APACHECON 2021

APACHE DEEP LEARNING 302

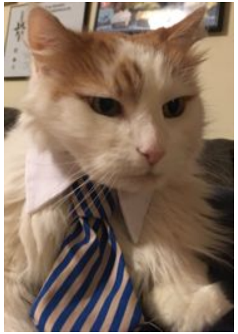
Timothy Spann

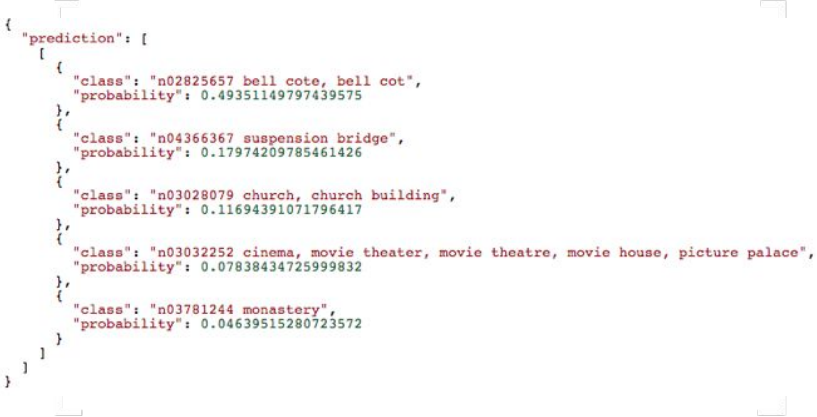
Developer Advocate / StreamNative

Tuesday 21-September-2021

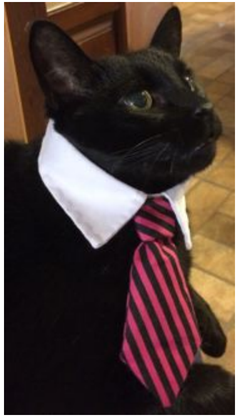
Apache Deep Learning 302

(For Programmers, Streaming, Messaging and Data Engineers)





```
{
  "prediction": [
    [
      {
        "class": "n02825657 bell cote, bell cot",
        "probability": 0.49351149797439575
      },
      {
        "class": "n04366367 suspension bridge",
        "probability": 0.17974209785461426
      },
      {
        "class": "n03028079 church, church building",
        "probability": 0.11694391071796417
      },
      {
        "class": "n03032252 cinema, movie theater, movie theatre, movie house, picture palace",
        "probability": 0.07838434725999832
      },
      {
        "class": "n03781244 monastery",
        "probability": 0.04639515280723572
      }
    ]
  ]
}
```



Timothy Spann

@PaaSDev

https://github.com/tspannhw/ApacheDeepLearning302

Agenda

Tuesday 18:00 UTC

Apache Deep Learning 302
Timothy Spann

This talk will discuss and show examples of using Apache Hadoop, Apache Kudu, Apache Flink, Apache Hive, Apache MXNet, Apache OpenNLP, Apache NiFi and Apache Spark for deep learning applications. This is the follow up to previous talks on Apache Deep Learning 101 and 201 and 301 at ApacheCon, Dataworks Summit, Strata and other events. As part of this talk, the presenter will walk through using Apache MXNet Pre-Built Models, integrating new open source Deep Learning libraries with Python and Java, as well as running real-time AI streams from edge devices to servers utilizing Apache NiFi and Apache NiFi - MiNiFi. This talk is geared towards Data Engineers interested in the basics of architecting Deep Learning pipelines with open source Apache tools in a Big Data environment. The presenter will also walk through source code examples available in github and run the code live on Apache NiFi and Apache Flink clusters.

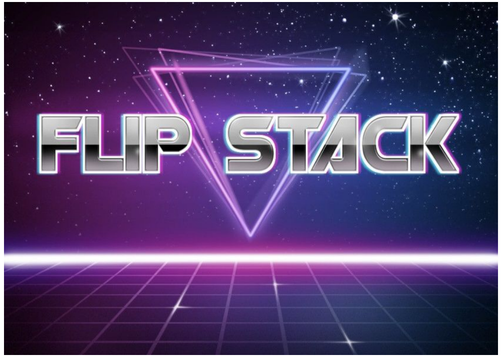
APACHECON
2021

Speaker Bio

Timothy Spann
Developer Advocate @ Stream Native

DZone Zone Leader and Big Data MVB
@PaasDev

https://github.com/tspannhw
https://www.datainmotion.dev/
https://github.com/tspannhw/SpeakerProfile
https://dev.to/tspannhw
https://sessionize.com/tspann/
https://www.slideshare.net/bunkertor



