



Pei J Chen
Jay Vyas

Apache Software Foundation

An aerial, top-down view of a large, circular building with a complex, repeating geometric pattern of octagonal and rectangular sections. In the center is a circular skylight featuring a four-pointed star with the letters 'E', 'X', 'A', and 'S' positioned around it. A semi-transparent blue horizontal band is overlaid across the middle of the image, containing the main title and subtitle in orange text.

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/commmitter

Boston University (Ata Turk/Orran Krieger)

Mass Open Cloud

Knowledge Extraction



Big Data

The image features a Venn diagram with two overlapping circles. The left circle is labeled 'Big Data' and the right circle is labeled 'Deep Knowledge Extraction'. The intersection of the two circles is shaded in a darker blue. The background of the entire slide is a grayscale image of a large, ornate dome ceiling with a central star-shaped skylight.

Deep
Knowledge
Extraction

Twitter Data

574265225718951936|Sat Mar 07 17:47:53 UTC 2015|'"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'|
578499006638227456|Thu Mar 19 10:11:25 UTC 2015|'@MSpals I have an active lesion. He doesn't believe **Copaxone** working- he wants **avonex**, **tecfigera** or **Rebif**. Side effects scare me'|
578496649154138112|Thu Mar 19 10:02:03 UTC 2015|'**Naltrexone** pellets for Medical treatment of **heroin** or **oxycontin** dependence <http://t.co/sZdY41RQ0Q>'|
578458385248444416|Thu Mar 19 07:30:00 UTC 2015|'I have been talking to many patients recently, many have developed **type 2 diabetes** after starting **crestor**. Sadly no one warn...'|
578149924568588289|Wed Mar 18 11:04:18 UTC 2015|'I was on **remicade** for two years then it stopped working. Next was **simponi** aria. Didn't work at all worst **flare up**... <http://t.co/Er0pirEFNE>'|
578018923896930304|Wed Mar 18 02:23:45 UTC 2015|'RT @stigmaactionnet: "**HIV** prevention finally has a game-changer, and it's called **Truvada**" <http://t.co/77RjzCdrI3>'|
577965996880838656|Tue Mar 17 22:53:26 UTC 2015|'It's been 4 weeks since my doctor agreed to change my **insulin** from **levemir** to **lantus** and I'm still waiting... D:'|
578410121174249472|Thu Mar 19 04:18:13 UTC 2015|'this **abilify** has me **sleepy** af but i need to finish this homework'|
578268620394008578|Wed Mar 18 18:55:57 UTC 2015|'**Crestor** and **diabetes** and **renal failure** risks. Docs: do no harm!! #statins <http://t.co/v6k1vrFkp1>'|
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'|
578311814431305728|Wed Mar 18 21:47:35 UTC 2015|'**Addicted To OxyContin (oxycodone) | Addiction Epidemic** <http://t.co/EgQjlxbl31>'|
574457003424219136|Sun Mar 08 06:29:57 UTC 2015|'Ugh, **Tamiflu** is to avoid complications post **flu (pneumonia)** not fight the **flu** #catalyst'|
574479430510362624|Sun Mar 08 07:59:04 UTC 2015|'Kim's **blood sugar** was up again, 500. Gave **Humalog** & her nightly dose of **Lantus**. Still **cranky** at the Dr's office. #diabetes #diabeticproblems'|

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

Apache cTAKES

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_		
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

ENT: Examined and normal.

Skin: Psoriasis over the kneecaps and elbows, and within his hair.

Lymph: Examined and normal.

Thyroid: Not enlarged.

Heart: Core S1, S2, no murmur.

Lungs: Examined and normal.

Abdomen: Soft and nontender. No obvious masses.

Extremities: No signs of joint damage due to his psoriatic arthritis. Ankle scar on left from surgery. Right knee arthroscopy scar.

Pulses: Normal.

Neuro: Reflexes are normal.

Rect: Normal prostate, no masses palpable.

