



Finance NLP

Certification Trainings

Jan 24, 2023

Bunyamin
Polat



Sai Sailesh
Gadde



David
Cecchini



Merve
Ertas Uslu



Enes
Yilmazturk



Jose Juan Martinez
Finance and Legal NLP Lead
juan@johnsnowlabs.com



Welcome - We have a lot of excited things ahead of us

Duration

Module

Notebooks

50 min

Introduction. Text Splitting. Tokenization. Embeddings. Binary, Multiclass, Multilabel Classification

01.Page_Splitting.ipynb
02.Sentence_Splitting_Tokenization.ipynb
03.Word_Sentence_EMBEDDINGS.ipynb
04.0.Document_Paragraph_Classification.ipynb
04.2.Training_Financial_Multiclass_Classifier.ipynb

10 min

break

60 min

Named Entity Recognition. Relation Extraction. Training. Zero-shot NER and RE.

05.0.NER_and_ZeroShotNER.ipynb
05.1.Training_Financial_NER.ipynb
06.0.Relation_Extraction.ipynb
06.2.ZeroShot_Relation_Extraction.ipynb

10 min

break

50 min

Understand Entities in Context with Assertion Status.
Question&Answering Models. Data Normalization, Mapping, Augmentation

07.0.Understand_Entities_in_Context.ipynb
08.0.Answering_Questions_Financial_Texts.ipynb
09.0.Normalization_with_Entity_Resolution_Edgar.ipynb
10.0.Data_Augmentation_with_ChunkMappers.ipynb

10 min

break

50 min

Financial Graph on a real use case. Deidentification. Integration with Visual NLP.

Databricks Solution Accelerator
11.0.Deidentification.ipynb
90.0.Financial_Visual_Classification.ipynb
90.1.Visual_and_Textual_Classification.ipynb



STATE OF THE ART

Introduction Reviewing Spark NLP

John Snow Labs in 2022



Globally awarded

As best AI Specialist
of the 2022 year

Global 100



Most popular

NLP library in
the enterprise

O'Reilly Media
PyPI downloads

#1 Accuracy

on 20 benchmarks in
peer-reviewed papers

Papers with Code

Microsoft

Google

amazon

intel

IBM

databricks

verizon[✓]

indeed[✓]

CapitalOne

VIACOM

MCKESSON

MERCK

Roche

selectdata

UiPath Robotic Work.

ASCO[®]
CANCER LINQ[™]
DISCOVERY[™]

cnrs

Imperial College
London

Georgia Tech

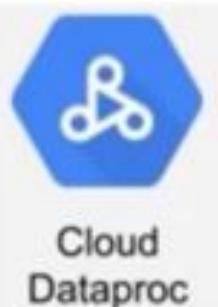
STANFORD
UNIVERSITY

Optimized, Tested, Supported Integrations



python™

kubernetes



databricks



Amazon
SageMaker



comet



amazon
EMR

mlflow

kaggle



Google Cloud

Microsoft
Azure

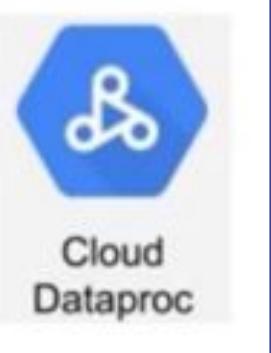


Synapse ML
Simple and Distributed
Machine Learning

Optimized, Tested, Supported Integrations



CLOUDERA



Spark NLP

Community & models hub:
<https://nlp.johnsnowlabs.com>

downloads 40M

downloads/month 2M

Entity Recognition	Information Extraction	Spelling & Grammar	Text Classification																					
I love Lucy PERSON	They met Last week DATE -> 29-04-2020	abc ✓ She become the first... -> She became the first																						
Translation	Summarization	Question Answering	Emotion Detection																					
 [je t'aime -> i love you]		 Q&A																						
Split Text <ul style="list-style-type: none"> Sentence Detector Tokenizer Normalizer nGram Generator Word Segmentation 		Clean Text <ul style="list-style-type: none"> Spell Checker Grammar Checker Writing Style Checker Stopword Cleaner Summarization 																						
Understand Grammar <ul style="list-style-type: none"> Stemmer Lemmatizer Part of Speech Tagger Dependency Parser Translation 		Find in Text <ul style="list-style-type: none"> Text Matcher Regex Matcher Date Matcher Chunker Question Answering 																						
Trainable & Tunable 		Scalable to a Cluster 																						
Transformers 		Spark MLLib 																						
Community 		10,000+ Pre-trained Pipelines, Models & Transformers <table border="1"> <tr><td>BERT</td><td>ELMO</td><td>GloVe</td></tr> <tr><td>ALBERT</td><td>DeBERTa</td><td>USE</td></tr> <tr><td>Longformer</td><td>ELECTRA</td><td></td></tr> <tr><td>T5</td><td>NMT</td><td>LaBSE</td></tr> <tr><td>DistilBERT</td><td>RoBERTa</td><td></td></tr> <tr><td></td><td>XLM-RoBERTa</td><td></td></tr> <tr><td>S-BERT</td><td>XLNet</td><td></td></tr> </table>		BERT	ELMO	GloVe	ALBERT	DeBERTa	USE	Longformer	ELECTRA		T5	NMT	LaBSE	DistilBERT	RoBERTa			XLM-RoBERTa		S-BERT	XLNet	
BERT	ELMO	GloVe																						
ALBERT	DeBERTa	USE																						
Longformer	ELECTRA																							
T5	NMT	LaBSE																						
DistilBERT	RoBERTa																							
	XLM-RoBERTa																							
S-BERT	XLNet																							
250+ Languages 																								