StreamNative Cloud

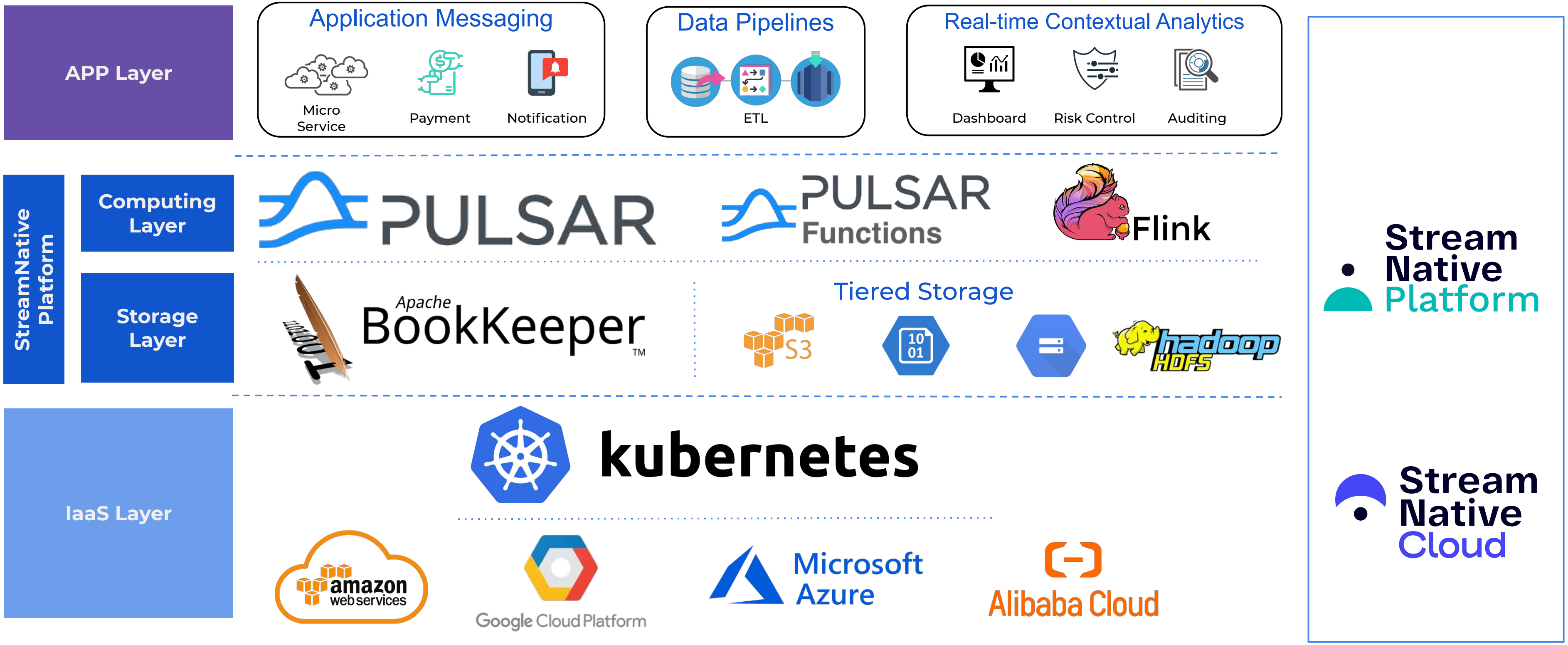
Powered by Apache Pulsar, StreamNative provides a cloud-native, real-time messaging and streaming platform to support multi-cloud and hybrid cloud strategies.

Cloud Native

Built for Containers

Flink SQL

StreamNative Solution



APP Layer

Application Messaging

Micro Service

Payment

Notification

Data Pipelines

ETL

Real-time Contextual Analytics

Dashboard

Risk Control

Auditing

StreamNative Platform

Computing Layer

Storage Layer

PULSAR

PULSAR Functions

Flink

Apache BookKeeper™

Tiered Storage

S3

hadoop HDFS

kubernetes

IaaS Layer

amazon web services

Google Cloud Platform

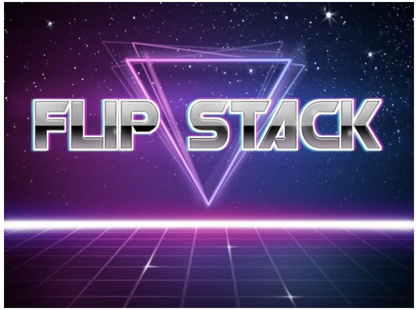
Microsoft Azure

Alibaba Cloud

Stream Native Platform

Stream Native Cloud

FLiP(N) Stack



- Apache **Fl**ink
- Apache **P**ulsar
- StreamNative's Flink Connector for Pulsar
- Apache **N**iFi
- Apache +++

Apache projects are the way for all streaming use cases.