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 treatment with methotrexate and topical steroid creams

PLANS/RECOMMENDATIONS:

1. A surgical consultation for possible right hemicolectomy 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.

Event Discovery

UMLS Classification

■ Sign / Symptom

■ Test / Procedure

■ Disease / Diagnosis

■ Medication

■ Anatomy / General

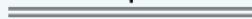
Negation Detection



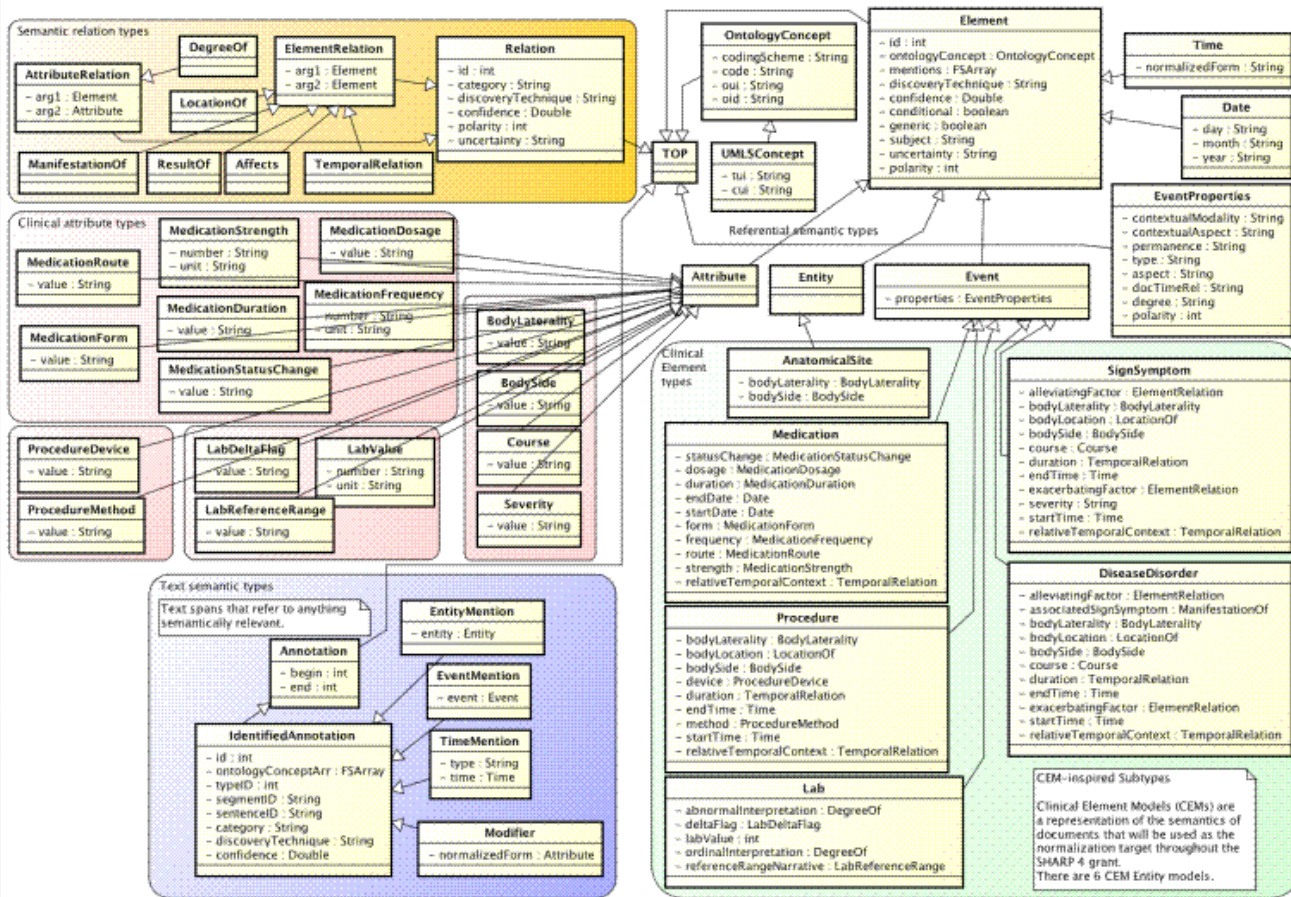
Uncertainty Detection



Time Expression Discovery



Type System





Apace cTAKES + Big Data

Apache cTAKES + Big



Spark



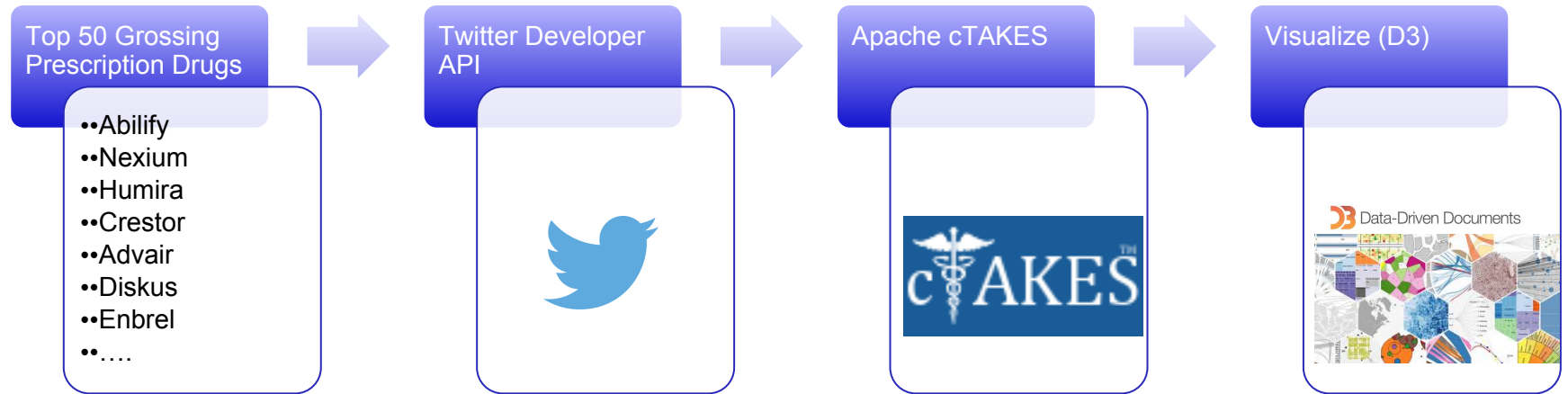
APACHE
HBASE



Solr
cassandra



Apache cTAKES + Big



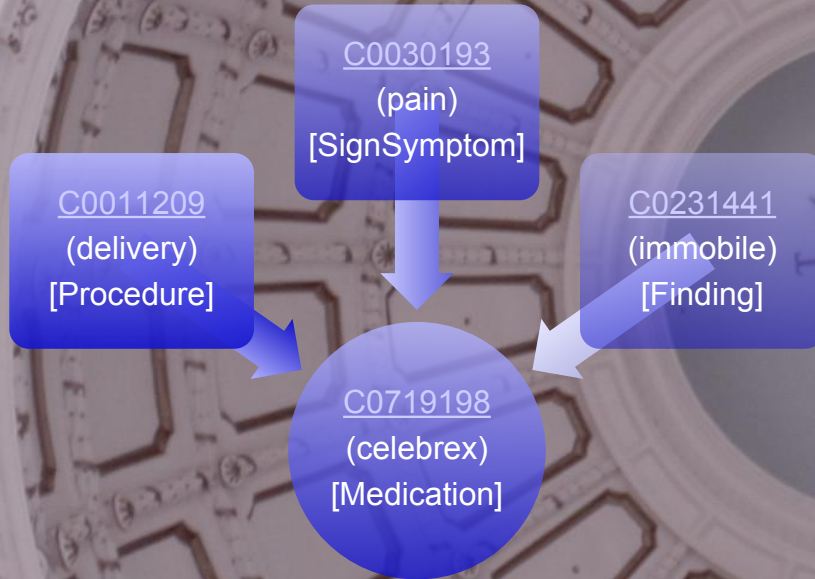
Apache cTAKES output

```
id|datetm|cui|type|polarity|text
578091377935802368|Wed Mar 18 07:11:39 UTC 2015|C1522449|5|1|RT
578091377935802368|Wed Mar 18 07:11:39 UTC 2015|C1445957|5|1|cholesterol
578091377935802368|Wed Mar 18 07:11:39 UTC 2015|C0011860|2|1|Type 2 diabetes
578091377935802368|Wed Mar 18 07:11:39 UTC 2015|C0011849|2|1|diabetes
578091377935802368|Wed Mar 18 07:11:39 UTC 2015|C0022658|2|1|kidney disease