Spark NLP

Community & models hub:
<https://nlp.johnsnowlabs.com>

downloads 40M

downloads/month 2M

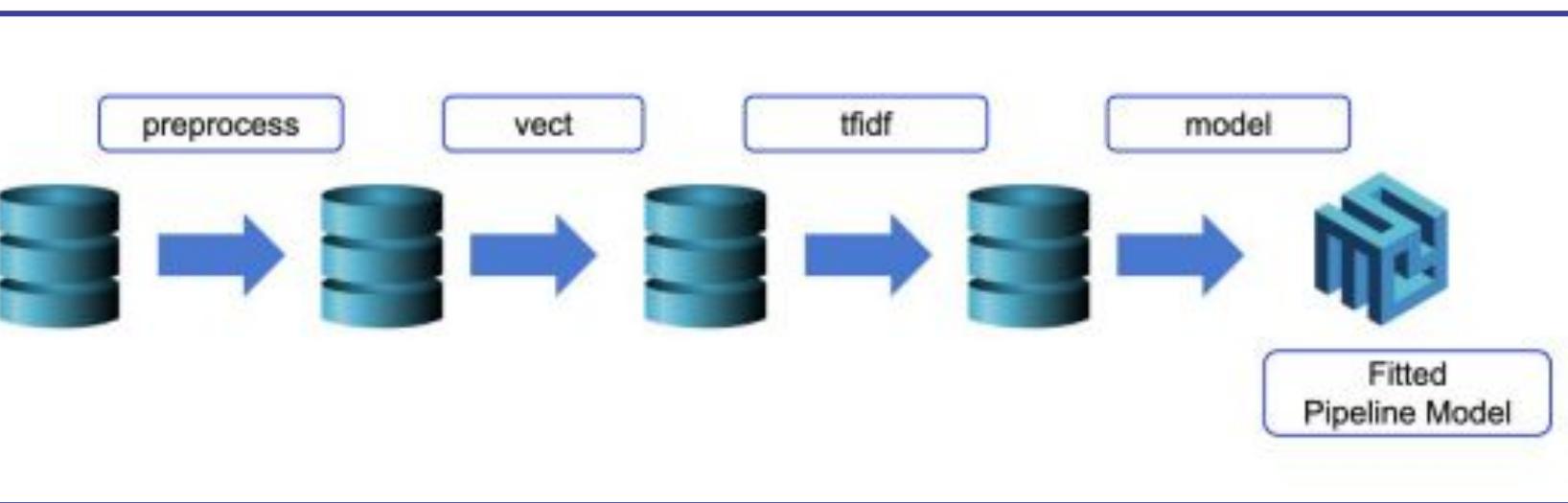
Entity Recognition	Information Extraction	Spelling & Grammar	Text Classification	Image Classification
I love Lucy PERSON	They met Last week time 29-04-2020	abc She become the first... -> She became the first		
Translation	Summarization	Question Answering	Emotion Detection	Automatic Speech Recognition
[Je t'aime -> I love you]		Q&A		

