cardiologist performance.



Structured and Unstructured Data

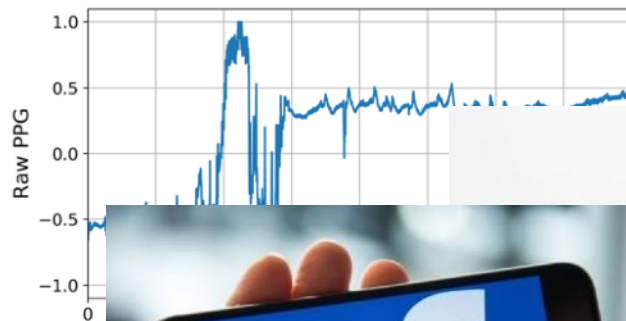
Structured

- 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 and averaged cardiologist ROCs plotted on the same figure. Figure 2 presents ROCs for all the models met or exceeded by the cardiologist performance.



```
server.log - Notepad
File Edit Format View Help
2012-10-11 03:54:28,578 INFO - Starting Backup Manager 5.0.0 build 18268
2012-10-11 03:54:29,422 WARN - Generating self-signed SSL Certificate (al
2012-10-11 03:54:29,781 WARN - Saved SSL Certificate (alias = cdp) to Key
2012-10-11 03:54:30,047 INFO - Operating System: Windows Server 2008 R2
2012-10-11 03:54:30,047 INFO - Architecture: amd64
2012-10-11 03:54:30,047 INFO - OS version: 6.1
2012-10-11 03:54:30,047 INFO - Detected: 1
2012-10-11 03:54:30,047 INFO - Physical Memory: 483.4 MB
2012-10-11 03:54:30,047 INFO - Virtual Memory: 2.0 GB
2012-10-11 03:54:30,047 INFO - Local Memory: 893.1 MB
2012-10-11 03:54:30,047 INFO - Service starting
2012-10-11 03:54:30,047 INFO - Embedded database 10.8.2.2 - (118
2012-10-11 03:54:30,047 INFO - Service started
2012-10-11 03:54:30,047 INFO - National Mapping Service starting
2012-10-11 03:54:30,047 INFO - Full: create index stateindex on R
2012-10-11 03:54:30,047 INFO - STATEINDEX already exists in schem
2012-10-11 03:54:30,047 INFO - National Mapping Service started
```

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. Figure 2 presents ROCs for all the model met or exceeded the cardiologist performance.



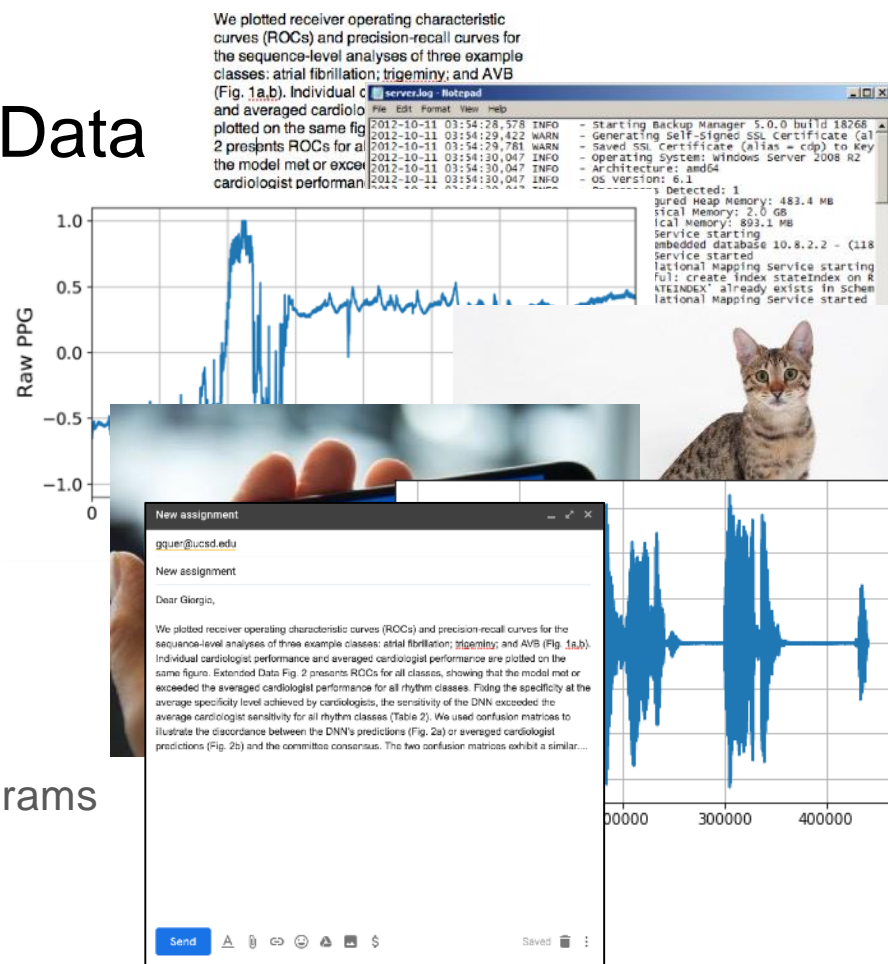
Structured and Unstructured Data

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')]
```

NLTK

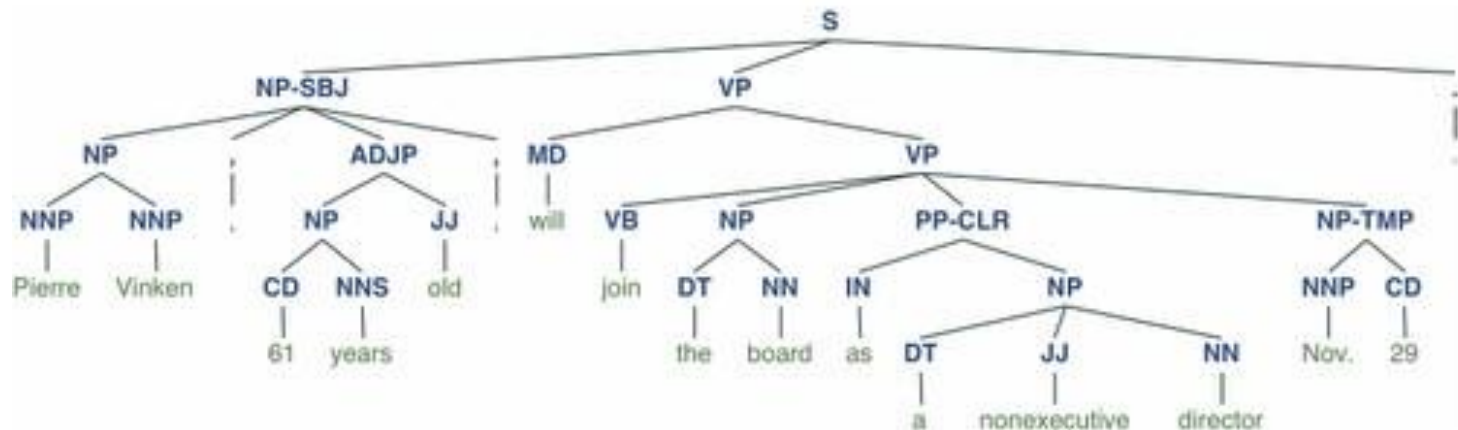
Identify named entities:

```
>>> 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