578091377935802368|Wed Mar 18 07:11:39 UTC 2015|C0022646|6|1|kidney
578091377935802368|Wed Mar 18 07:11:39 UTC 2015|C1278978|6|1|kidney
578091377935802368|Wed Mar 18 07:11:39 UTC 2015|C0012634|2|1|disease
578412261276065792|Thu Mar 19 04:26:44 UTC 2015|C1170625|1|1|suboxone
578410121174249472|Thu Mar 19 04:18:13 UTC 2015|C0013144|3|1|sleepy
578410121174249472|Thu Mar 19 04:18:13 UTC 2015|C0234450|3|1|sleepy
578418311299407872|Thu Mar 19 04:50:46 UTC 2015|C0939400|1|1|nexium
578415072248143872|Thu Mar 19 04:37:54 UTC 2015|C0524222|5|1|OxyContin
578415072248143872|Thu Mar 19 04:37:54 UTC 2015|C0722364|1|1|OxyContin
578419430134165504|Thu Mar 19 04:55:13 UTC 2015|C0939400|1|1|nexium
578420317418315776|Thu Mar 19 04:58:44 UTC 2015|C0011209|5|1|delivery
578420317418315776|Thu Mar 19 04:58:44 UTC 2015|C0040610|1|1|tramadol
578420317418315776|Thu Mar 19 04:58:44 UTC 2015|C0719198|1|1|celebrex
578420317418315776|Thu Mar 19 04:58:44 UTC 2015|C0677049|3|1|Sucks
578420317418315776|Thu Mar 19 04:58:44 UTC 2015|C0030193|3|1|pain
578420317418315776|Thu Mar 19 04:58:44 UTC 2015|C0231441|3|1|immobile
```