Split Text	Clean Text	12,000+	250+
<ul style="list-style-type: none"> Sentence Detector Tokenizer Normalizer nGram Generator Word Segmentation 	<ul style="list-style-type: none"> Spell Checker Grammar Checker Writing Style Checker Stopword Cleaner Summarization 	12,000+ Pre-trained Pipelines, Models & Transformers	250+ Languages
Understand Grammar	Find in Text	Pre-trained Pipelines, Models & Transformers	Languages
<ul style="list-style-type: none"> Stemmer Lemmatizer Part of Speech Tagger Dependency Parser Translation 	<ul style="list-style-type: none"> Text Matcher Regex Matcher Date Matcher Chunker Question Answering 	<div style="display: flex; justify-content: space-around;"> <div>BERT</div> <div>ELMO</div> <div>TAPAS</div> </div> <div style="display: flex; justify-content: space-around;"> <div>ALBERT</div> <div>DeBERTa</div> <div>USE</div> </div> <div style="display: flex; justify-content: space-around;"> <div>Longformer</div> <div>ELECTRA</div> </div> <div style="display: flex; justify-content: space-around;"> <div>T5</div> <div>NMT</div> <div>VIT</div> </div> <div style="display: flex; justify-content: space-around;"> <div>DistilBERT</div> <div>RoBERTa</div> </div> <div style="display: flex; justify-content: space-around;"> <div>XLM-RoBERTa</div> </div> <div style="display: flex; justify-content: space-around;"> <div>Wav2Vec2</div> <div>XLNet</div> </div>	

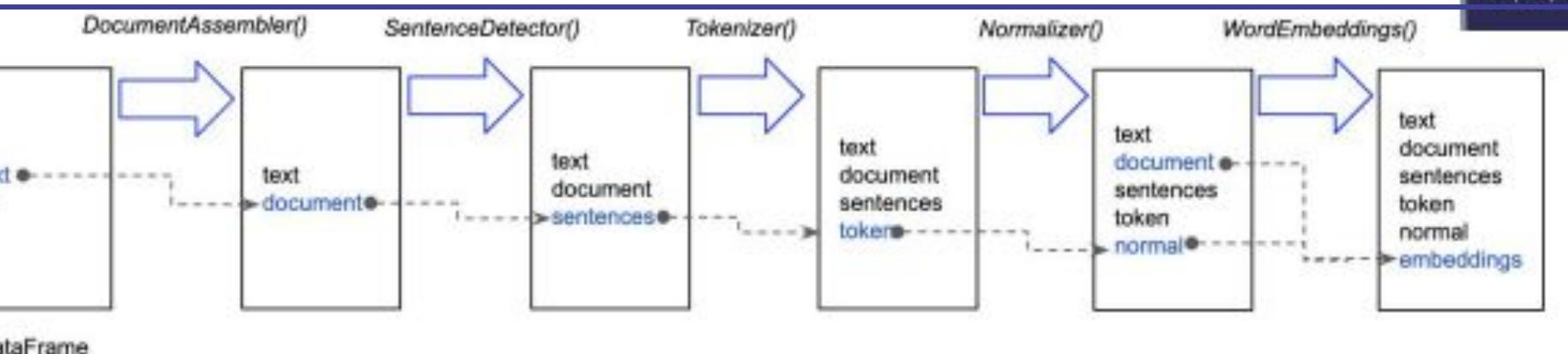
Trainable & Tunable	Scalable to a Cluster	Transformers	Hardware Optimized	Community

Introducing Spark NLP

Pipeline of annotators

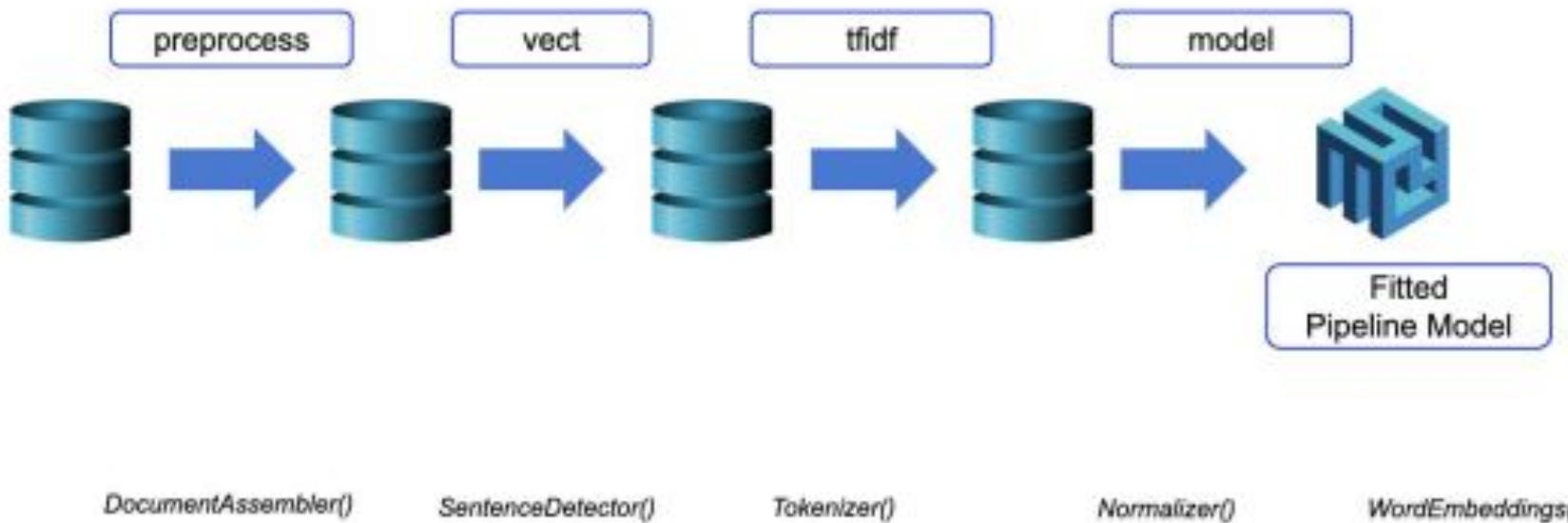


```
from pyspark.ml import Pipeline  
  
documentAssembler = DocumentAssembler()\n    .setInputCol("text")\n    .setOutputCol("document")  
  
sentenceDetector = SentenceDetector()\n    .setInputCols(["document"])\n    .setOutputCol("sentences")  
  
tokenizer = Tokenizer()\n    .setInputCols(["sentences"])\n    .setOutputCol("token")  
  
normalizer = Normalizer()\n    .setInputCols(["token"])\n    .setOutputCol("normal")  
  
word_embeddings=WordEmbeddingsModel.pretrained()\n    .setInputCols(["document","normal"])\n    .setOutputCol("embeddings")  
  
nlpPipeline = Pipeline(stages=[\n    documentAssembler,\n    sentenceDetector,\n    tokenizer,\n    normalizer,\n    word_embeddings,\n])  
  
nlpPipeline.fit(df).transform(df)
```

