UNIVERSITY OF SAN FRANCISCO
CHANGE THE WORLD FROM HERE

Project title:

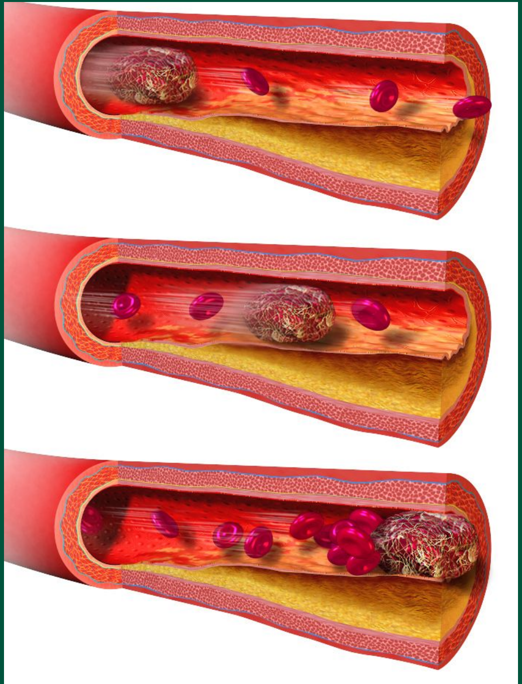
Detecting true cases of Pulmonary Embolism in MIMIC-III dataset.

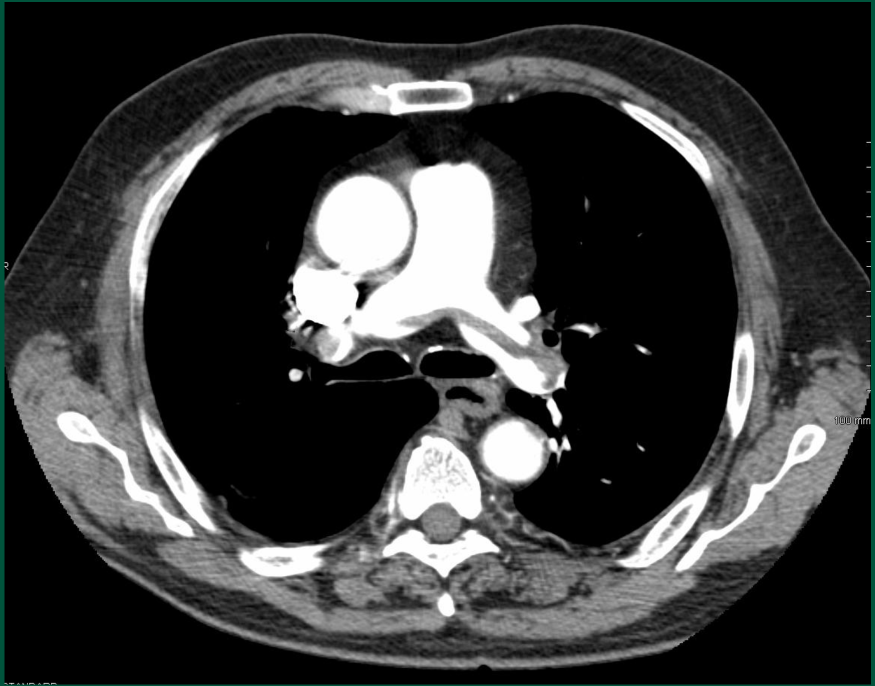
Instructor: Andrew Nguyen

Student: Omid Khazaie

HS 619: Natural Language Processing
Fall 2018

What is Pulmonary Embolism (PE)?





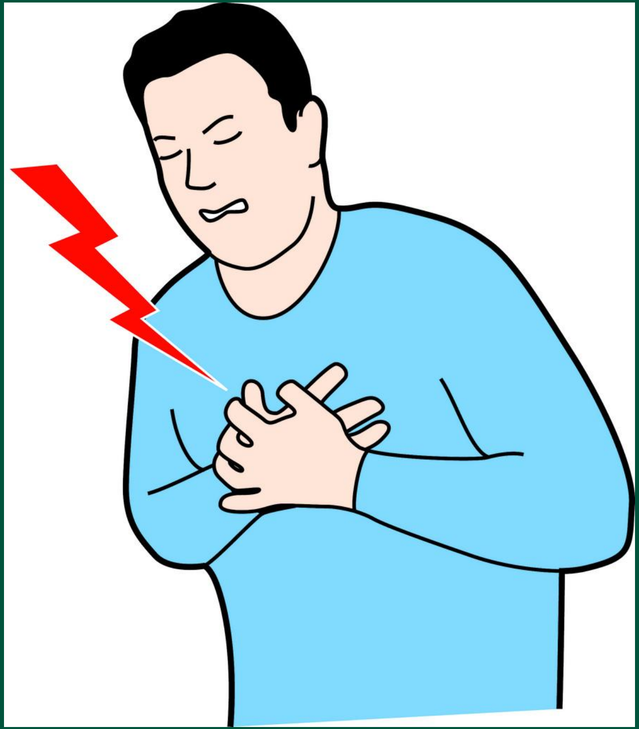
PE Symptoms

Common signs and symptoms include:

- Shortness of breath
- Chest pain
- Cough

Other signs and symptoms:

- Leg pain or swelling, or both, usually in the calf
- Clammy or discolored skin
- Fever
- Excessive sweating
- Rapid or irregular heartbeat
- Lightheadedness or dizziness [2]



Risk factors:

Medical history:
Family members history of venous blood clots or pulmonary embolism

Medical conditions and treatments:

- Heart disease
- Cancer
- Surgery

Prolonged immobility:

- Bed rest
- Long trips

Other risk factors:

- Smoking
- Being overweight
- Supplemental estrogen
- Pregnancy

Issues and Background

Recent study by patient safety experts at Johns Hopkins Medicine in Baltimore:

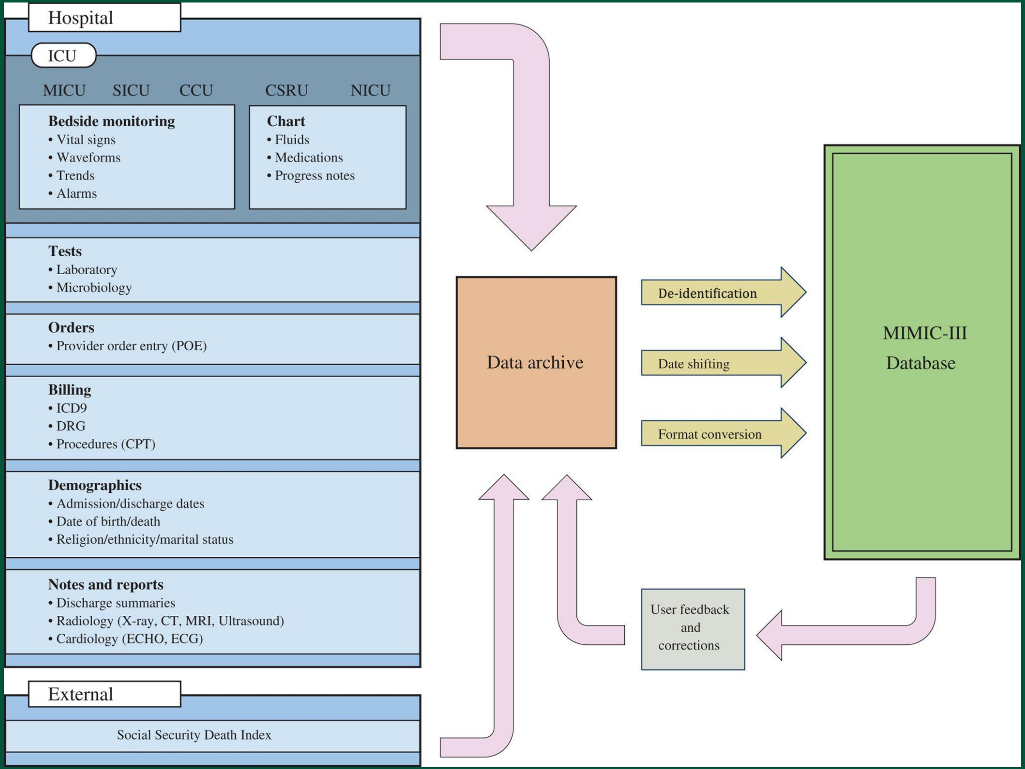
- 40,500 ICU adult patients a year die with an unknown medical condition
- Doctors receive about 7,000 pieces of information a day in this complex, distracting environment
- Misdiagnoses also occur frequently in emergency rooms, where doctors are scrambling to decide whether patients should be admitted to the hospital or sent home.
- Just five conditions account for more than one-third of all missed diagnoses in the ICU

5 Most Commonly Misdiagnosed Conditions in the ICU:

1. Heart attack
2. Pulmonary embolism
3. Pneumonia
4. Aspergillosis
5. Abdominal bleeding

Dataset: MIMIC-III

- MIMIC is an openly available dataset developed by the MIT Lab for Computational Physiology
- Comprising de-identified health data associated with ~40,000 critical care patients.
- 26 tables
  - Track patient stays
  - Data collected in the critical care unit
  - Data collected in the hospital record system
  - Dictionaries



Data Extraction

noteevents table:

- Number of rows: 2,083,180
- TEXT is often large and contains many newline characters
- Some reports are available for both inpatient and outpatient stays
- If a patient is an outpatient, there will not be an HADM_ID associated with the note

Processing all the notes?

- Computationally expensive
- Not required for this project

How to extract data?

- Use ICD codes (billing code) to filter cases with PE or similar conditions
- Filter Discharge summary notes

Data Extraction

diagnoses_icd table:

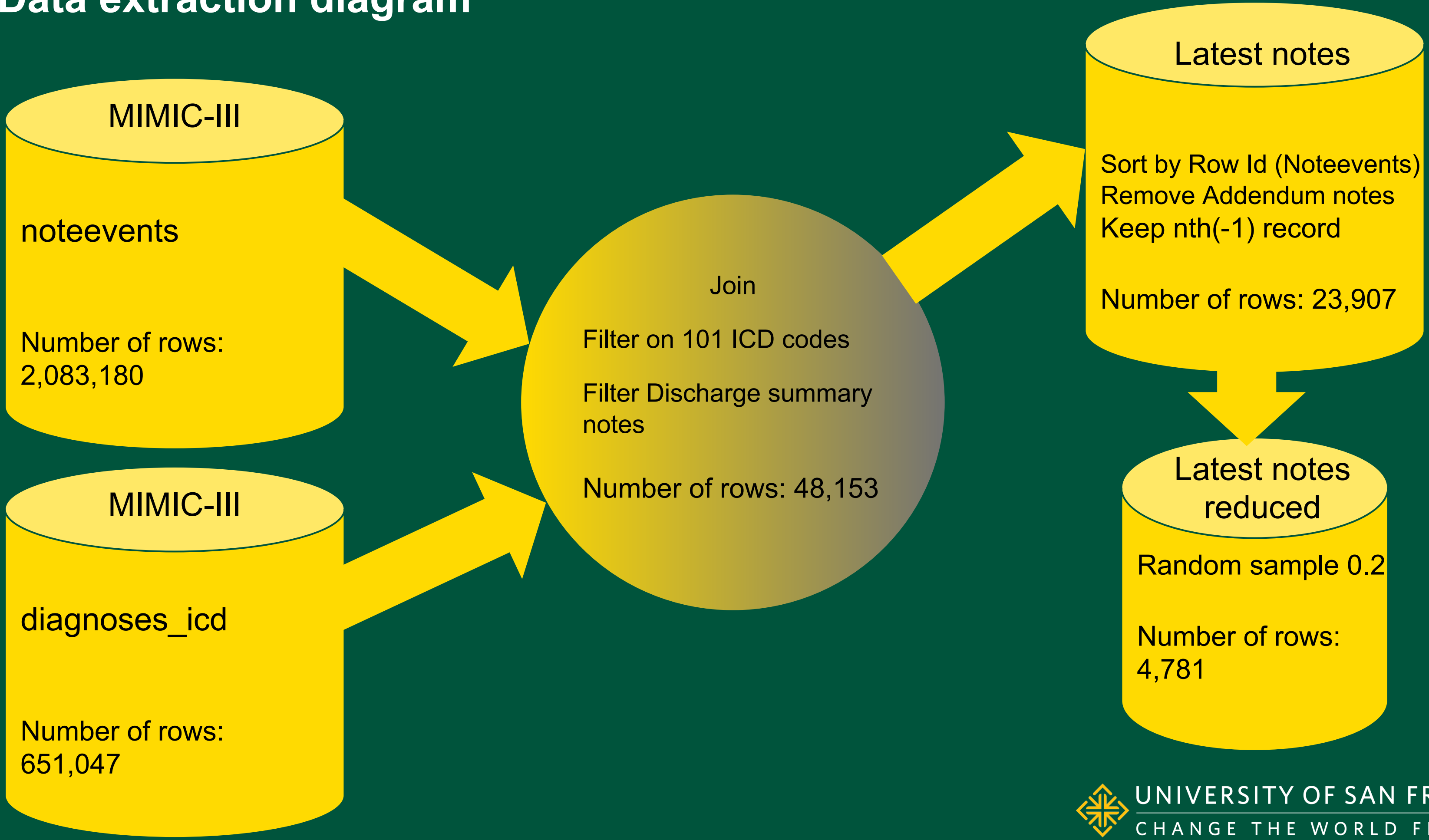
- The ICD codes are generated for billing purposes at the end of the hospital stay.
- All ICD codes in MIMIC-III are ICD-9 based
- Number of rows: 651,047
- SEQ_NUM: provides the order in which the ICD diagnoses relate to the patient.

ICD codes used to extract data:

- Asthma, Pneumonia, Bronchitis, Heart attack
- Total of 101 ICD codes

Use the most recent note for each patient

Data extraction diagram



MIMIC-III

noteevents

Number of rows: 2,083,180

MIMIC-III

diagnoses_icd

Number of rows: 651,047

Join

Filter on 101 ICD codes

Filter Discharge summary notes

Number of rows: 48,153

Latest notes

Sort by Row Id (Noteevents)
Remove Addendum notes
Keep nth(-1) record

Number of rows: 23,907

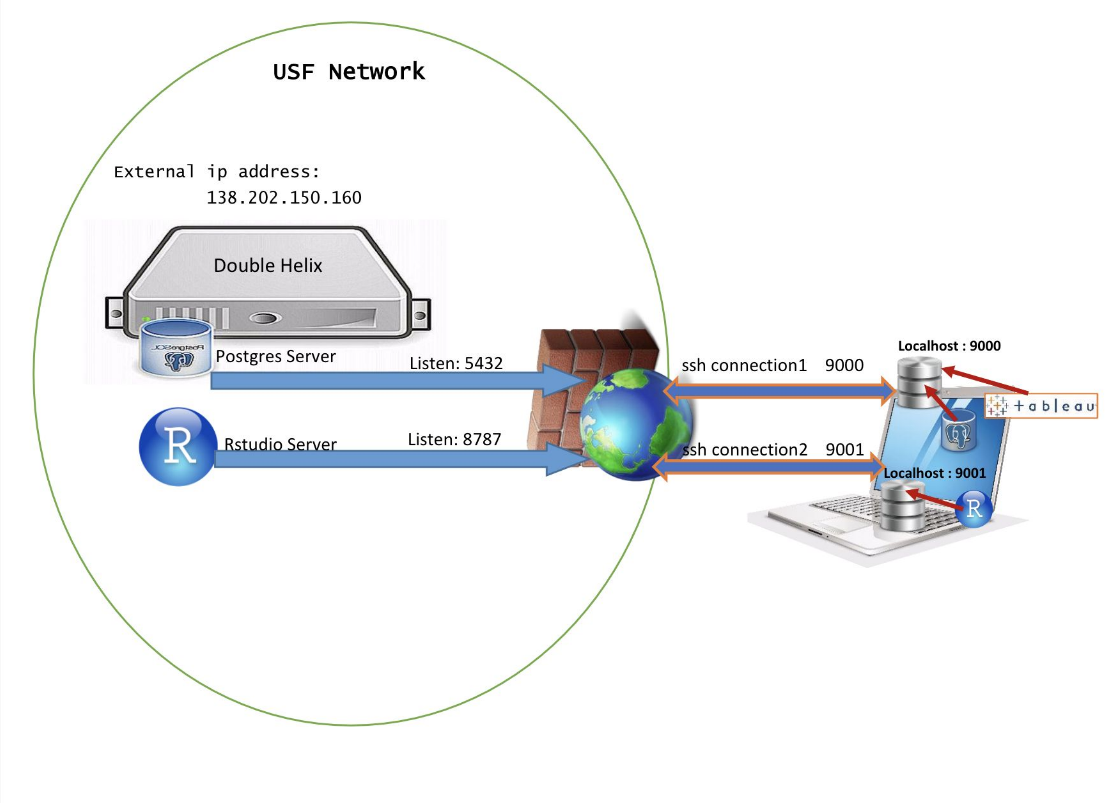
Latest notes reduced

Random sample 0.2

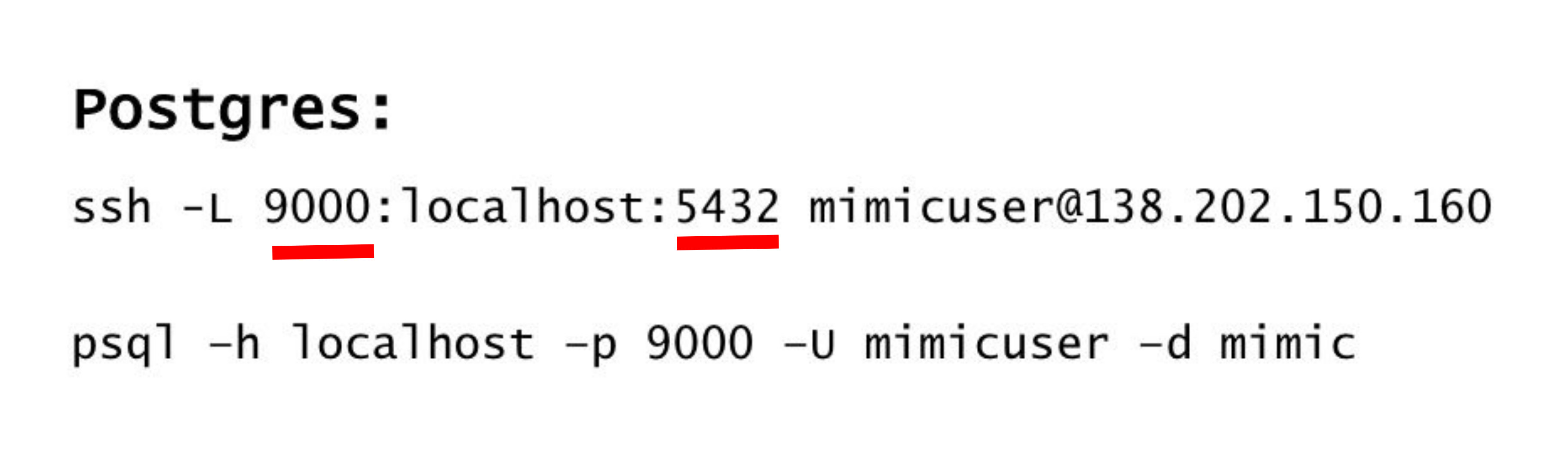
Number of rows: 4,781

Data extraction tools and techniques

- Postgres server
- pgAdmin
- Tableau
- IntelliJ
- R
- Spark
- Python



Remote ssh connection

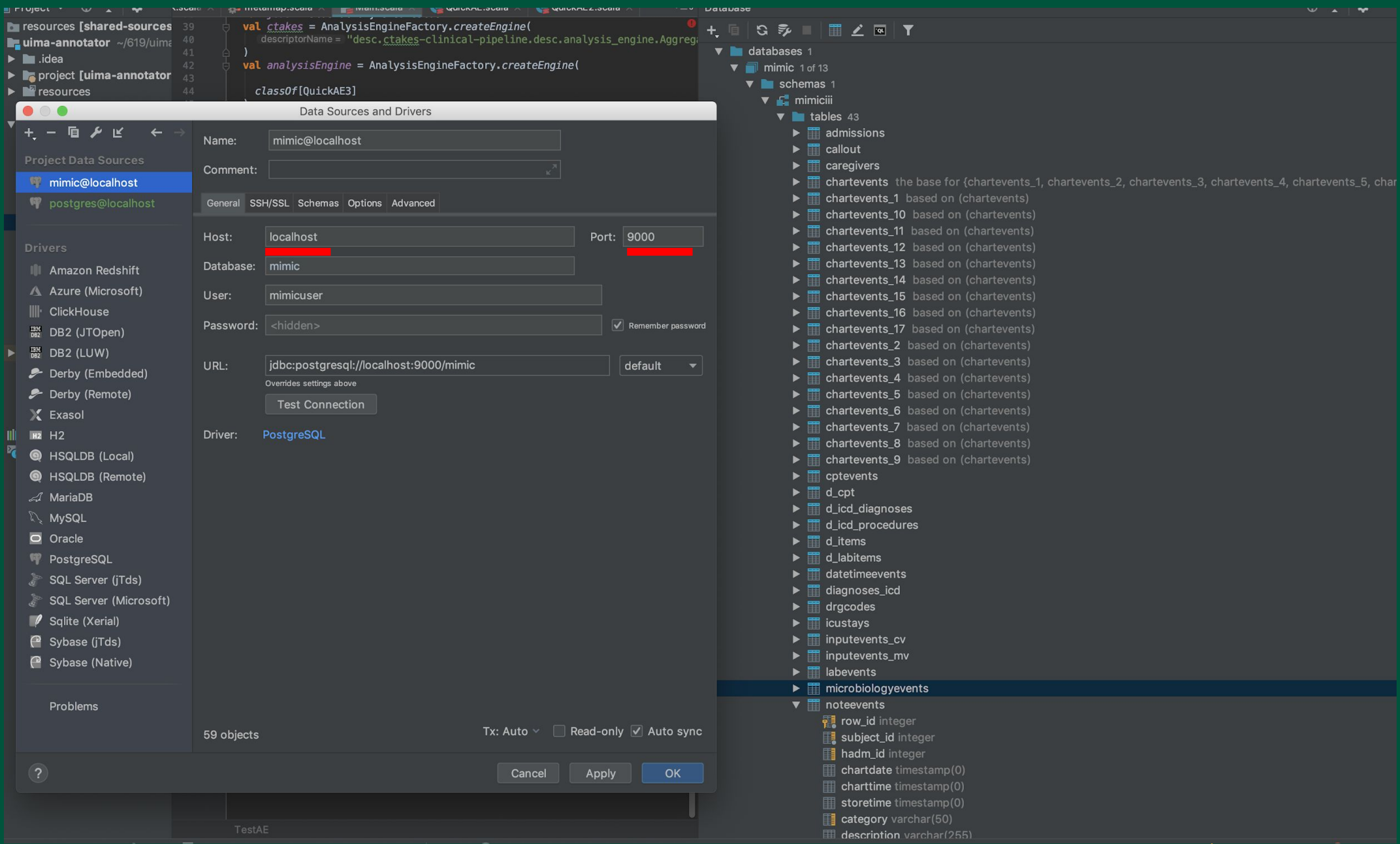


Postgres:

```
ssh -L 9000:localhost:5432 mimicuser@138.202.150.160

psql -h localhost -p 9000 -U mimicuser -d mimic
```