Agenda - Data Engineering With Apache Deep Learning

- Introduction – This is my personal workflow
- Architecture Overview
- Apache Pulsar
- Apache MXNet
- Apache OpenNLP and Apache Tika
- Demos
- Questions

Deep Learning for Data Engineers

Multiple users, frameworks, languages, devices, data sources & clusters

DATA ENGINEER

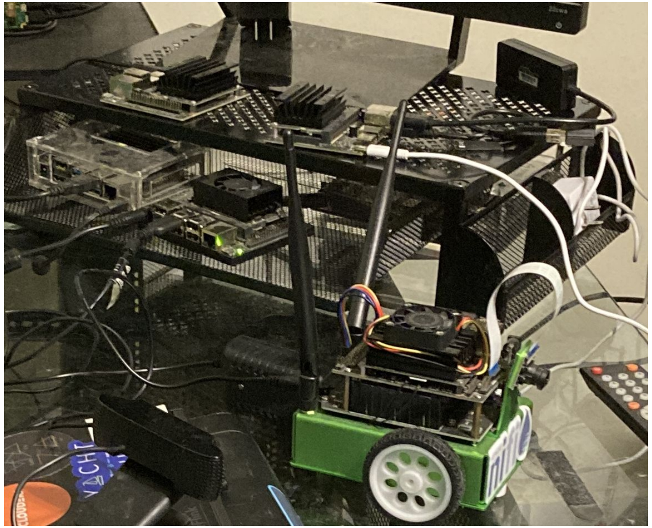
- Experience in ETL
- Coding skills in Scala, Python, Java
- Experience with Apache Hadoop
- Knowledge of database query languages such as SQL
- Knowledge of Lambda, Airflow, Debezium.

CAT

- Expert in ETL (Eating, Ties and Laziness)
- Social Media Maven
- Deep SME in Buzzwords
- No Coding Skills

AI

- Will Drive your Car
- Will Fix Your Code
- Will Beat You At Q-Bert
- Will Not Be Discussed Today
- Will Not Finish This Talk For Me, This Time



http://gluon.mxnet.io

Use Cases

So Why Am I Orchestrating These Complex Deep Learning Workflows?

Computer Vision

- Object Recognition
- Image Classification
- Object Detection
- Motion Estimation
- Annotation
- Visual Question and Answer
- Autonomous Driving

Speech Recognition

- Speech to Text
- Speech Recognition
- Chat Bot
- Voice UI

Natural Language Processing

- Sentiment Analysis
- Text Classification
- Named Entity Recognition

Recommender Systems

- Content-based Recommendations

https://gluon.mxnet.io/

Apache Deep Learning Components

Flink

- ➡ Continuous SQL
- ➡ Machine Learning
- ➡ Continuous ETL

PULSAR

- ➡ Distributed queue
- ➡ Buffering
- ➡ Messaging
- ➡ Functions

Streaming Analytics Manager

APACHE nifi

- ➡ Orchestration
- ➡ Queueing
- ➡ Simple Event Processing

Apache Deep Learning Components

mxnet

- Deep Learning Framework
- Run everywhere

OPEN NLP™

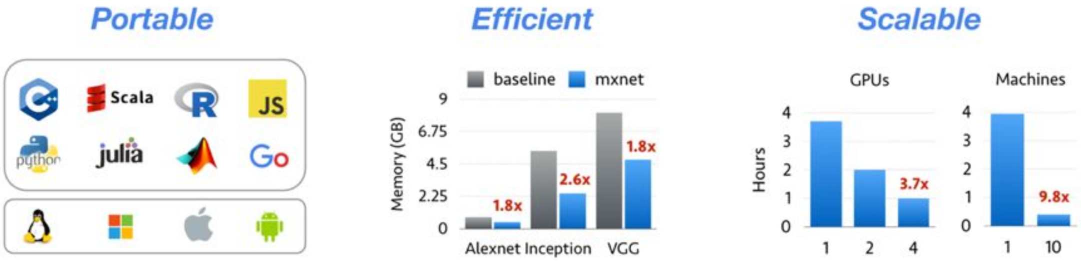
- Natural Language Processing
- Entity Resolution

Streaming Analytics
Manager

Apache™ Tika

- Detect metadata and data
- Extract metadata and data
- Content Analysis

mxnet

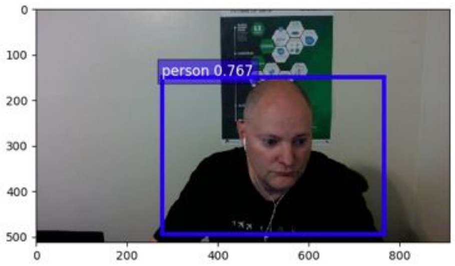


- Cloud ready
- Python, C++, Scala, R, Julia, Matlab, MXNet.js and Perl Support
- Experienced team (**XGBoost**)
- AWS, Microsoft, NVIDIA, Baidu, Intel
- Apache Incubator Project
- Run distributed on K8
- Runs on Raspberry PI, NVidia Jetson Nano and other constrained devices

https://mxnet.apache.org/versions/1.8.0/

https://cv.gluon.ai/api/model_zoo.html

mxnet



- Great documentation
- Crash Course
- **Gluon (Open API), GluonCV, GluonNLP**
- **Keras (One API Many Runtime Options)**
- Great Python Interaction
- Model Server Available
- **ONNX (Open Neural Network Exchange Format)** Support for AI Models
- Now in Version 1.8
- Rich Model Zoo!
- TensorBoard compatible

https://djl.ai/ http://mxnet.incubator.apache.org/ http://gluon.mxnet.io/ https://gluon-nlp.mxnet.io/

pip3 install -U keras-mxnet pip3 install --pre --upgrade mxnet pip3 install gluonnlp

