Pharmacovigilance
Big Data for Drug Monitoring
Acknowledgments

Pei J Chen

Disclosures/Affiliations

- NLP Lab & Boston Children’s Hospital (Teaching Affiliate of Harvard Medical School)
- Co-Founder Wired Informatics
- VP, Apache cTAKES, ASF Member

NIH Funding/Projects

- PPRN-1306-04814 - PCORI - Phelan-McDermid Syndrome Data Network (PMS_DN)
- U01HG006828 - NHGRI - Pediatric eMERGE as part of EMR Phenotypes and Community Engaged Genomic Associations (eMERGE)
- U54LM008748 - NLM, NIH - Informatics for Integrating Biology and the Bedside (i2b2)
- U01 90TR0002/01- SHARP/ONC- SHARP Area 4: Secondary use of the EMR (SHARP)

Jay Vyas

Disclosures/Affiliations

- Red Hat Emerging Technologies
- Apache cTAKES, BigTop PMC/committer

Boston University (Ata Turk/Orran Krieger)

Mass Open Cloud
Knowledge Extraction

Big Data

Deep Knowledge Extraction
"We all get addicted to something that takes the pain away." OxyContin and Percocet is my addiction\]574251814964584448|Sat Mar 07 16:54:36 UTC 2015|(ignore me, just testing tweet for a ctakes app) Abilify, Nexium, Humira, Crestor, Advair\]574500003860869120|Sun Mar 08 09:20:49 UTC 2015|[Fell asleep at around 22:00 and woke up at 04:30 then realized I was supposed to take my Lantus and… https://t.co/R8FzBiZgpZ]

574286693215432704|Sat Mar 07 19:13:12 UTC 2015|[I'm a lot better than I was. I take suboxone which stop cravings and keep me from hurting n shit']
574398572994555904|Sun Mar 08 02:37:46 UTC 2015|[I've seen oxycontin take three lives. I've seen cocaine bring out the demons inside']
574398572994555904|Sun Mar 08 02:37:46 UTC 2015|[I've seen oxycontin take three lives. I've seen cocaine bring out the demons inside']
574457003424219136|Sun Mar 08 06:29:57 UTC 2015|[Ugh, Tamiflu is to avoid complications post (pneumonia) not fight the flu #catalyst]
Apache cTAKES

Pre
- Sectionizer
- Sentence Detector
- Tokenizer
- POS Tagger

Parser
- Shallow/Chunking
- Dependency
- Constituency

Normalization
- Lexical Variants
- Standardized Codes (UMLS CUI’s)

Assertion
- Polarity/Negation
- Subject (Patient/Family Member)
- Generic (“Diabetes Clinic”)
- History Of
- Conditional

Relations
- Temporality
- Co-Reference
- Severity/Degree Of
- Location Of
- Treats/Manages
Java 1.7 or higher

Dependency on UMLS which requires a UMLS license (free)

Apache Unstructured Information Management Architecture (UIMA) engineering framework

Existing Standards/Technologies:

UIMA, UIMA-AS, OpenNLP, clearTK, uimaFIT
### Apache cTAKES

**Boundary detection**
- Fx of obesity but no fx of coronary artery diseases.

**Tokenization**
- Fx of obesity but no fx of coronary artery diseases.

**Normalization**
- Disease or disorder
  - UMLS ID: C0028754
  - Status: family history
  - Negated: no

**Part-of-speech tagging**
- NN IN NN CC DT NN IN JJ NN NNS

**Shallow parsing**
- NP PP NP NN NP

**Entity recognition**
- Obesity
  - Disease or disorder
  - UMLS ID: C0028754
  - Status: family history
  - Negated: no

- Coronary artery disease
  - Disease or disorder
  - UMLS ID: C0010054
  - Status: family history
  - Negated: yes

- Coronary artery
  - Anatomy
  - UMLS ID: C0205042
**PHYSICAL EXAMINATION**


**IMPRESSION/REPORT/PLAN**

#1 Colorectal cancer of the cecum, biopsy proven. No evidence for metastatic disease.
#2 Thyroid insufficiency, on treatment.
#3 Psoriatic arthritis, adequately treated with methotrexate and topical steroid creams.

**PLANS/RECOMMENDATIONS:**

1. A surgical consultation for possible right hemicolec tomy in the next 1-2 weeks.
2. Complete pre-anesthetic medical evaluation, and obtain electrocardiogram.
3. Obtain the outside CT scan and have it formally reviewed by Mayo Clinic radiologist.
4. Obtain the outside colorectal biopsies and have these formally reviewed by Mayo Clinic pathologist.
Apace cTAKES + Big Data
Apache cTAKES + Big
Top 50 Grossing Prescription Drugs:
- Abilify
- Nexium
- Humira
- Crestor
- Advair
- Diskus
- Enbrel
- ...