IntelliJ



resources [shared-sources
uima-annotator ~/619/uima
.idea
project [uima-annotator
resources

```
val ctakes = AnalysisEngineFactory.createEngine(
  descriptorName = "desc.ctakes-clinical-pipeline.desc.analysis_engine.Aggrega
)
val analysisEngine = AnalysisEngineFactory.createEngine(

  classOf[QuickAE3]
```

Data Sources and Drivers

Project Data Sources
- mimic@localhost
- postgres@localhost

Drivers
- Amazon Redshift
- Azure (Microsoft)
- ClickHouse
- DB2 (JTOpen)
- DB2 (LUW)
- Derby (Embedded)
- Derby (Remote)
- Exasol
- H2
- HSQLDB (Local)
- HSQLDB (Remote)
- MariaDB
- MySQL
- Oracle
- PostgreSQL
- SQL Server (jTds)
- SQL Server (Microsoft)
- Sqlite (Xerial)
- Sybase (jTds)
- Sybase (Native)

Problems

Name: mimic@localhost

Comment:

General | SSH/SSL | Schemas | Options | Advanced

Host: localhost

Port: 9000

Database: mimic

User: mimicuser

Password: <hidden>

☑ Remember password

URL: jdbc:postgresql://localhost:9000/mimic

default

Overrides settings above

Test Connection

Driver: PostgreSQL

59 objects

Tx: Auto

☐ Read-only ☑ Auto sync

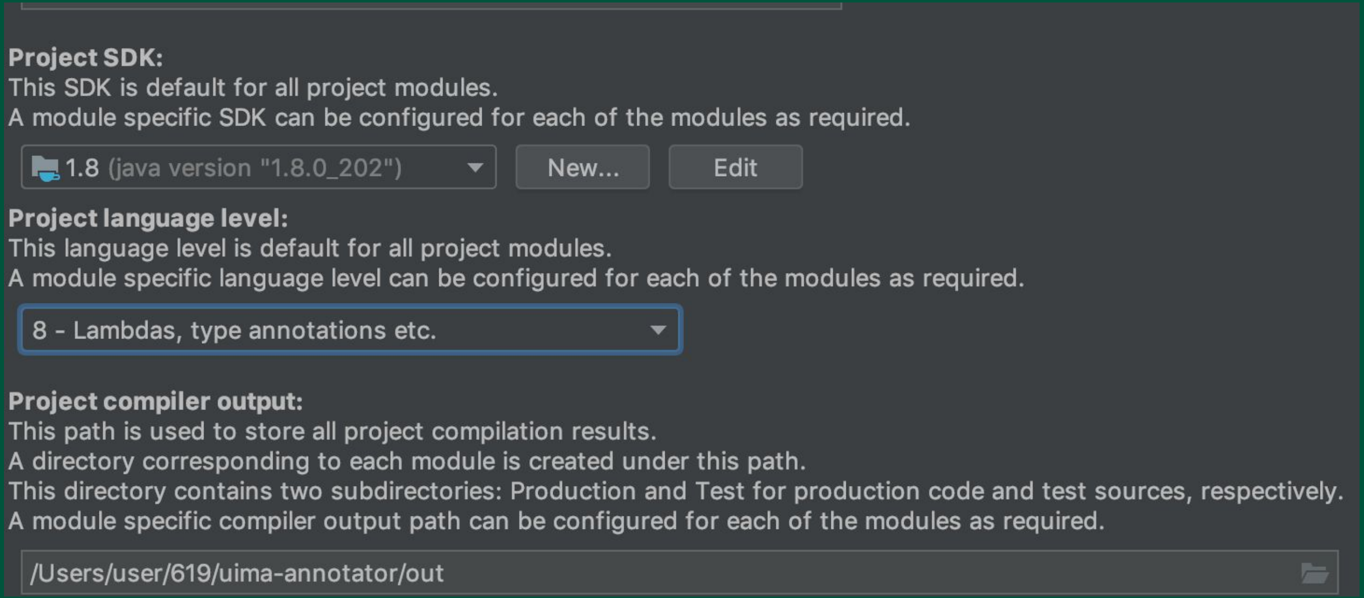
Cancel | Apply | OK

Database
- databases 1
  - mimic 1 of 13
    - schemas 1
      - mimiciii
        - tables 43
          - admissions
          - callout
          - caregivers
          - chartevents the base for {chartevents_1, chartevents_2, chartevents_3, chartevents_4, chartevents_5, char
          - chartevents_1 based on (chartevents)
          - chartevents_10 based on (chartevents)
          - chartevents_11 based on (chartevents)
          - chartevents_12 based on (chartevents)
          - chartevents_13 based on (chartevents)
          - chartevents_14 based on (chartevents)
          - chartevents_15 based on (chartevents)
          - chartevents_16 based on (chartevents)
          - chartevents_17 based on (chartevents)
          - chartevents_2 based on (chartevents)
          - chartevents_3 based on (chartevents)
          - chartevents_4 based on (chartevents)
          - chartevents_5 based on (chartevents)
          - chartevents_6 based on (chartevents)
          - chartevents_7 based on (chartevents)
          - chartevents_8 based on (chartevents)
          - chartevents_9 based on (chartevents)
          - cptevents
          - d_cpt
          - d_icd_diagnoses
          - d_icd_procedures
          - d_items
          - d_labitems
          - datetimeevents
          - diagnoses_icd
          - drgcodes
          - icustays
          - inputevents_cv
          - inputevents_mv
          - labevents
          - microbiologyevents
          - noteevents
            - row_id integer
            - subject_id integer
            - hadm_id integer
            - chartdate timestamp(0)
            - charttime timestamp(0)
            - storetime timestamp(0)
            - category varchar(50)
            - description varchar(255)

TestAE

Spark set-up and challenges

- Java 8
- Docker set-up
- Dependency hell



Project SDK:
This SDK is default for all project modules.
A module specific SDK can be configured for each of the modules as required.

1.8 (java version "1.8.0_202")

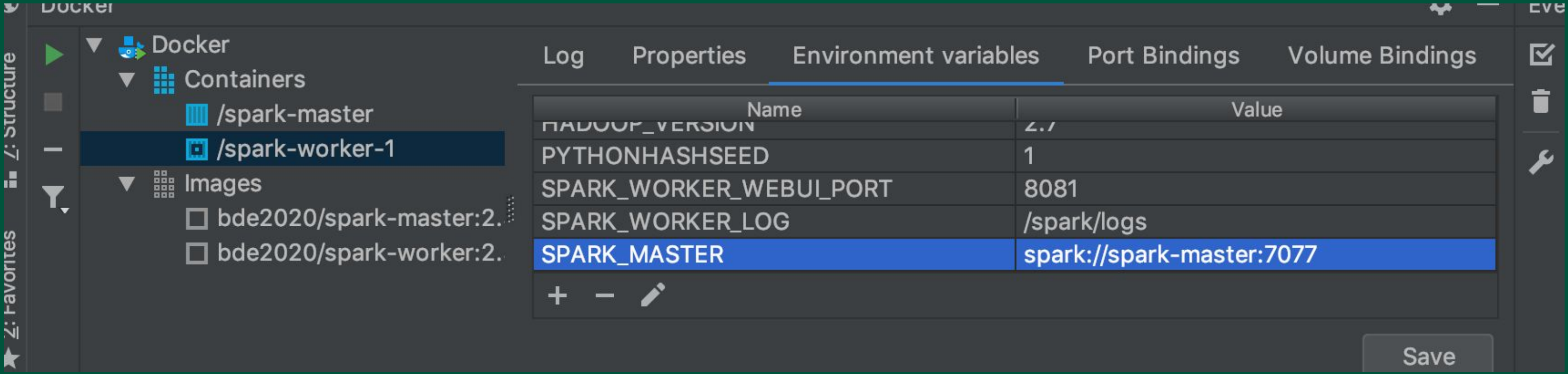
Project language level:
This language level is default for all project modules.
A module specific language level can be configured for each of the modules as required.

8 - Lambdas, type annotations etc.

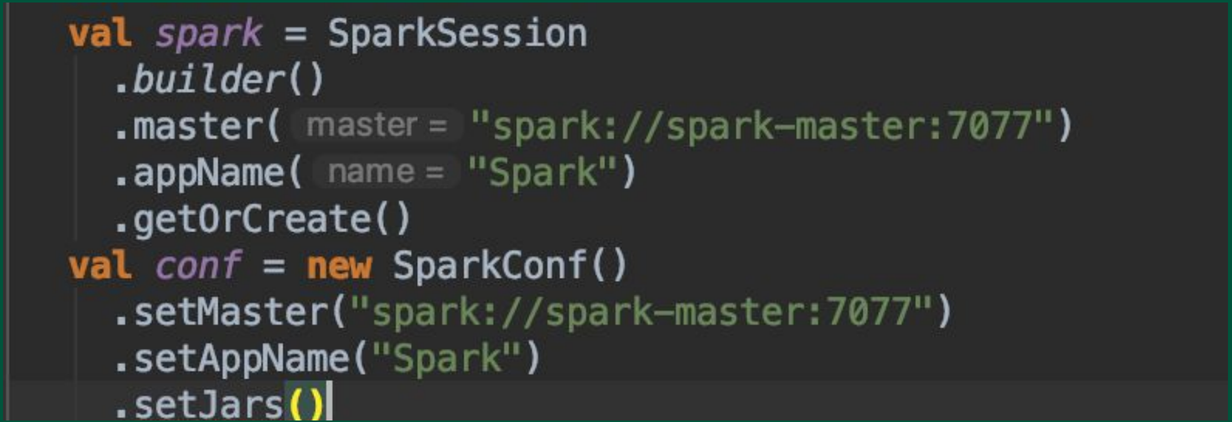
Project compiler output:
This path is used to store all project compilation results.
A directory corresponding to each module is created under this path.
This directory contains two subdirectories: Production and Test for production code and test sources, respectively.
A module specific compiler output path can be configured for each of the modules as required.