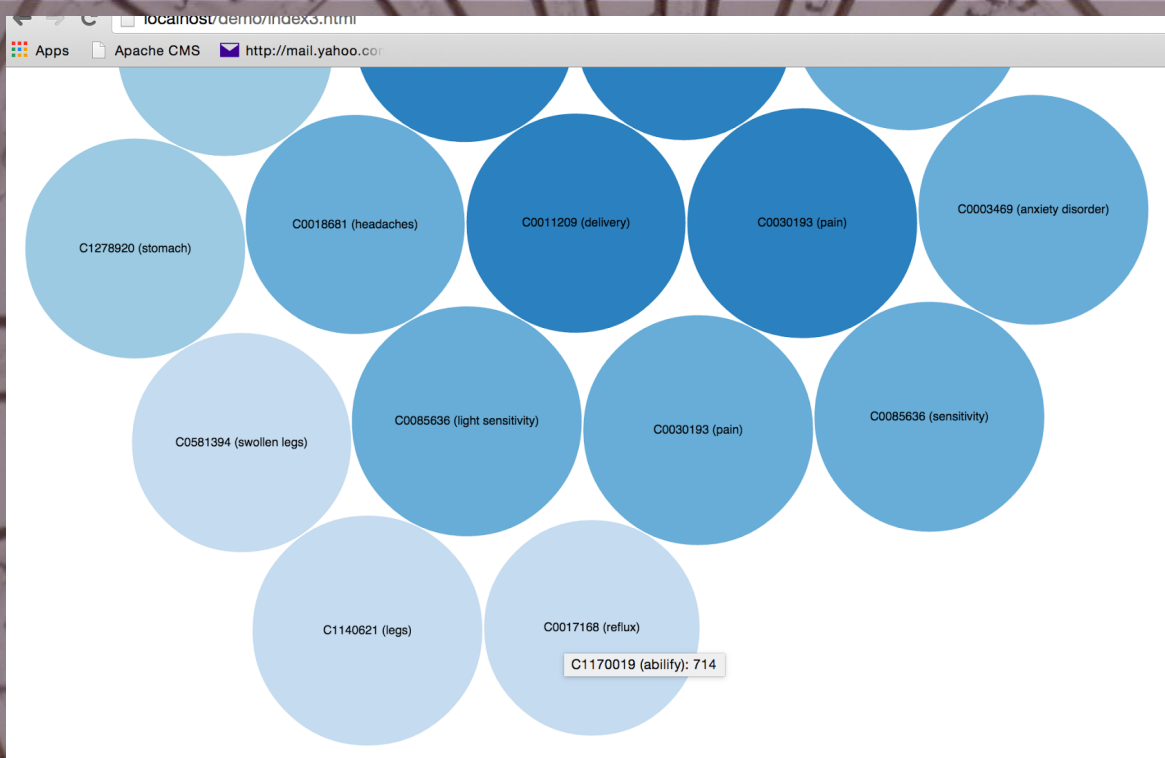


Visualization

Co-Occurrence



Demo



There's 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.

Spark/BigTop

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

New Project

Project name: SparkStreamingApps-APACHECON

Project location: /Users/jayunit100/Development/SparkStreamingApps-APACHECON

Project SDK: 1.7 (java version "1.7.0_71") New...