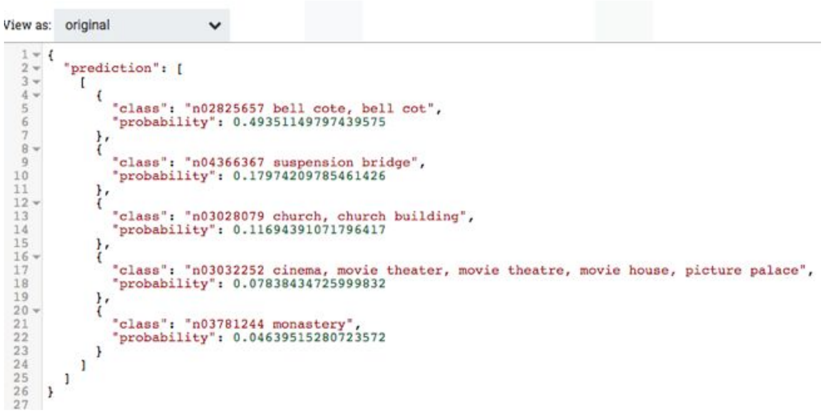
Apache MXNet Pre-Built Models - Model Zoo

- CaffeNet
- SqueezeNet v1.1
- Inception v3
- Single Shot Detection (SSD)
- VGG16
- VGG19
- ResidualNet 152
- LSTM

https://mxnet.apache.org/api/python/docs/api/gluon/model_zoo/index.html

Apache MXNet Model Server with Apache NiFi

multi-model-server --start --models squeezenet=https://s3.amazonaws.com/model-server/models/squeezenet_v1.1/squeezenet_v1.1.model --mms-config server.config --foreground



```
{
  "prediction": [
    [
      {
        "class": "n02825657 bell cote, bell cot",
        "probability": 0.49351149797439575
      },
      {
        "class": "n04366367 suspension bridge",
        "probability": 0.17974209785461426
      },
      {
        "class": "n03028079 church, church building",
        "probability": 0.11694391071796417
      },
      {
        "class": "n03032252 cinema, movie theater, movie theatre, movie house, picture palace",
        "probability": 0.07838434725999832
      },
      {
        "class": "n03781244 monastery",
        "probability": 0.04639515280723572
      }
    ]
  ]
}
```

https://community.cloudera.com/t5/Community-Articles/Apache-Deep-Learning-101-Processing-Apache-MXNet-Model/ta-p/247944
https://github.com/awslabs/multi-model-server/blob/master/docs/server.md
https://github.com/awslabs/multi-model-server/blob/master/docs/model_zoo.md

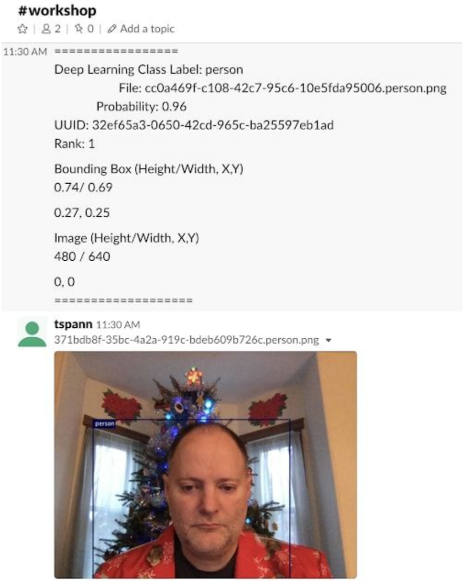
Using Model Server For Character-level CNN Model

```
multi-model-server --start --models
crepe=https://s3.amazonaws.com/model-server/model_archive_1.0/crepe.mar --mms-config
server.config
```

```
{
  "category": "Movies_and_TV"
}
```

```
curl -X POST http://127.0.0.1:9999/predictions/crepe -F "data=[{\"review_title\":\"Inception is the best\",\"review\": \"great direction and story\"}]"
```