/Users/user/619/uima-annotator/out

Docker set-up



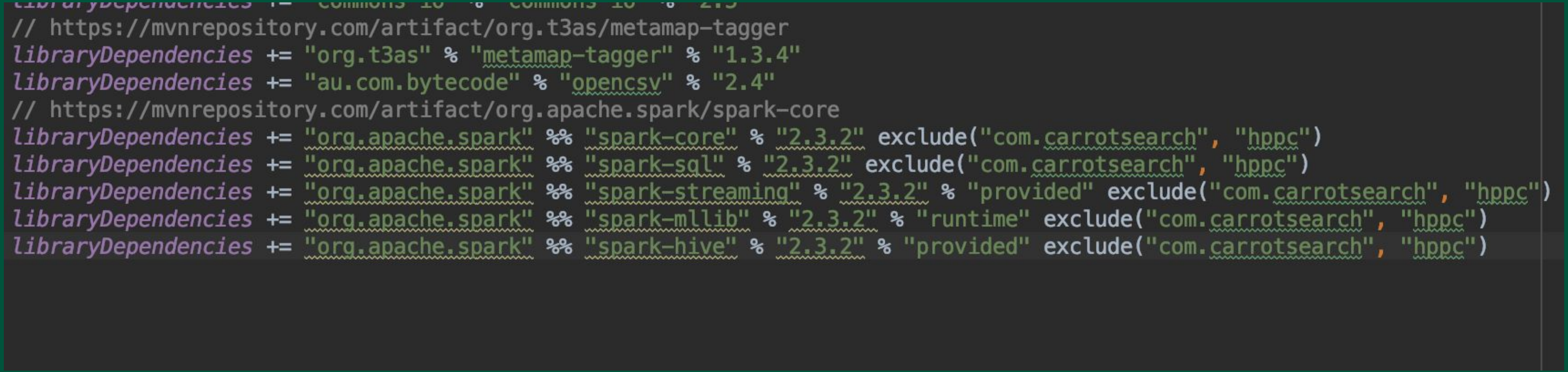
Name	Value
HADOOP_VERSION	2.7
PYTHONHASHSEED	1
SPARK_WORKER_WEBUI_PORT	8081
SPARK_WORKER_LOG	/spark/logs
SPARK_MASTER	spark://spark-master:7077



```scala
val spark = SparkSession
  .builder()
  .master( master = "spark://spark-master:7077")
  .appName( name = "Spark")
  .getOrCreate()
val conf = new SparkConf()
  .setMaster("spark://spark-master:7077")
  .setAppName("Spark")
  .setJars()
```

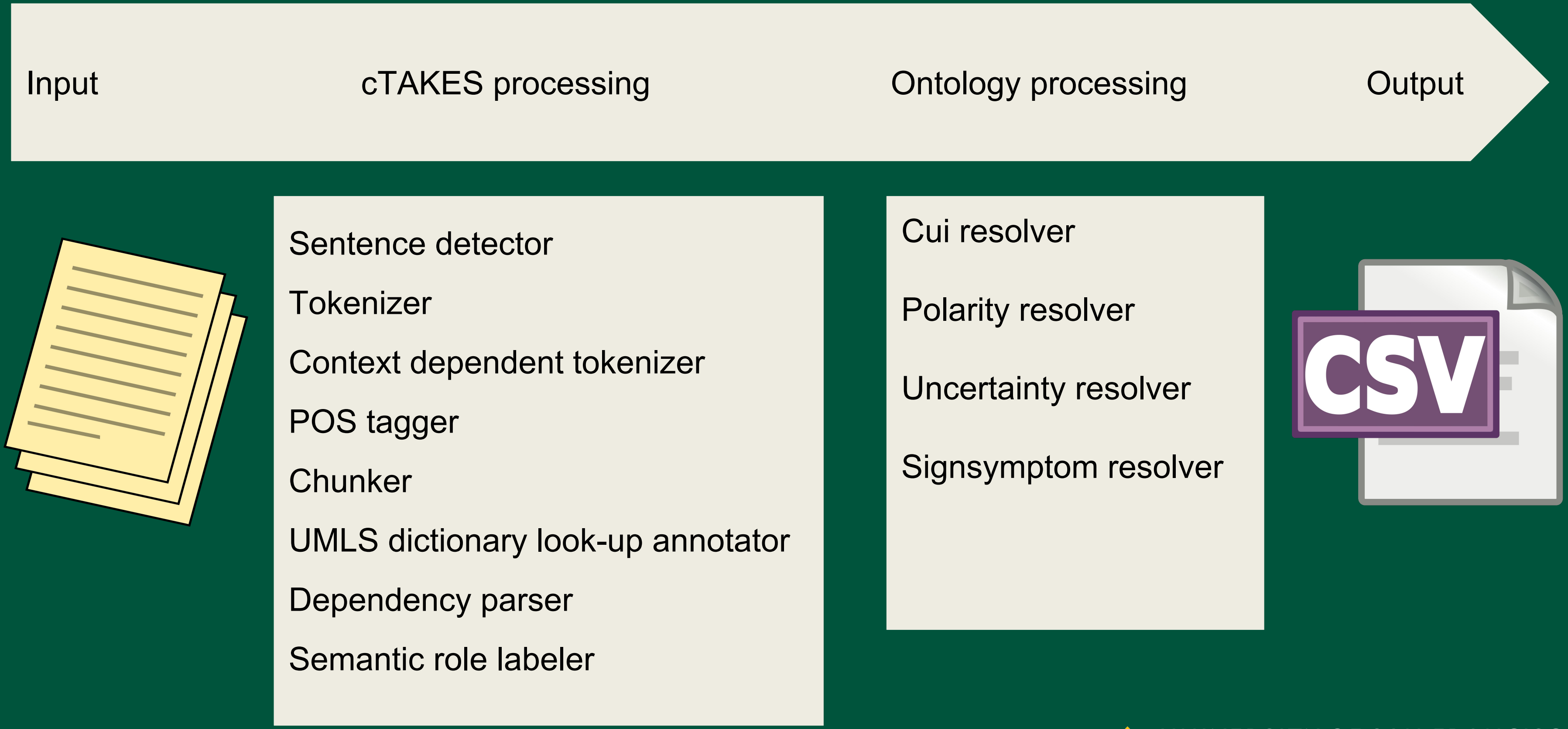
Dependency hell

cTAKES and Spark both were built on carrotsearch but Spark is using the older version of carrotsearch



```
libraryDependencies += "commons-io" % "commons-io" % "2.5"
// https://mvnrepository.com/artifact/org.t3as/metamap-tagger
libraryDependencies += "org.t3as" % "metamap-tagger" % "1.3.4"
libraryDependencies += "au.com.bytecode" % "opencsv" % "2.4"
// https://mvnrepository.com/artifact/org.apache.spark/spark-core
libraryDependencies += "org.apache.spark" %% "spark-core" % "2.3.2" exclude("com.carrotsearch", "hppc")
libraryDependencies += "org.apache.spark" %% "spark-sql" % "2.3.2" exclude("com.carrotsearch", "hppc")
libraryDependencies += "org.apache.spark" %% "spark-streaming" % "2.3.2" % "provided" exclude("com.carrotsearch", "hppc")
libraryDependencies += "org.apache.spark" %% "spark-mllib" % "2.3.2" % "runtime" exclude("com.carrotsearch", "hppc")
libraryDependencies += "org.apache.spark" %% "spark-hive" % "2.3.2" % "provided" exclude("com.carrotsearch", "hppc")
```