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

More Settings

Module name: SparkStreamingApps-APACHECON

Content root: /Users/jayunit100/Development/SparkStreamingApps-APACHECON

Module file location: /Users/jayunit100/Development/SparkStreamingApps-APACHECON

Project format: .idea (directory based)

? Cancel Previous Finish

Spark/BigTop

Result

The screenshot shows the IntelliJ IDEA IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, SBT Commands, and Help. The title bar indicates the current project is SparkStreamingApps-APACHECON-2. The left sidebar shows the Project Structure view with the following tree:

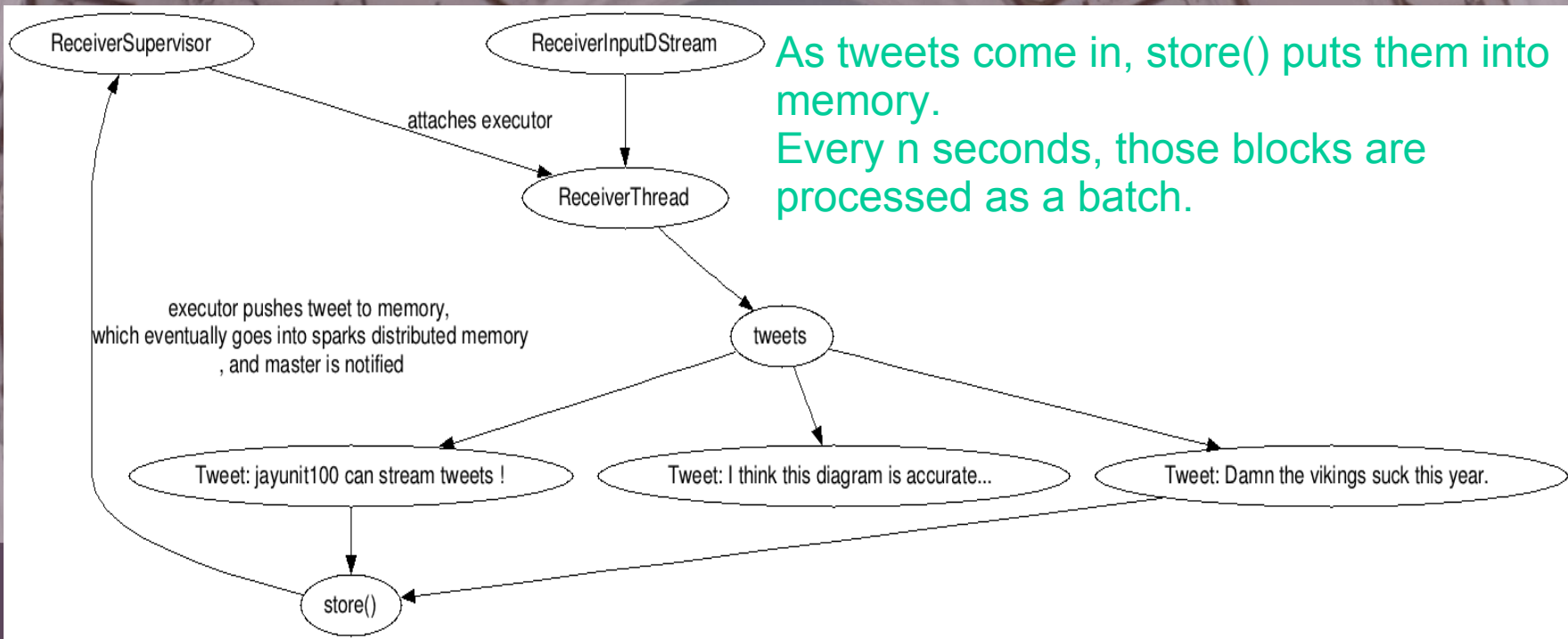
- Project
- SparkStreamingApps-APACHECON-2 [sparkstreamingapps-apachecon-2]
- .idea
- deploy
- project [sparkstreamingapps-apachecon-2-build] (sources root)
- src
 - main
 - java
 - scala
 - sparkapps
 - ctakes
 - CtakesTermAnalyzer
 - CTakesTwitterStreamingApp.scala
 - TwitterInputStreamCTakes.scala
 - tweetstream
 - ArgParser.scala
 - MockInputStream
 - TwitterAppTemplate
 - TwitterUtils
 - Utils
 - SparkApp1
 - test
 - target
 - .gitignore
 - build.sbt
 - README.md
 - External Libraries

The main editor area displays "No files are open" with a list of actions:

- Search Everywhere with Double ⇧
- Open a file by name with ⇧ ⌘ O
- Open Recent Files with ⇧ ⌘ E
- Open Navigation Bar with ⇧ T
- Drag and Drop file(s) here from Finder

At the bottom, the SBT Console shows the message: "SBT: [info] Resolving com.eed3si9n#sbt-assembly:0.12.0 ...". The status bar at the bottom right indicates "Git: master".