Apache MXNet Native Processor through DJL.AI for Apache NiFi



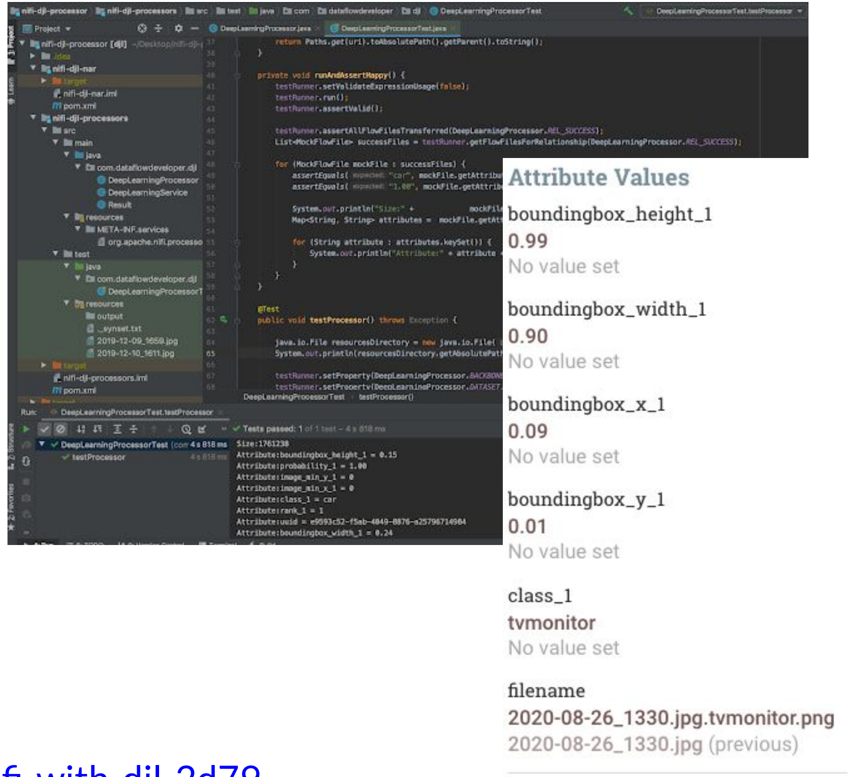
#workshop

☆ | 2 | 0 | Add a topic

```
11:30 AM ================
Deep Learning Class Label: person
            File: cc0a469f-c108-42c7-95c6-10e5fda95006.person.png
        Probability: 0.96
UUID: 32ef65a3-0650-42cd-965c-ba25597eb1ad
Rank: 1
Bounding Box (Height/Width, X,Y)
0.74/ 0.69
0.27, 0.25
Image (Height/Width, X,Y)
480 / 640
0, 0
==================
```

tspann 11:30 AM

371bdb8f-35bc-4a2a-919c-bdeb609b726c.person.png



```
        return Paths.get(uri).toAbsolutePath().getParent().toString();
    }

    private void runAndAssertHappy() {
        testRunner.setValidateExpressionUsage(false);
        testRunner.run();
        testRunner.assertValid();

        testRunner.assertAllFlowFilesTransferred(DeepLearningProcessor.REL_SUCCESS);
        List<MockFlowFile> successFiles = testRunner.getFlowFilesForRelationship(DeepLearningProcessor.REL_SUCCESS);

        for (MockFlowFile mockFile : successFiles) {
            assertEquals( expected: "car", mockFile.getAttribu
            assertEquals( expected: "1.00", mockFile.getAttrib

            System.out.println("Size:" +             mockFil
            Map<String, String> attributes =  mockFile.getAtt

            for (String attribute : attributes.keySet()) {
                System.out.println("Attribute:" + attribute +
            }
        }
    }

    @Test
    public void testProcessor() throws Exception {

        java.io.File resourcesDirectory = new java.io.File(
        System.out.println(resourcesDirectory.getAbsolutePat

        testRunner.setProperty(DeepLearningProcessor.BACKBON
        testRunner.setProperty(DeepLearningProcessor.DATASET.
```

✓ Tests passed: 1 of 1 test – 4 s 818 ms

DeepLearningProcessorTest (com 4 s 818 ms

✓ testProcessor 4 s 818 ms

```
Size:1761238
Attribute:boundingbox_height_1 = 0.15
Attribute:probability_1 = 1.00
Attribute:image_min_y_1 = 0
Attribute:image_min_x_1 = 0
Attribute:class_1 = car
Attribute:rank_1 = 1
Attribute:uuid = e9593c52-f5ab-4049-8876-a25796714984
Attribute:boundingbox_width_1 = 0.24
```

Attribute Values

boundingbox_height_1
0.99
No value set

boundingbox_width_1
0.90
No value set

boundingbox_x_1
0.09
No value set

boundingbox_y_1
0.01
No value set

class_1
tvmonitor
No value set

filename
2020-08-26_1330.jpg.tvmonitor.png
2020-08-26_1330.jpg (previous)

This processor uses the DJL.AI Java Interface

https://github.com/tspannhw/nifi-djl-processor

https://dev.to/tspannhw/easy-deep-learning-in-apache-nifi-with-djl-2d79

Apache Deep Learning 101,201,202

- https://www.slideshare.net/bunkertor/apache-deep-learning-101-apachecon-montreal-2018-v031
- https://www.slideshare.net/bunkertor/apache-deep-learning-202-washington-dc-dws-2019
- https://www.slideshare.net/bunkertor/apache-deep-learning-201-barcelona-dws-march-2019

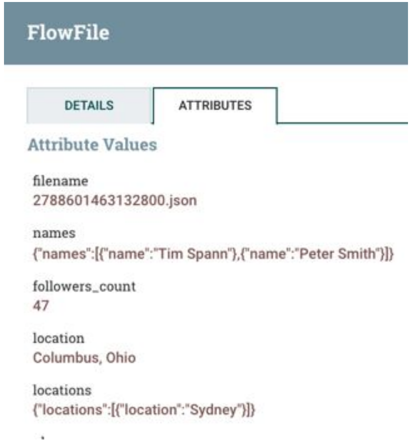
mxnet

Apache OpenNLP 1.9.3 with Apache NiFi 1.14.0

Apache OpenNLP for Entity Resolution Processor
https://github.com/tspannhw/nifi-nlp-processor

Requires installation of NAR and Apache OpenNLP Models (http://opennlp.sourceforge.net/models-1.5/).