NLP pipeline



Input

cTAKES processing

Ontology processing

Output

Sentence detector

Tokenizer

Context dependent tokenizer

POS tagger

Chunker

UMLS dictionary look-up annotator

Dependency parser

Semantic role labeler

Cui resolver

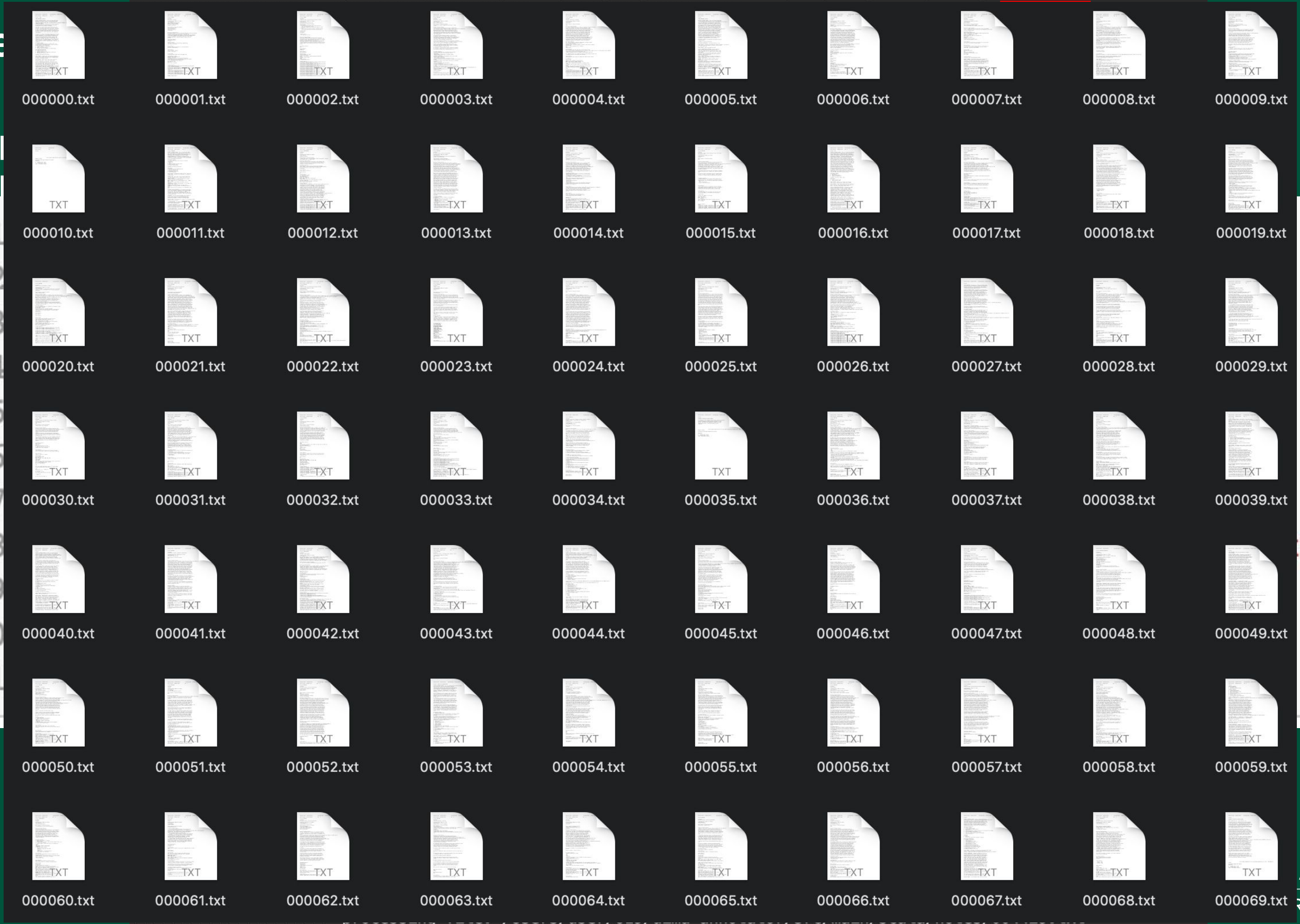
Polarity resolver

Uncertainty resolver

Signsymptom resolver

Challe

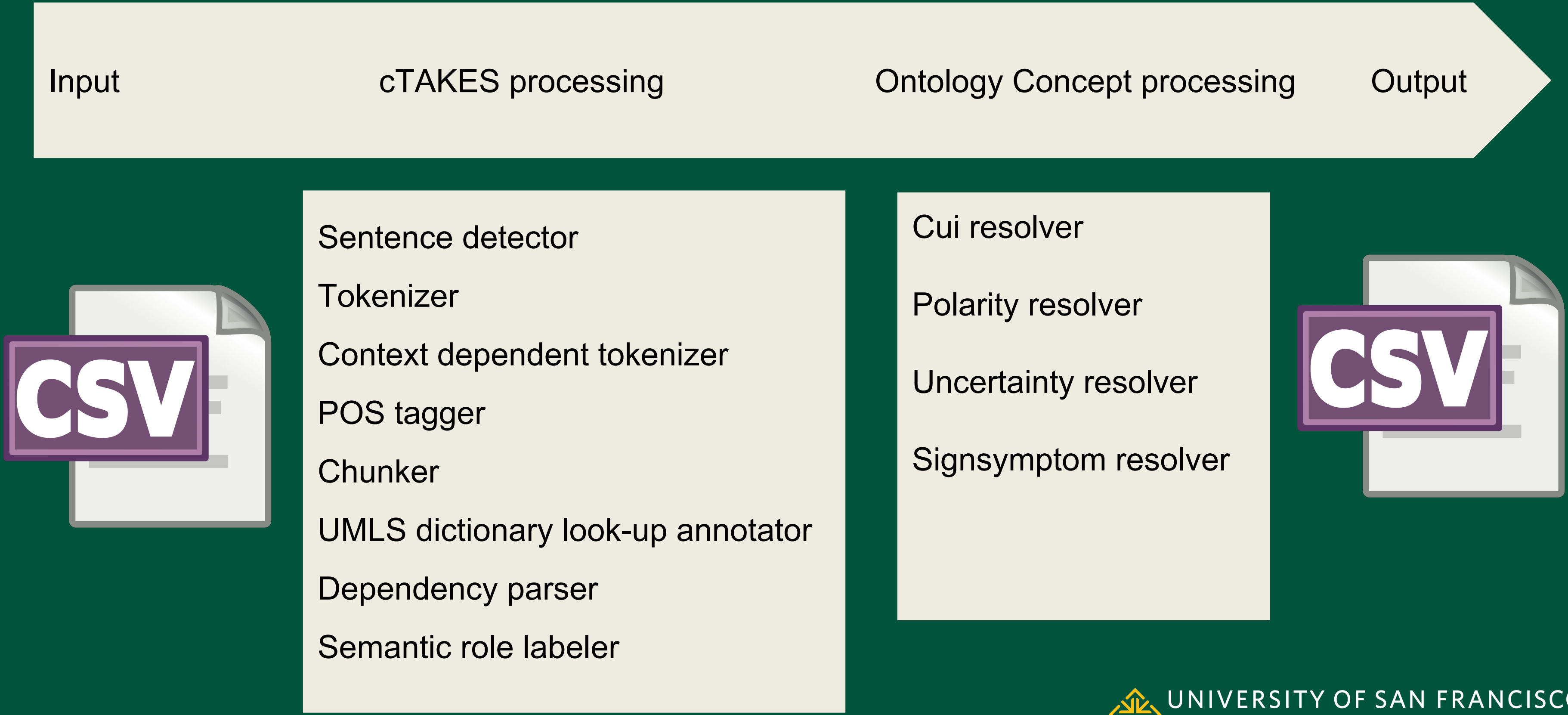
```
>>>>>>>>>>>>>>>>>>>>>> processing file: /Users/user/619/uima-annotator/src/main/scala/notes/003346.txt
```



```
t")
```

```
02 Dec 2018 22:51:53  INFO SentenceDetector - Starting processing.
```

New NLP pipeline



Input

cTAKES processing

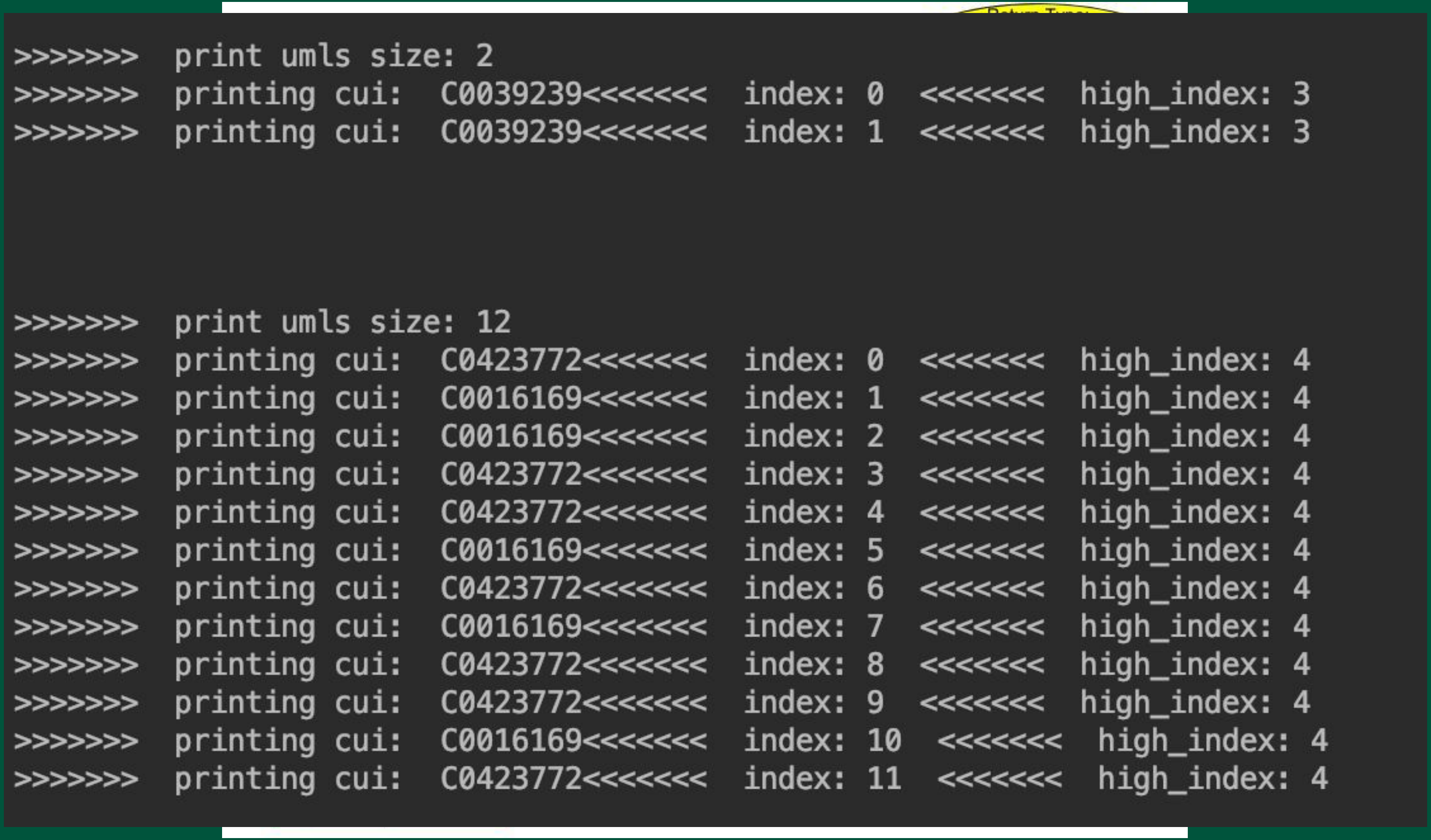
- Sentence detector
- Tokenizer
- Context dependent tokenizer
- POS tagger
- Chunker
- UMLS dictionary look-up annotator
- Dependency parser
- Semantic role labeler

Ontology Concept processing

- Cui resolver
- Polarity resolver
- Uncertainty resolver
- Signsymptom resolver

Output

OntologyConcept processing

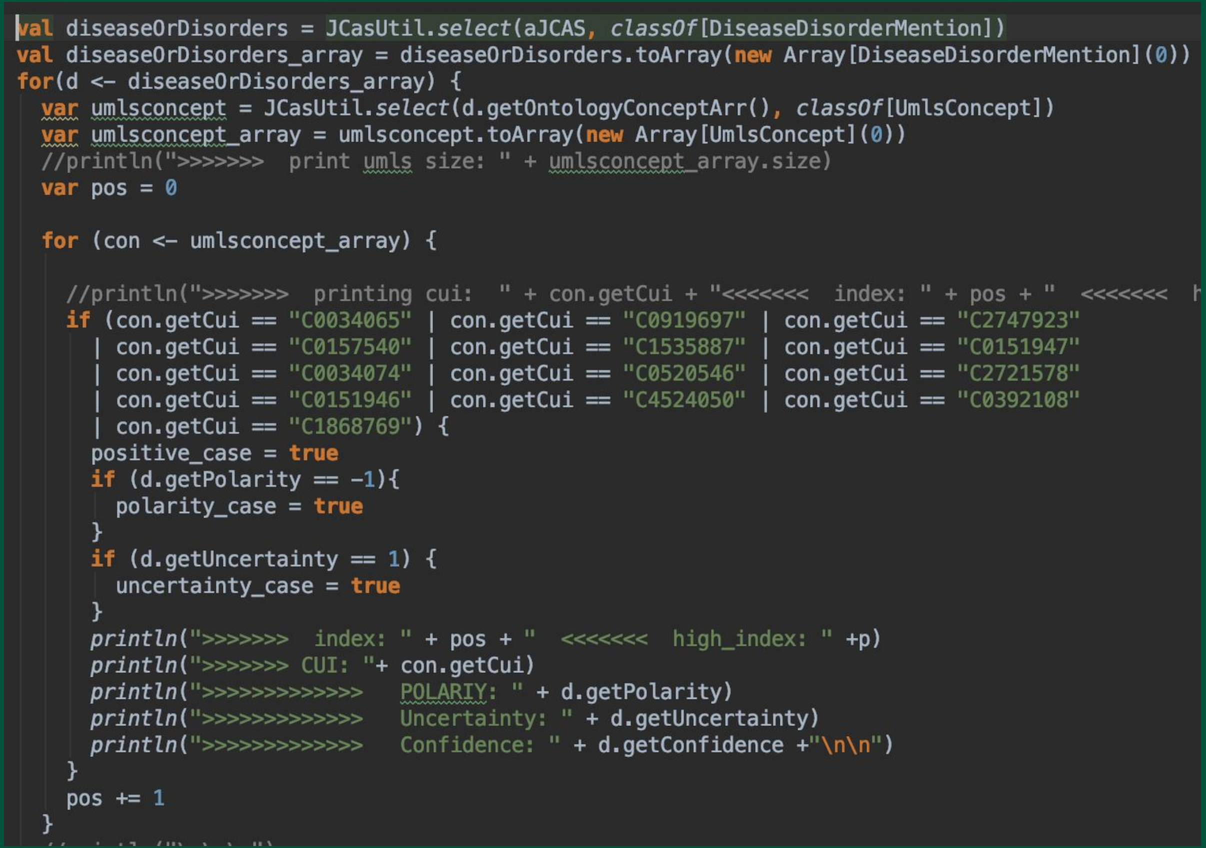


```
>>>>>>>  print umls size: 2
>>>>>>>  printing cui:  C0039239<<<<<<<  index: 0  <<<<<<<  high_index: 3
>>>>>>>  printing cui:  C0039239<<<<<<<  index: 1  <<<<<<<  high_index: 3




>>>>>>>  print umls size: 12
>>>>>>>  printing cui:  C0423772<<<<<<<  index: 0  <<<<<<<  high_index: 4
>>>>>>>  printing cui:  C0016169<<<<<<<  index: 1  <<<<<<<  high_index: 4
>>>>>>>  printing cui:  C0016169<<<<<<<  index: 2  <<<<<<<  high_index: 4
>>>>>>>  printing cui:  C0423772<<<<<<<  index: 3  <<<<<<<  high_index: 4
>>>>>>>  printing cui:  C0423772<<<<<<<  index: 4  <<<<<<<  high_index: 4
>>>>>>>  printing cui:  C0016169<<<<<<<  index: 5  <<<<<<<  high_index: 4
>>>>>>>  printing cui:  C0423772<<<<<<<  index: 6  <<<<<<<  high_index: 4
>>>>>>>  printing cui:  C0016169<<<<<<<  index: 7  <<<<<<<  high_index: 4
>>>>>>>  printing cui:  C0423772<<<<<<<  index: 8  <<<<<<<  high_index: 4
>>>>>>>  printing cui:  C0423772<<<<<<<  index: 9  <<<<<<<  high_index: 4
>>>>>>>  printing cui:  C0016169<<<<<<<  index: 10  <<<<<<<  high_index: 4
>>>>>>>  printing cui:  C0423772<<<<<<<  index: 11  <<<<<<<  high_index: 4
```