Spark Streaming Twitter ETL



Spark Twitter Receiver (part of spark core)

onStart()
onStop()
are the API hooks
to start ingesting
external data

```
/**
 * This will need to be injected somehow
 * so that we can have a unit test for confirming
 * that the processor works correctly.
 */
def thread():Thread = {
  return new Thread(
    new Runnable() {
      def run() = {
        try {
          System.out.println("Consumer k = " + System.getProperty("twitter4j.oauth.consumerKey"))
          val newTwitterStream = new TwitterStreamFactory().getInstance(twitterAuth)
          newTwitterStream.addListener(new StatusListener)

          val query = new FilterQuery
          if (filters.size > 0) {
            query.track(filters.toArray)
            newTwitterStream.filter(query)
          }
          else {
            newTwitterStream.sample()
          }
          setTwitterStream(newTwitterStream)

          logInfo("Twitter receiver started")
        }
        catch {
          case e: Exception =>
            restart("Error starting Twitter stream", e)
        }
      }
    }
  );
}

override def onStart()= {
  logInfo("Waiting 5 seconds to start to prevent abuse.")
  Thread.sleep(5000)
  stopped=false;
  val future = thread()
  future.start();
}

def onStop() {
  stopped=true;
  setTwitterStream(null)
}
```

Spark/BigTop

Testing spark streaming apps... Mock DStreams

A complete
DStream
implementation (for
mocking twitter
data).

```
case class MockInputDStream(sec:Long)(@transient ssc_ : StreamingContext)
  extends ReceiverInputDStream[Status](ssc_) {

  override def slideDuration(): Duration = {
    return Seconds(sec)
  }

  override def getReceiver(): Receiver[Status] = {
    new Receiver[Status](StorageLevel.DISK_ONLY) {

      def newStatus(i:Long):Status = {
        new Status() {override def getPlace: Place = ???
          override def isRetweet: Boolean = ???
          override def isFavorited: Boolean = ???
          override def getCreatedAt: Date = ???
          override def getUser: User = ???
          override def getContributors: Array[Long] = ???
          override def getRetweetedStatus: Status = ???
          override def getInReplyToScreenName: String = ???
          override def isTruncated: Boolean = ???
          override def getId: Long = i*System.currentTimeMillis();
          override def getCurrentUserRetweetId: Long = ???
          override def isPossiblySensitive: Boolean = ???
          override def getRetweetCount: Long = ???
          override def getGeoLocation: GeoLocation = ???
          override def getInReplyToUserId: Long = ???
          override def getSource: String = ???
          override def getText: String = "a tweet # " + i + " " + System.curr
          override def getInReplyToStatusId: Long = ???
          override def isRetweetedByMe: Boolean = ???
          override def compareTo(o: Status): Int = ???
          override def getHashtagEntities: Array[HashtagEntity] = ???
          override def getURLEntities: Array[URLEntity] = ???
          override def getMediaEntities: Array[MediaEntity] = ???
          override def getUserMentionEntities: Array[UserMentionEntity] = ???
          override def getAccessLevel: Int = ???
          override def getRateLimitStatus: RateLimitStatus = ???
        };
      }

      @volatile var killed=false;
      override def onStart() = {
        new Thread(
          new Runnable() {
            override def run(): Unit = {
              while(!killed) {
                System.out.println("-")
                //how does this = Status, but compiles to String?
                store(newStatus(System.currentTimeMillis()))
                Thread.sleep(1000);
              }
            }
          }
        ).start()
      }
      override def onStop() = {
        System.out.println("Stop requested.");
        killed=true;
      }
    }
  }
}
```