This is a non-supported processor that I wrote and put into the community. You can write one too!



FlowFile

DETAILS | ATTRIBUTES

Attribute Values

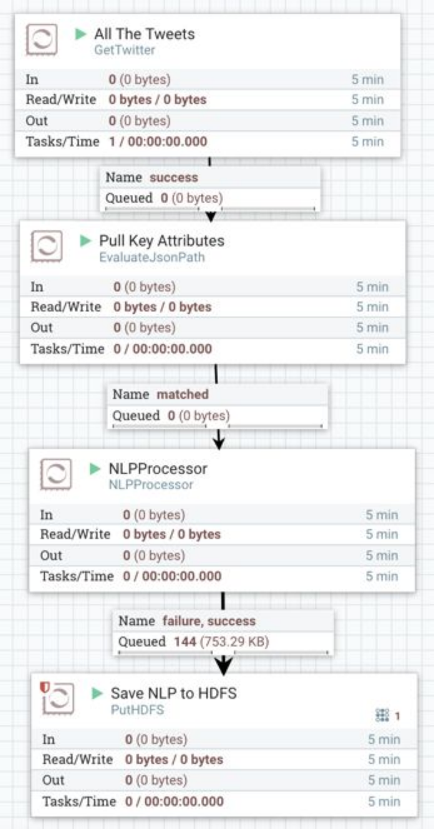
filename
2788601463132800.json

names
{"names":[{"name":"Tim Spann"},{"name":"Peter Smith"}]}

followers_count
47

location
Columbus, Ohio

locations
{"locations":[{"location":"Sydney"}]}



All The Tweets
GetTwitter

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	1 / 00:00:00.000	5 min

Name success
Queued 0 (0 bytes)

Pull Key Attributes
EvaluateJsonPath

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

Name matched
Queued 0 (0 bytes)

NLPProcessor
NLPProcessor

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

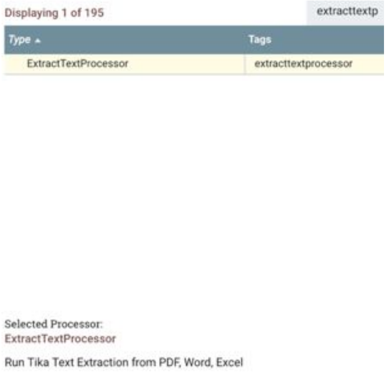
Name failure, success
Queued 144 (753.29 KB)

Save NLP to HDFS
PutHDFS
1

In	0 (0 bytes)	5 min
Read/Write	0 bytes / 0 bytes	5 min
Out	0 (0 bytes)	5 min
Tasks/Time	0 / 00:00:00.000	5 min

https://community.cloudera.com/t5/Community-Articles/Open-NLP-Example-Apache-NiFi-Processor/ta-p/249293