Twitter Developer API

Apache cTAKES

Visualize (D3)
<table>
<thead>
<tr>
<th>id</th>
<th>datetm</th>
<th>cui</th>
<th>type</th>
<th>polarity</th>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td>578091377935802368</td>
<td>Wed Mar 18 07:11:39 UTC 2015</td>
<td>C1522449</td>
<td>5</td>
<td>1</td>
<td>RT</td>
</tr>
<tr>
<td>578091377935802368</td>
<td>Wed Mar 18 07:11:39 UTC 2015</td>
<td>C1445957</td>
<td>5</td>
<td>1</td>
<td>cholesterol</td>
</tr>
<tr>
<td>578091377935802368</td>
<td>Wed Mar 18 07:11:39 UTC 2015</td>
<td>C0011860</td>
<td>2</td>
<td>1</td>
<td>Type 2 diabetes</td>
</tr>
<tr>
<td>578091377935802368</td>
<td>Wed Mar 18 07:11:39 UTC 2015</td>
<td>C0011849</td>
<td>2</td>
<td>1</td>
<td>diabetes</td>
</tr>
<tr>
<td>578091377935802368</td>
<td>Wed Mar 18 07:11:39 UTC 2015</td>
<td>C0022658</td>
<td>2</td>
<td>1</td>
<td>kidney disease</td>
</tr>
<tr>
<td>578091377935802368</td>
<td>Wed Mar 18 07:11:39 UTC 2015</td>
<td>C0022646</td>
<td>6</td>
<td>1</td>
<td>kidney</td>
</tr>
<tr>
<td>578091377935802368</td>
<td>Wed Mar 18 07:11:39 UTC 2015</td>
<td>C1278978</td>
<td>6</td>
<td>1</td>
<td>kidney</td>
</tr>
<tr>
<td>578091377935802368</td>
<td>Wed Mar 18 07:11:39 UTC 2015</td>
<td>C0012634</td>
<td>2</td>
<td>1</td>
<td>disease</td>
</tr>
<tr>
<td>578412261276065792</td>
<td>Thu Mar 19 04:26:44 UTC 2015</td>
<td>C1170625</td>
<td>1</td>
<td>1</td>
<td>suboxone</td>
</tr>
<tr>
<td>578410121174249472</td>
<td>Thu Mar 19 04:18:13 UTC 2015</td>
<td>C0013144</td>
<td>3</td>
<td>1</td>
<td>sleepy</td>
</tr>
<tr>
<td>578410121174249472</td>
<td>Thu Mar 19 04:18:13 UTC 2015</td>
<td>C0234450</td>
<td>3</td>
<td>1</td>
<td>sleepy</td>
</tr>
<tr>
<td>578418311299407872</td>
<td>Thu Mar 19 04:50:46 UTC 2015</td>
<td>C0939400</td>
<td>1</td>
<td>1</td>
<td>nexium</td>
</tr>
<tr>
<td>578415072248143872</td>
<td>Thu Mar 19 04:37:54 UTC 2015</td>
<td>C0524222</td>
<td>5</td>
<td>1</td>
<td>OxyContin</td>
</tr>
<tr>
<td>578415072248143872</td>
<td>Thu Mar 19 04:37:54 UTC 2015</td>
<td>C0722364</td>
<td>1</td>
<td>1</td>
<td>OxyContin</td>
</tr>
<tr>
<td>578419430134165504</td>
<td>Thu Mar 19 04:55:13 UTC 2015</td>
<td>C0939400</td>
<td>1</td>
<td>1</td>
<td>nexium</td>
</tr>
<tr>
<td>578420317418315776</td>
<td>Thu Mar 19 04:58:44 UTC 2015</td>
<td>C0011209</td>
<td>5</td>
<td>1</td>
<td>delivery</td>
</tr>
<tr>
<td>578420317418315776</td>
<td>Thu Mar 19 04:58:44 UTC 2015</td>
<td>C0040610</td>
<td>1</td>
<td>1</td>
<td>tramadol</td>
</tr>
<tr>
<td>578420317418315776</td>
<td>Thu Mar 19 04:58:44 UTC 2015</td>
<td>C0719198</td>
<td>1</td>
<td>1</td>
<td>celebrex</td>
</tr>
<tr>
<td>578420317418315776</td>
<td>Thu Mar 19 04:58:44 UTC 2015</td>
<td>C0677049</td>
<td>3</td>
<td>1</td>
<td>Sucks</td>
</tr>
<tr>
<td>578420317418315776</td>
<td>Thu Mar 19 04:58:44 UTC 2015</td>
<td>C0030193</td>
<td>3</td>
<td>1</td>
<td>pain</td>
</tr>
<tr>
<td>578420317418315776</td>
<td>Thu Mar 19 04:58:44 UTC 2015</td>
<td>C0231441</td>
<td>3</td>
<td>1</td>
<td>immobile</td>
</tr>
</tbody>
</table>
Theres no cluster in this talk!
But we still use spark.