OntologyConcept processing

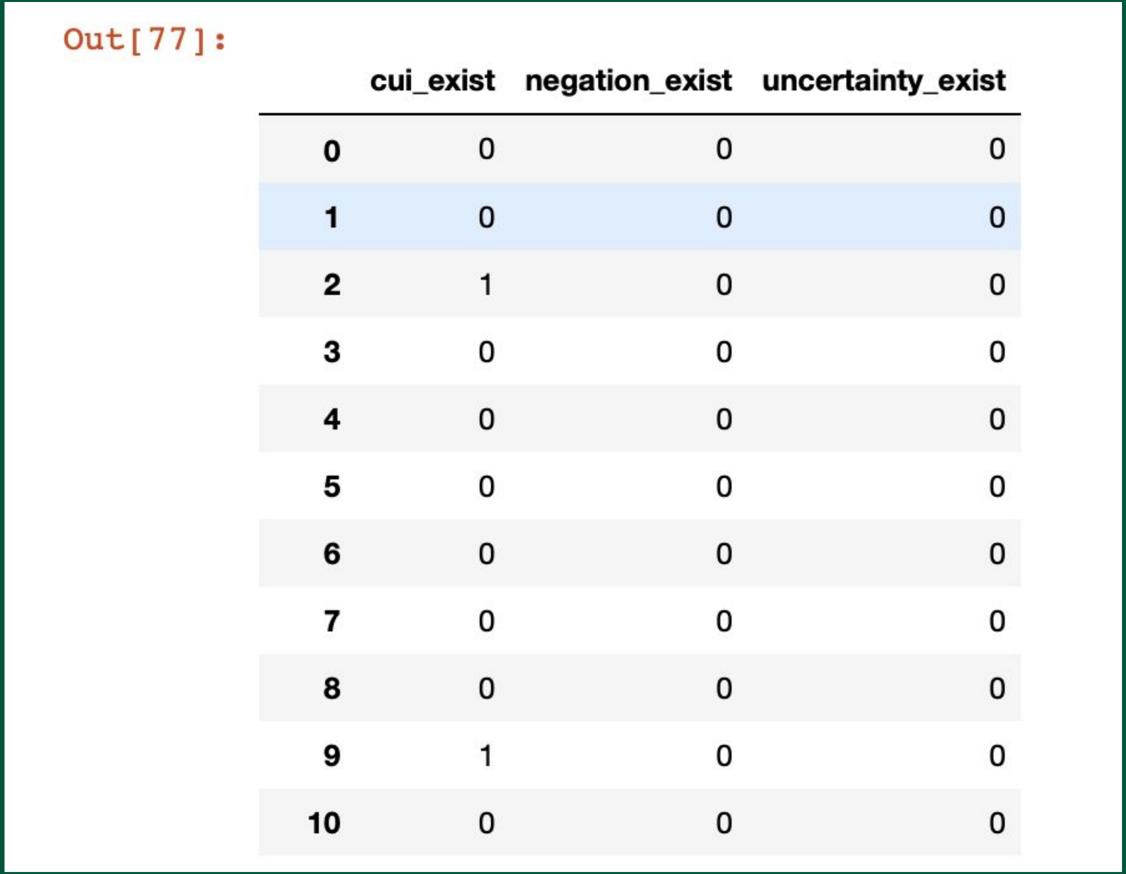


```scala
val diseaseOrDisorders = JCasUtil.select(aJCAS, classOf[DiseaseDisorderMention])
val diseaseOrDisorders_array = diseaseOrDisorders.toArray(new Array[DiseaseDisorderMention](0))
for(d <- diseaseOrDisorders_array) {
  var umlsconcept = JCasUtil.select(d.getOntologyConceptArr(), classOf[UmlsConcept])
  var umlsconcept_array = umlsconcept.toArray(new Array[UmlsConcept](0))
  //println(">>>>>>>  print umls size: " + umlsconcept_array.size)
  var pos = 0

  for (con <- umlsconcept_array) {

    //println(">>>>>>>  printing cui:  " + con.getCui + "<<<<<<<  index: " + pos + "  <<<<<<<  h
    if (con.getCui == "C0034065" | con.getCui == "C0919697" | con.getCui == "C2747923"
      | con.getCui == "C0157540" | con.getCui == "C1535887" | con.getCui == "C0151947"
      | con.getCui == "C0034074" | con.getCui == "C0520546" | con.getCui == "C2721578"
      | con.getCui == "C0151946" | con.getCui == "C4524050" | con.getCui == "C0392108"
      | con.getCui == "C1868769") {
      positive_case = true
      if (d.getPolarity == -1){
        polarity_case = true
      }
      if (d.getUncertainty == 1) {
        uncertainty_case = true
      }
      println(">>>>>>>  index: " + pos + "  <<<<<<<  high_index: " +p)
      println(">>>>>>> CUI: "+ con.getCui)
      println(">>>>>>>>>>>>>   POLARIY: " + d.getPolarity)
      println(">>>>>>>>>>>>>   Uncertainty: " + d.getUncertainty)
      println(">>>>>>>>>>>>>   Confidence: " + d.getConfidence +"\n\n")
    }
    pos += 1
  }
```

Features extracted



Out[77]:

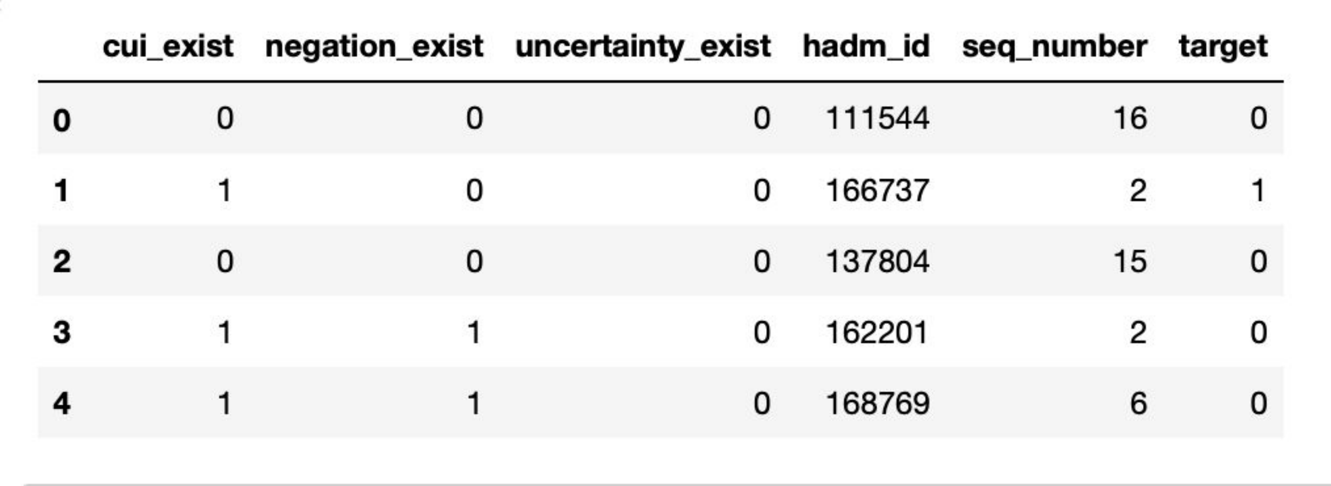
	cui_exist	negation_exist	uncertainty_exist
0	0	0	0
1	0	0	0
2	1	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	1	0	0
10	0	0	0

What is my gold standard?

Machine Learning - Gold Standard

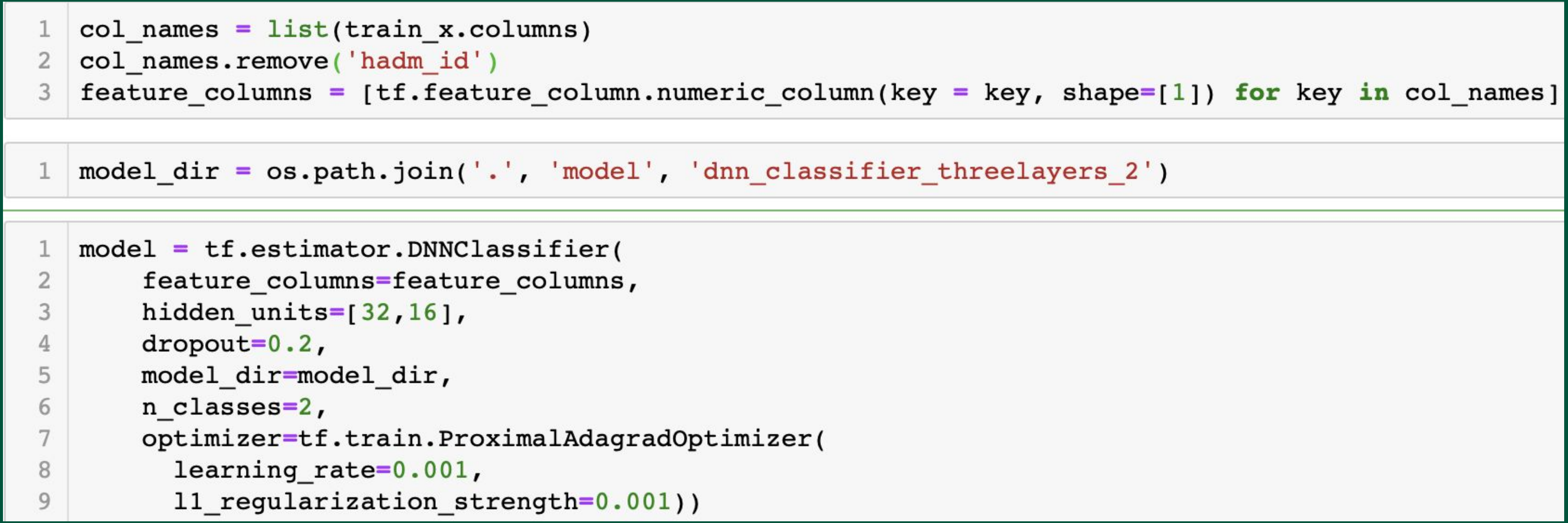
The ICD codes are generated for billing purposes at the end of the hospital stay:

- 4150
- 41511
- 41512
- 41513
- 41519



	cui_exist	negation_exist	uncertainty_exist	hadm_id	seq_number	target
0	0	0	0	111544	16	0
1	1	0	0	166737	2	1
2	0	0	0	137804	15	0
3	1	1	0	162201	2	0
4	1	1	0	168769	6	0

ML Model- DNN

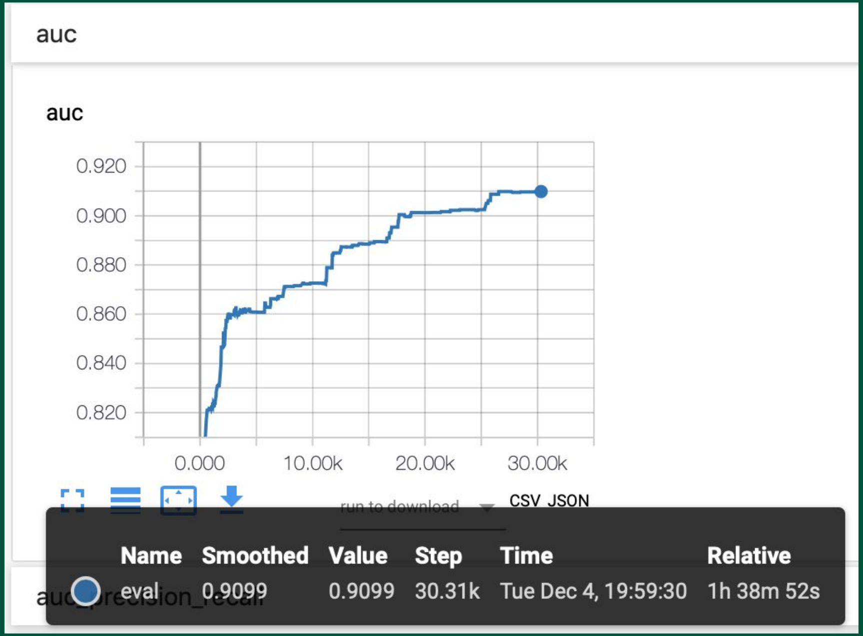


```
col_names = list(train_x.columns)
col_names.remove('hadm_id')
feature_columns = [tf.feature_column.numeric_column(key = key, shape=[1]) for key in col_names]
```

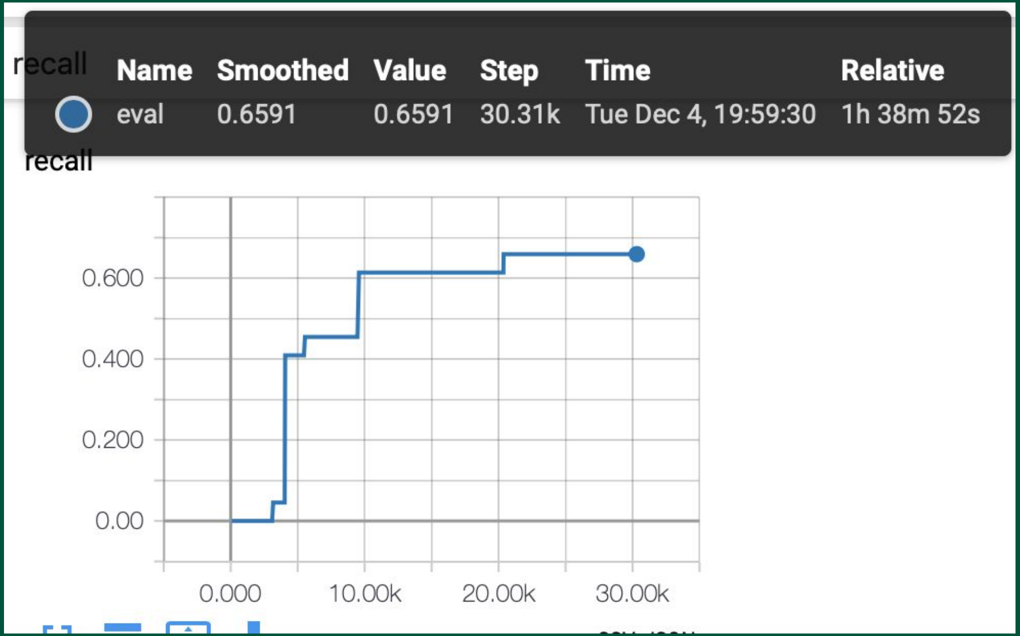
```
model_dir = os.path.join('.', 'model', 'dnn_classifier_threelayers_2')
```

```
model = tf.estimator.DNNClassifier(
    feature_columns=feature_columns,
    hidden_units=[32,16],
    dropout=0.2,
    model_dir=model_dir,
    n_classes=2,
    optimizer=tf.train.ProximalAdagradOptimizer(
      learning_rate=0.001,
      l1_regularization_strength=0.001))
```

Results-AUC

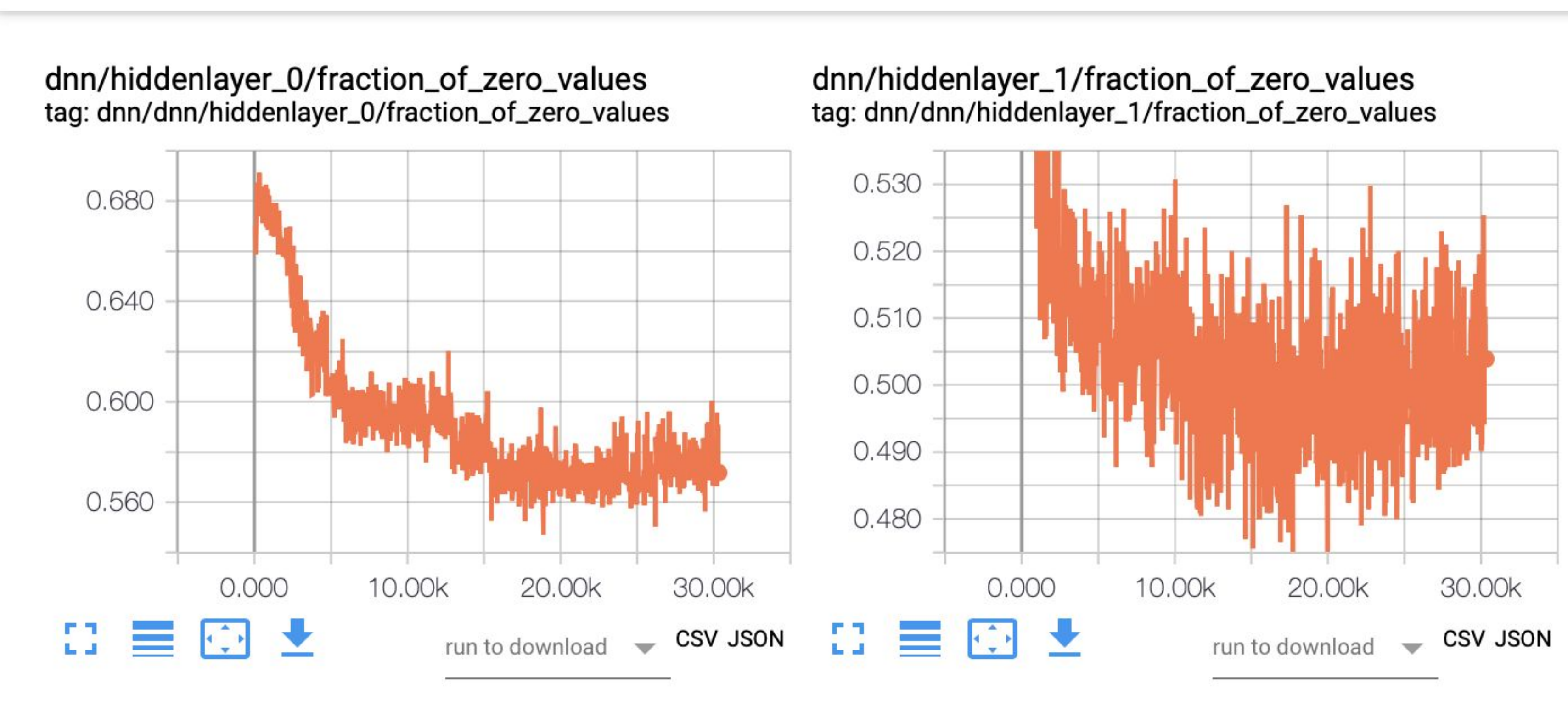


	Name	Smoothed	Value	Step	Time	Relative
●	eval	0.9099	0.9099	30.31k	Tue Dec 4, 19:59:30	1h 38m 52s



	Name	Smoothed	Value	Step	Time	Relative
●	eval	0.6591	0.6591	30.31k	Tue Dec 4, 19:59:30	1h 38m 52s

DNN-Layer activity


dnn/hiddenlayer_0/fraction_of_zero_values
tag: dnn/dnn/hiddenlayer_0/fraction_of_zero_values
0.680
0.640
0.600
0.560
0.000
10.00k
20.00k
30.00k
run to download
CSV JSON
dnn/hiddenlayer_1/fraction_of_zero_values
tag: dnn/dnn/hiddenlayer_1/fraction_of_zero_values
0.530
0.520
0.510
0.500
0.490
0.480
0.000
10.00k
20.00k
30.00k
run to download
CSV JSON