https://opennlp.apache.org/download.html



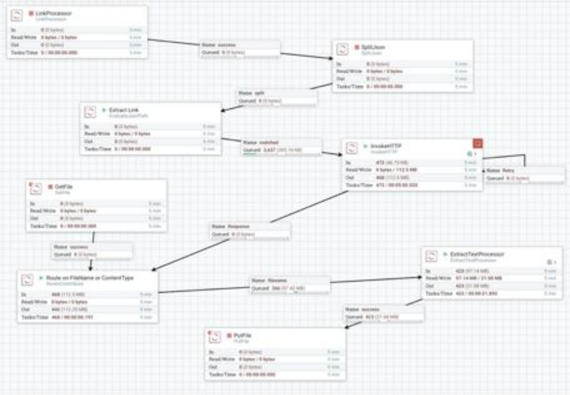
Displaying 1 of 195

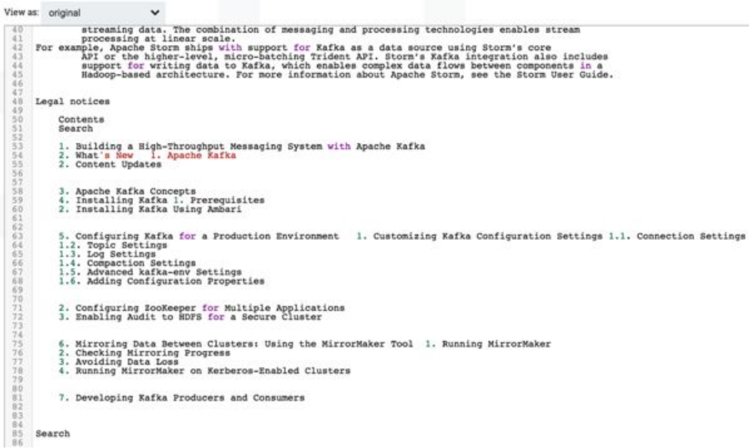
extracttextp

Type ▴	Tags
ExtractTextProcessor	extracttextprocessor

Selected Processor:
ExtractTextProcessor

Run Tika Text Extraction from PDF, Word, Excel





View as: original

```
          streaming data. The combination of messaging and processing technologies enables stream
          processing at linear scale.
For example, Apache Storm ships with support for Kafka as a data source using Storm's core
          API or the higher-level, micro-batching Trident API. Storm's Kafka integration also includes
          support for writing data to Kafka, which enables complex data flows between components in a
          Hadoop-based architecture. For more information about Apache Storm, see the Storm User Guide.


Legal notices

     Contents
     Search

     1. Building a High-Throughput Messaging System with Apache Kafka
     2. What's New    1. Apache Kafka
     2. Content Updates


     3. Apache Kafka Concepts
     4. Installing Kafka 1. Prerequisites
     2. Installing Kafka Using Ambari


     5. Configuring Kafka for a Production Environment    1. Customizing Kafka Configuration Settings 1.1. Connection Settings
     1.2. Topic Settings
     1.3. Log Settings
     1.4. Compaction Settings
     1.5. Advanced kafka-env Settings
     1.6. Adding Configuration Properties


     2. Configuring ZooKeeper for Multiple Applications
     3. Enabling Audit to HDFS for a Secure Cluster


     6. Mirroring Data Between Clusters: Using the MirrorMaker Tool  1. Running MirrorMaker
     2. Checking Mirroring Progress
     3. Avoiding Data Loss
     4. Running MirrorMaker on Kerberos-Enabled Clusters


     7. Developing Kafka Producers and Consumers



Search
```

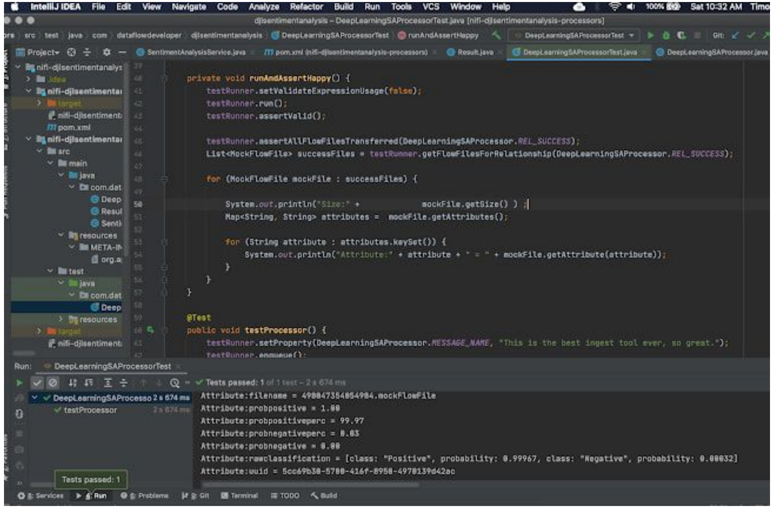
https://github.com/tspannhw/nifi-extracttext-processor

https://community.cloudera.com/t5/Community-Articles/Data-Processing-Pipeline-Parsing-PDFs-and-Identifying-Names/ta-p/249105

https://community.cloudera.com/t5/Community-Articles/ExtractText-NiFi-Custom-Processor-Powered-by-Apache-Tika/ta-p/249392