No need for separate ETL phases.
Scala serialization patterns OOTB.
RDD life cycle.
Will scale if/when you need it to.
Easy ways to create a spark cluster... just use ASF BigTop.

- clone bigtop
- gradle spark-yum
- vim vagrantconfig.yaml # add spark
- vagrant up
Spark/BigTop

https://github.com/jayunit100/SparkStreamingApps.git

Create project from existing sources

Import project from external model

Eclipse
Flash Builder
Gradle
Maven

SBT

Project name: SparkStreamingApps-APACHECON
Project location: /Users/jayunit100/Development/SparkStreamingApps-APACHECON
Project SDK: 1.7 (java version "1.7.0_71")
SBT version: 0.13.5
Scala version: 2.11.6

- Use auto-import
- Create directories for empty content roots automatically
- Download sources and docs
- Download SBT sources and docs

Module name: SparkStreamingApps-APACHECON
Content root: /Users/jayunit100/Development/SparkStreamingApps-APACHECON
Module file location: /Users/jayunit100/Development/SparkStreamingApps-APACHECON
Project format: .idea (directory based)
Spark/BigTop

Result
As tweets come in, `store()` puts them into memory. Every n seconds, those blocks are processed as a batch.
Spark Twitter Receiver (part of spark core)

onStart()
onStop()
are the API hooks
to start ingesting external data
Testing spark streaming apps...

Mock DStreams

A complete DStream implementation (for mocking twitter data).
Ingesting the Tweets

runDISK() sets the saveAsTextFiles.

```scala
def streamingFunction(ssc: StreamingContext): ReceiverInputDStream[Status] = {
  TwitchUtils.createStream(ssc, 
    filter { 
      case "Abilify, Nizatidine, Creator, Advair, Diskus, Embrel," + 
        "Bemicore, Cymbalta, Copaxone, Neulasta, Lantus, Solasta,"
        + 
        "Biltrax, Spiriva, Handihaler, Januvia, Atipila, Lantus, OxyContin," + 
        "Celebrex, Celebrex, Diclofen, Gleevec, Merceptin, Luscentis, Namenda," + 
        "Truvada, Zerubel, Anceza, Humalog, Novolog, Humulin, Januvia, Humana," + 
        "Benicar, Namenda, Suboxone, Symbricort, Lutricol, OxyContin, Xarelto"), 
    StorageLevel(MEMORY_AND_DISK))
  }
}

def runDISK(master: String, intervalSecs: Int, partitionsEachInterval: Int, numTweetsToCollect: Int, file: File) = {
  println("Initializing Streaming Spark Context...")

  val conf = new SparkConf()
  .setAppName(this.getClass.getName + " + System.currentTimeMillis")
  .setMaster(master)
  val sCon = new SparkContext(conf)
  val ssc = new StreamingContext(sCon, Seconds(intervalSecs))
  val tweetStream: ReceiverInputDStream[Status] = streamingFunction(ssc)

  // lists of empty files if 10 second interval, obviously.
  tweetStream.saveAsTextFiles(file.getAbsolutePath)
  ssc.start()
  ssc.awaitTermination()
  ssc.stop()
  System.exit(0)
}
```

Internally: this creates a new DStream... which will do the work of writing RDDs to disk.
Plugging in alternative processing schemes Cassandra

Rather than creating another child RDD, we can define our own forEachRDD callback directly.

```scala
/**
 * Example of cassandra implementation.
 * Not yet supported by the app but easy to add
 * by simply updating the parameters for setting up the cassandra connector etc.
 */

def runCassandra(master: String, intervalSecs: Int, partitionsEveryInterval: Int, numTweetsToCol: Int) {
  println("Initializing Streaming Spark Context...")

  val conf = new SparkConf()
    .setAppName(this.getClass.getSimpleName + "" + System.currentTimeMillis())
    .setMaster(master)
  val sCon = new SparkContext(conf)
  val ssc = new StreamingContext(sCon, Seconds(10))
  val tweetStream: ReceiverInputDStream[Status] = streamingFunction(ssc);

  tweetStream.foreachRDD(transactions => {
    CassandraConnector(conf).withSessionDo {
      session => {
        val x=1
        transactions.foreach{
          xN =>
            System.out.println("Running Cassandra Insert..." + xN)
            System.out.println("Note that this can fail if cassandra isn't working...")
            val nxNxt=xN.toString(" + xN.getTweet; session.executeAsync(s"INSERT INTO streaming_test.key_value (key, value) VALUES
```
Processing in place w/ ASF cTAKES.