Ingesting the Tweets

runDISK() sets the
saveAsTextFiles.

```
def streamingFunction(sssc: StreamingContext): ReceiverInputDStream[Status] = {
  TwitterUtils.createStream(
    sssc,
    Uutils.getAuth,
    Seq(
      "Abilify,Nexium,Humira,Crestor,Advair_Diskus,Enbrel,"+
      "Remicade,Cymbalta,Copaxone,Neulasta,Lantus_Solostar,"+
      "Rituxan,Spiriva_Handihaler,Januvia,Atripla,Lantus,Oxycontin,"+
      "Celebrex,Celebrex_Diovan,Gleevec,Herceptin,Lucentis,Namenda,"+
      "Truvada,Enbrel,Ranexa,Humalog,Novolog,Tamiflu,Januvia,Namenda,"+
      "Benicar,Nasonex,Suboxone,Symbicort,Bystolic,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.getSimpleName + "" + System.currentTimeMillis())
    .setMaster(master)
  val sCon = new SparkContext(conf)
  val ssc = new StreamingContext(sCon, Seconds(10));
  val tweetStream: ReceiverInputDStream[Status] = streamingFunction(ssc);

  //lots 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.

```
/**
 * Save each RDD in this DStream as at text file, using string representation
 * of elements. The file name at each batch interval is generated based on
 */
L, @scala.specialized -T2, @scala.specialized +R] extends Object
def saveAsTextFiles(prefix: String, suffix: String = "") {
  val saveFunc = (rdd: RDD[T], time: Time) => {
    val file = rddToFile(prefix, suffix, time)
    rdd.saveAsTextFile(file)
  }
  this.foreachRDD(saveFunc)
}
```


Plugging in alternative processing schemes Cassandra

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

```
/**
 * 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, partitionsEachInterval:Int, numTweetsToCol
println("Initialzing 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
      Thread.sleep(1)
      transactions.foreach({
        xN =>
          System.out.println("Running Cassandra Insert..." + xN)
          System.out.println("Note that this can fail if cassandra isnt working...")
          val xNtxt=xN.toString+" "+xN.getText;
          session.executeAsync(s"INSERT INTO streaming_test.key_value (key, value) VALUES
      })
    }
  }
})
}
```

Spark/BigTop

Processing in place w/ ASF cTAKES.

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

```
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"
```

```
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.getOntologyConceptArr;  
    var i = 0;  
    if (mentions!=null && mentions.size > 0) {  
      val uniqueCuis = scala.collection.mutable.Set[String]()  
      for (i <- 0 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.print("----"+entity.getCoveredText + " " + entity.getPolarity+"----");  
    System.out.print(entity);  
  }  
  //return the iterator.  
  JCasUtil.select(jcas,classOf[BaseToken]).iterator()  
  jcas.reset();  
}
```

Spark/BigTop

Running it locally with 'sbt run'

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

```
jayunit100smacbookpro:SparkStreamingApps-APACHECON-2 jayunit100$ 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)
[info] Compiling 2 Scala sources to /Users/jayunit100/Development/SparkStreamingApps-APACHECON-2/target/scala-2.10
[warn] /Users/jayunit100/Development/SparkStreamingApps-APACHECON-2/src/main/scala/sparkapps/ctakes/CTakesTwitterS
[warn] import sparkapps.ctakes.TwitterInputDStreamCTakes
[warn]                                     ^
[warn] /Users/jayunit100/Development/SparkStreamingApps-APACHECON-2/src/main/scala/sparkapps/tweetstream/ArgParser
[warn] It would fail on the following input: List(_)
[warn]   list match {
[warn]     ^
[warn] two warnings found

Multiple main classes detected, select one to run:

[1] sparkapps.SparkApp1
[2] sparkapps.ctakes.CtakesTermAnalyzer
[3] sparkapps.ctakes.Driver

Enter number: 3
```


Spark/BigTop

1000s of tweets per day.

Every 60 seconds
a directory with
tweets
(or without),
is created.