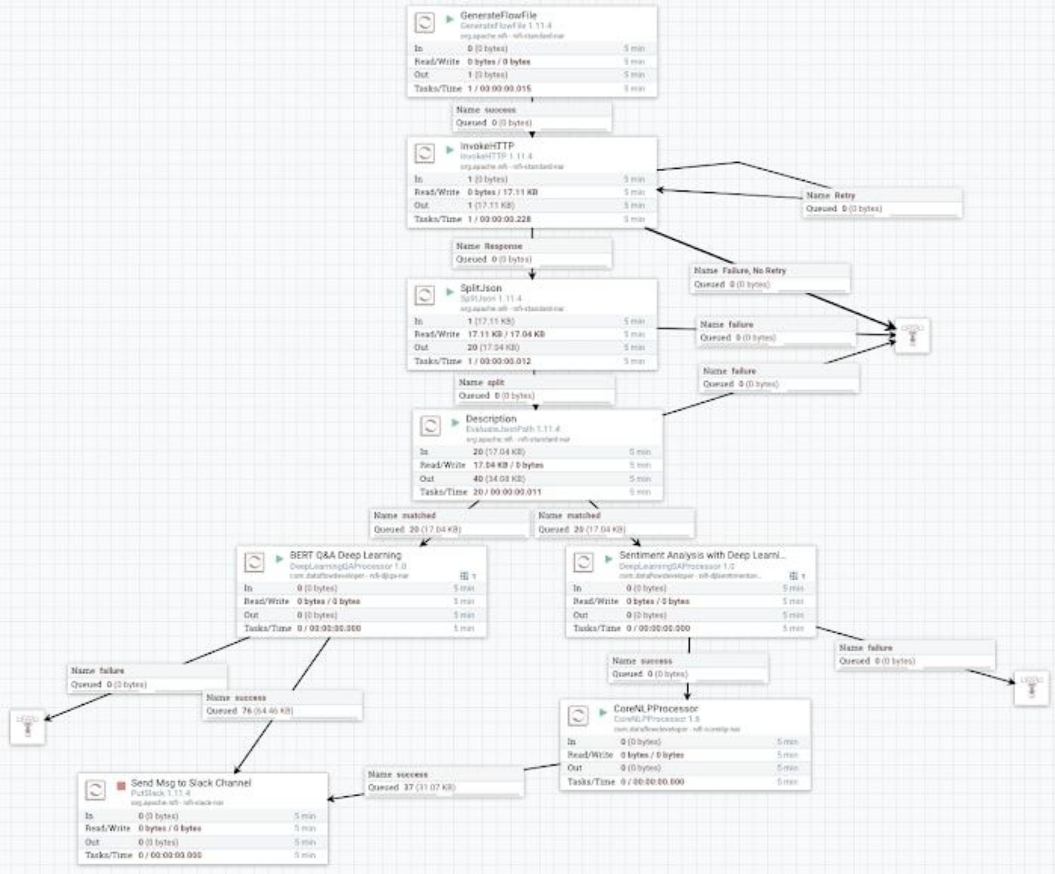
https://community.cloudera.com/t5/Community-Articles/Parsing-Any-Document-with-Apache-NiFi-1-5-with-Apache-Tika/ta-p/247672

https://community.cloudera.com/t5/Community-Articles/Creating-HTML-from-PDF-Excel-and-Word-Documents-using-Apache/ta-p/247968


IntelliJ IDEA
djlsentimentanalysis – DeepLearningSAProcessorTest.java [nifi-djlsentimentanalysis-processors]
DeepLearningSAProcessorTest
runAndAssertHappy
SentimentAnalysisService.java
pom.xml (nifi-djlsentimentanalysis-processors)
Result.java
DeepLearningSAProcessorTest.java
DeepLearningSAProcessor.java
Run: DeepLearningSAProcessorTest
Tests passed: 1 of 1 test – 2 s 674 ms
DeepLearningSAProcesso 2 s 674 ms
testProcessor
Tests passed: 1
Services
Run
Problems
Git
Terminal
TODO
Build

```java
    private void runAndAssertHappy() {
        testRunner.setValidateExpressionUsage(false);
        testRunner.run();
        testRunner.assertValid();

        testRunner.assertAllFlowFilesTransferred(DeepLearningSAProcessor.REL_SUCCESS);
        List<MockFlowFile> successFiles = testRunner.getFlowFilesForRelationship(DeepLearningSAProcessor.REL_SUCCESS);

        for (MockFlowFile mockFile : successFiles) {

            System.out.println("Size:" +              mockFile.getSize() ) ;
            Map<String, String> attributes =  mockFile.getAttributes();

            for (String attribute : attributes.keySet()) {
                System.out.println("Attribute:" + attribute + " = " + mockFile.getAttribute(attribute));
            }
        }
    }

    @Test
    public void testProcessor() {
        testRunner.setProperty(DeepLearningSAProcessor.MESSAGE_NAME, "This is the best ingest tool ever, so great.");
        testRunner.enqueue();
```

```
Attribute:filename = 498047354054984.mockFlowFile
Attribute:probpositive = 1.00
Attribute:probpositiveperc = 99.97
Attribute:probnegativeperc = 0.03
Attribute:probnegative = 0.00
Attribute:rawclassification = [class: "Positive", probability: 0.99967, class: "Negative", probability: 0.00032]
Attribute:uuid = 5cc69b38-5780-416f-8958-4970139d42ac
```



https://dev.to/tspannhw/using-djl-ai-for-deep-learning-based-sentiment-analysis-in-nifi-dataflow-3c3a

Deeper Content

- https://github.com/tspannhw/EverythingApacheNiFi
- https://github.com/streamnative/pulsar-flink
- https://www.linkedin.com/pulse/2021-schedule-tim-spann/
- https://github.com/tspannhw/SpeakerProfile/
- https://streamnative.io/en/blog/release/2021-04-20-flink-sql-on-streamnative-cloud
- https://docs.streamnative.io/cloud/stable/compute/flink-sql
- https://mxnet.apache.org/versions/1.8.0/get_started

@PaasDev **in timothyspann**

https://www.pulsardeveloper.com/