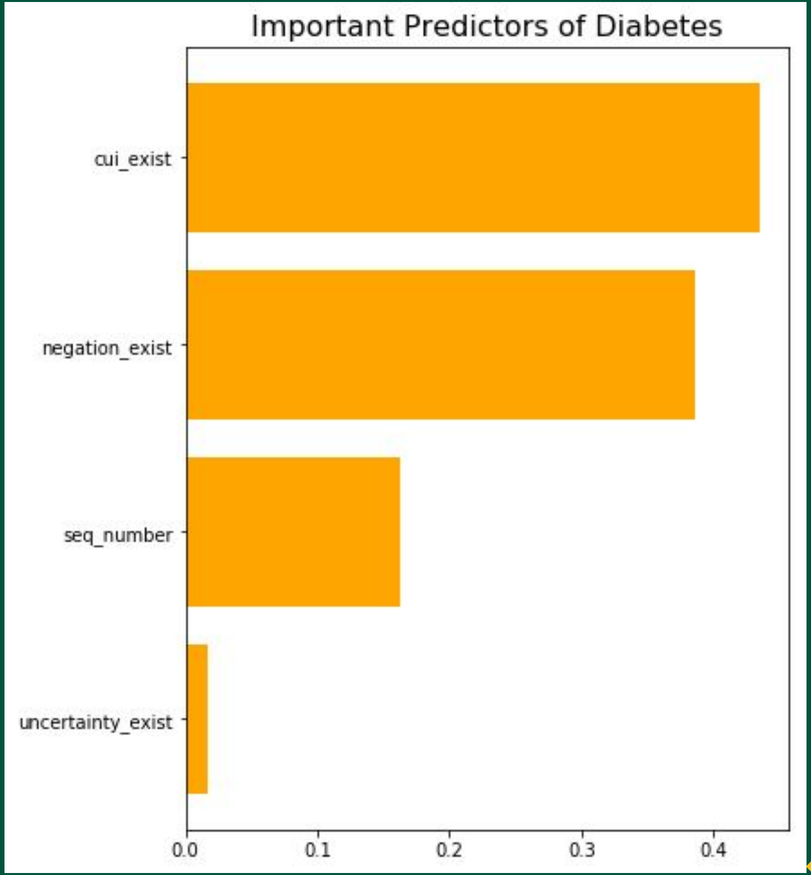
UNIVERSITY OF SAN FRANCISCO
CHANGE THE WORLD FROM HERE

ML-Other models:

Imbalanced data: 0.07 have PE

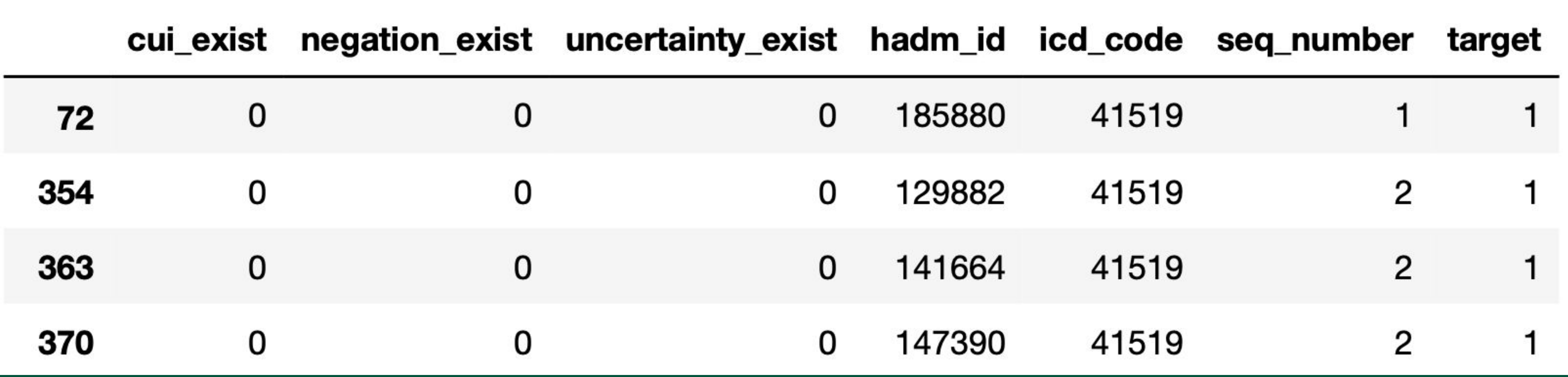
Model	AUC with Negation and Uncertainty	AUC with Negation	AUC without Negation & Uncertainty
DNN	0.92	0.91	0.88
Gaussian Naive Bayes Classifier	0.82	0.82	0.80
Random Forest Classifier	0.79	0.80	0.78

RF Feature Importance



1. Do Negation and Uncertainty improve our model?

2. Does training with gold standard make our result worse?



	cui_exist	negation_exist	uncertainty_exist	hadm_id	icd_code	seq_number	target
72	0	0	0	185880	41519	1	1
354	0	0	0	129882	41519	2	1
363	0	0	0	141664	41519	2	1
370	0	0	0	147390	41519	2	1

Future work

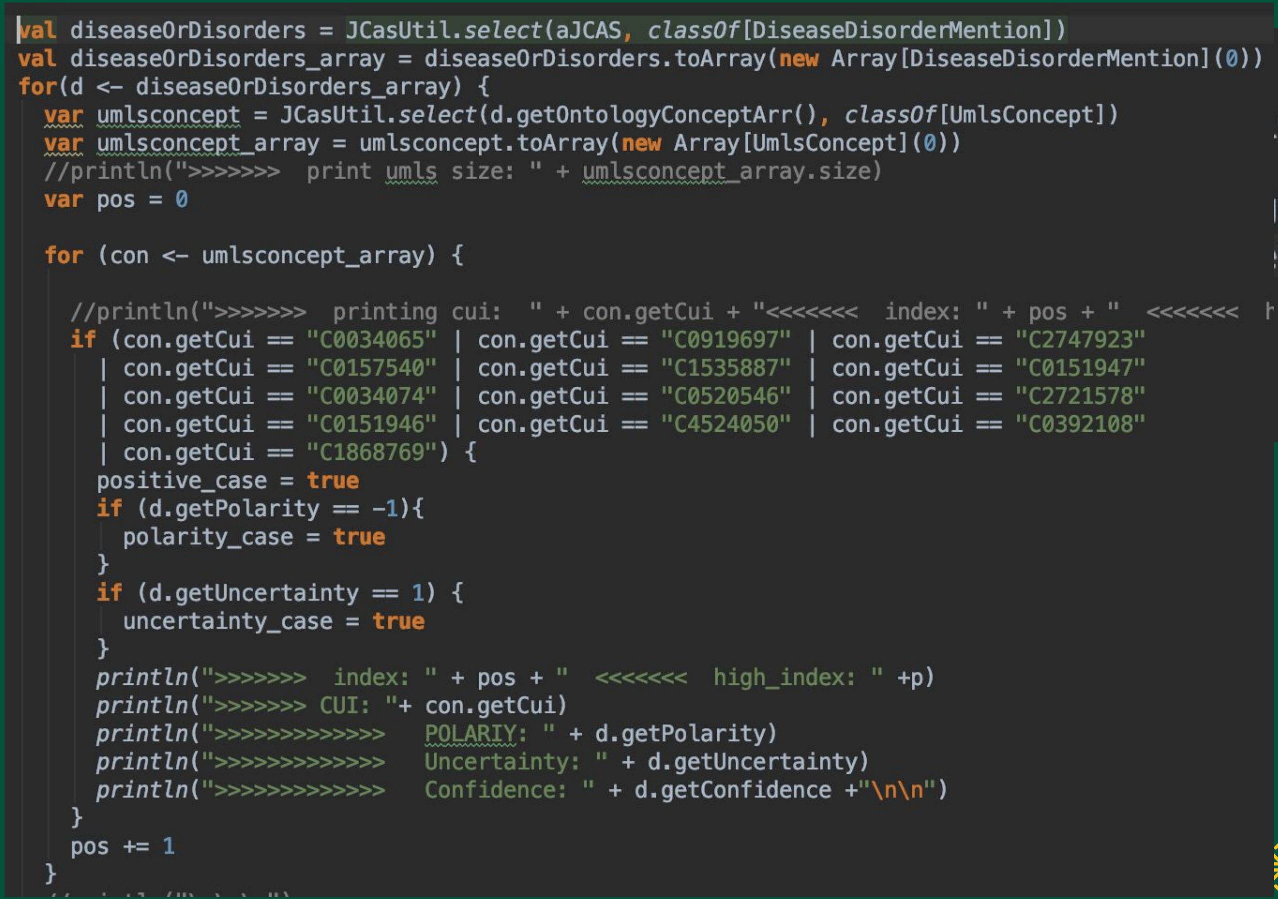
Improve the results:

- Train the model with the gold standard (clinical notes reviewed by medical annotators)
- Extract more features from clinical notes:
    - Number of times cui found in the text
    - Number of time negation found in the text
    - Number of time uncertainty found in the text
    - Bag of cui's for sign and symptoms found in the text

Improve the model:

- Optimize the algorithm
- Try other biomedical semantic annotation tools like MetaMap

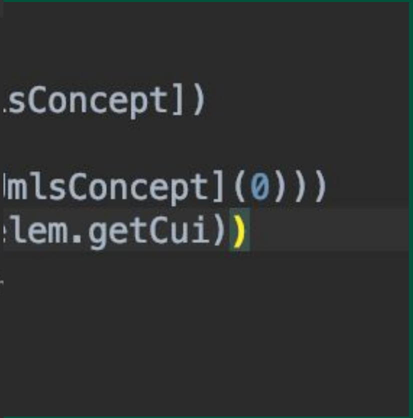
Optimize the algorithm O(n^2) → O(n)



```
val diseaseOrDisorders = JCasUtil.select(aJCAS, classOf[DiseaseDisorderMention])
val diseaseOrDisorders_array = diseaseOrDisorders.toArray(new Array[DiseaseDisorderMention](0))
for(d <- diseaseOrDisorders_array) {
  var umlsconcept = JCasUtil.select(d.getOntologyConceptArr(), classOf[UmlsConcept])
  var umlsconcept_array = umlsconcept.toArray(new Array[UmlsConcept](0))
  //println(">>>>>>>  print umls size: " + umlsconcept_array.size)
  var pos = 0

  for (con <- umlsconcept_array) {

    //println(">>>>>>>  printing cui:  " + con.getCui + "<<<<<<<  index: " + pos + "  <<<<<<<  h
    if (con.getCui == "C0034065" | con.getCui == "C0919697" | con.getCui == "C2747923"
      | con.getCui == "C0157540" | con.getCui == "C1535887" | con.getCui == "C0151947"
      | con.getCui == "C0034074" | con.getCui == "C0520546" | con.getCui == "C2721578"
      | con.getCui == "C0151946" | con.getCui == "C4524050" | con.getCui == "C0392108"
      | con.getCui == "C1868769") {
      positive_case = true
      if (d.getPolarity == -1){
        polarity_case = true
      }
      if (d.getUncertainty == 1) {
        uncertainty_case = true
      }
      println(">>>>>>>  index: " + pos + "  <<<<<<<  high_index: " +p)
      println(">>>>>>> CUI: "+ con.getCui)
      println(">>>>>>>>>>>>>   POLARIY: " + d.getPolarity)
      println(">>>>>>>>>>>>>   Uncertainty: " + d.getUncertainty)
      println(">>>>>>>>>>>>>   Confidence: " + d.getConfidence +"\n\n")
    }
    pos += 1
  }
```



```
sConcept])

mlsConcept](0)))
lem.getCui))
```

References:

1. https://commons.wikimedia.org/wiki/File:SaddlePE.PNG
2. https://www.mayoclinic.org/diseases-conditions/pulmonary-embolism/symptoms-causes/syc-20354647
3. https://www.flickr.com/photos/easy-pics/9609168594
4. https://pixabay.com/en/anatomy-blood-vessel-red-156854/

Thank you!

UNIVERSITY OF SAN FRANCISCO
CHANGE THE WORLD FROM HERE