Adding cTAKES hooks and calling in the same forEach...

def analyze(text: String): Any = {
  val aed: AnalysisEngineDescription = getDefaultPipeline()
  val jcas: JCas = JCasFactory.createJCas()
  jcas.setDocumentText(text)
  SimplePipeline.runPipeline(jcas, aed)
  val iter = JCasUtil.select(jcas, classOf[IdentifiedAnnotation]).iterator()
  while (iter.hasNext)
  {
    val entity = iter.next()
    // for demonstration purposes, we print all this stuff.
    val mentions = entity.getOntologyConceptArray
    var i = 0
    if (mentions != null & mentions.size > 0) {
      val uniqueCuis = scala.collection.mutable.Set[String]()
      for (i <- 1 to mentions.size - 1) {
        if (mentions.get(i) != null & mentions.get(i).isInstanceOf[UmlsConcept]){
          val concept = mentions.get(i).asInstanceOf[UmlsConcept] ;
          uniqueCuis += concept.getCui
        }
      }
      uniqueCuis.foreach(println)
    }
    System.out.println("==" + entity.getCoveredText + " " + entity.getPolarity + "==")
    System.out.println(entity);
  }
  // return the iterator.
  JCasUtil.select(jcas, classOf[BaseToken]).iterator()}
  jcas.reset();
  }

libraryDependencies += "org.apache.ctakes" % "ctakes-core" % "3.2.1"
libraryDependencies += "org.apache.ctakes" % "ctakes-core-res" % "3.2.1"
libraryDependencies += "org.apache.ctakes" % "ctakes-constituency-parser" % "3.2.1"
libraryDependencies += "org.apache.ctakes" % "ctakes-clinical-pipeline" % "3.2.1"
libraryDependencies += "org.apache.ctakes" % "ctakes-dictionary-lookup-fast" % "3.2.1"
libraryDependencies += "org.apache.ctakes" % "ctakes-drug-ner" % "3.2.1"
libraryDependencies += "org.apache.ctakes" % "ctakes-assertion" % "3.2.1"
Spark/BigTop

Running it locally with `sbt run`

```
sbt run.
default args will read credentials from /tmp/twitter.
```

```
sbt run
[info] Loading project definition from /Users/jayunit100/Development/SparkStreamingApps-APACHECON-2/project
[info] Set current project to SparkSBT (in build file:/Users/jayunit100/Development/SparkStreamingApps-APACHECON-2/project)
[warn] import sparkapps.ctakes.TwitterInputDStreamCTakes ^
[warn] It would fail on the following input: List(_) ^
[warn] list match { ^
[warn] two warnings found

Multiple main classes detected, select one to run:

[1] sparkapps.SparkApp1
[2] sparkapps.ctakes.CTakesTermAnalyzer

Enter number: 3
```
Spark/BigTop

1000s of tweets per day.

Every 60 seconds a directory with tweets (or without), is created.
Running it in a cluster

User spark submit, as you normally would.

Submit a non-local master.

BTW, when running locally make sure to use local [2]... guess why!
Results were very exciting, even after removing noise. Most tweets are jokes about cialis/narcotics. Remove that. Even still, 1000s of tweets per day. Lots of interesting (reasonable) NLP challenges.

Some samples, collected in just 15 minutes...

**Substitutions**: “Anyway, then she saw I was really hurting using Aspercreme (nighttime) all the Tylenol, so she gave me 2 10 mg oxycontin a day until I ...”

**Abuse**: “celebrex recreational use http://t.co/wy3M2PQwdR”

**Venting**: “Serious ?: Canadians in BC on #Humira, how long did it take 2 get approval from fair pharmacare & your health insurance? #stillwaiting”

**Scientific**: "Could A Mitochondrial Enhancer [Acetyl L-carnitine] Replace Cymbalta in #Fibromyalgia? http://t.co/tVNYBNaqJj"
Discussion