```
11 Apr 2015 21:33:40 INFO SparkContext - Starting job: saveAsTextFile at DStream.sc:0 directories, 17 files
11 Apr 2015 21:33:40 INFO SparkContext - Job finished: saveAsTextFile at DStream.sc: jayunt1t00smacbookpro:~ jayunt
11 Apr 2015 21:33:40 INFO JobScheduler - Finished job streaming job 1428802420000 m: /tmp/tweets-1428802280000
11 Apr 2015 21:33:40 INFO JobScheduler - Total delay: 0.011 s for time 1428802420000 m: SUCCESS
11 Apr 2015 21:33:40 INFO BlockRDD - Removing RDD 27 from persistence list /tmp/tweets-1428802290000
11 Apr 2015 21:33:40 INFO BlockManager - Removing RDD 27 SUCCESS
11 Apr 2015 21:33:40 INFO TwitterInputDStream - Removing blocks of RDD BlockRDD[27] /tmp/tweets-1428802300000
11 Apr 2015 21:33:50 INFO ReceiverTracker - Stream 0 received 0 blocks SUCCESS
11 Apr 2015 21:33:50 INFO JobScheduler - Added jobs for time 1428802430000 ms /tmp/tweets-1428802320000
11 Apr 2015 21:33:50 INFO JobScheduler - Starting job streaming job 1428802430000 m: SUCCESS
11 Apr 2015 21:33:50 INFO SparkContext - Starting job: saveAsTextFile at DStream.sc: /tmp/tweets-1428802330000
11 Apr 2015 21:33:50 INFO SparkContext - Job finished: saveAsTextFile at DStream.sc: SUCCESS
11 Apr 2015 21:33:50 INFO JobScheduler - Finished job streaming job 1428802430000 m: /tmp/tweets-1428802340000
11 Apr 2015 21:33:50 INFO JobScheduler - Total delay: 0.009 s for time 1428802430000 m: SUCCESS
11 Apr 2015 21:33:50 INFO BlockRDD - Removing RDD 29 from persistence list /tmp/tweets-1428802350000
11 Apr 2015 21:33:50 INFO BlockManager - Removing RDD 29 SUCCESS
11 Apr 2015 21:33:50 INFO TwitterInputDStream - Removing blocks of RDD BlockRDD[29] /tmp/tweets-1428802370000
11 Apr 2015 21:34:00 INFO ReceiverTracker - Stream 0 received 0 blocks SUCCESS
11 Apr 2015 21:34:00 INFO JobScheduler - Added jobs for time 1428802440000 ms /tmp/tweets-1428802380000
11 Apr 2015 21:34:00 INFO JobScheduler - Starting job streaming job 1428802440000 m: SUCCESS
11 Apr 2015 21:34:00 INFO SparkContext - Starting job: saveAsTextFile at DStream.sc: /tmp/tweets-1428802390000
11 Apr 2015 21:34:00 INFO SparkContext - Job finished: saveAsTextFile at DStream.sc: SUCCESS
11 Apr 2015 21:34:00 INFO JobScheduler - Finished job streaming job 1428802440000 m: /tmp/tweets-1428802400000
11 Apr 2015 21:34:00 INFO JobScheduler - Total delay: 0.012 s for time 1428802440000 m: SUCCESS
11 Apr 2015 21:34:00 INFO BlockRDD - Removing RDD 31 from persistence list /tmp/tweets-1428802420000
11 Apr 2015 21:34:00 INFO BlockManager - Removing RDD 31 SUCCESS
11 Apr 2015 21:34:00 INFO TwitterInputDStream - Removing blocks of RDD BlockRDD[31] /tmp/tweets-1428802430000
11 Apr 2015 21:34:10 INFO ReceiverTracker - Stream 0 received 0 blocks SUCCESS
11 Apr 2015 21:34:10 INFO JobScheduler - Added jobs for time 1428802450000 ms /tmp/tweets-1428802440000
11 Apr 2015 21:34:10 INFO JobScheduler - Starting job streaming job 1428802450000 m: SUCCESS
11 Apr 2015 21:34:10 INFO SparkContext - Starting job: saveAsTextFile at DStream.sc: /tmp/tweets-1428802450000
11 Apr 2015 21:34:10 INFO SparkContext - Job finished: saveAsTextFile at DStream.sc:0 directories, 18 files
11 Apr 2015 21:34:10 INFO JobScheduler - Finished job streaming job 1428802450000 m: jayunt1t00smacbookpro:~ jayunt
11 Apr 2015 21:34:10 INFO JobScheduler - Total delay: 0.012 s for time 1428802450000 ms (execution: 0.010 s)
11 Apr 2015 21:34:10 INFO BlockRDD - Removing RDD 33 from persistence list
```

Spark/BigTop

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 !

Spark/BigTop

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

APACHE CON
NORTH AMERICA



The background of the slide is a photograph of the Arizona State Capitol building in Phoenix, Arizona, featuring a prominent dome and classical architectural elements. In the foreground, there is a large, ornate stone monument with several statues, likely the Pima County Courthouse in Tucson. The scene is set outdoors with trees and a clear sky.

Pei J Chen

@peistation

chenpei@apache.org

Jay Vyas

@jayunit100

jay@apache.org