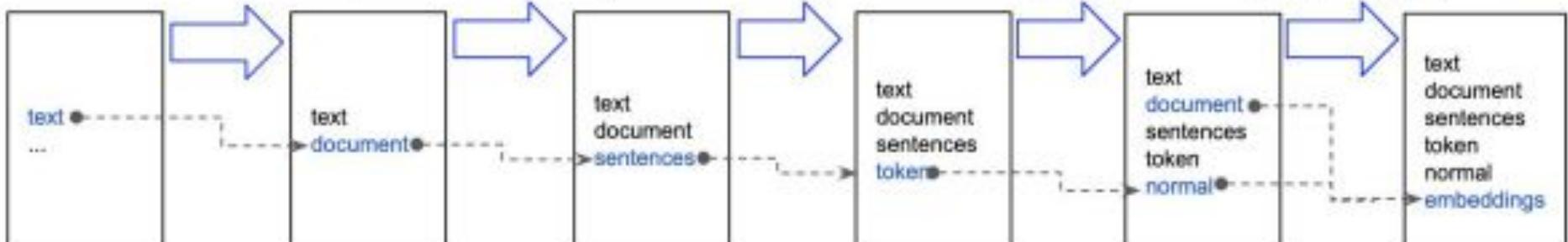


Introducing Spark NLP

Pipeline of annotators



DocumentAssembler() SentenceDetector() Tokenizer() Normalizer() WordEmbeddings()



DataFrame

```
documentAssembler = nlp.DocumentAssembler()\n    .setInputCol("text")\n    .setOutputCol("document")\n\n# Consider using SentenceDetector with rules/patterns to get smaller chunks of text\nsentence_detector = nlp.SentenceDetectorDLModel.pretrained("sentence_detector")\n    .setInputCols(["document"])\n    .setOutputCol("sentence")\n\ntokenizer = nlp.Tokenizer()\n    .setInputCols(["sentence"])\n    .setOutputCol("token")\n\nembeddings = nlp.BertEmbeddings.pretrained("bert_embeddings_legal_bert_base_uncased")\n    .setInputCols(["sentence", "token"])\n    .setOutputCol("embeddings")\n\nner_model = finance.NerModel.pretrained("finnern_secton_chunk", "en", "finance/models")\n    .setInputCols(["sentence", "token", "embeddings"])\n    .setOutputCol("ner")\n\nner_converter = finance.NerConverterInternal()\n    .setInputCols(["sentence", "token", "ner"])\n    .setOutputCol("ner_chunk")\n\nnlpPipeline = nlp.Pipeline(stages=[\n    documentAssembler,\n    sentence_detector,\n    tokenizer,\n    embeddings,\n    ner_model,\n    ner_converter])\n\nempty_data = spark.createDataFrame([[""]]).toDF("text")\nmodel = nlpPipeline.fit(empty_data)
```

Spark NLP can be run both at **cluster level**, leveraging all the nodes, and a **master-only level**, working only in the driver machine (*1 node*)

```
documentAssembler = nlp.DocumentAssembler()\
    .setInputCol("text")\
    .setOutputCol("document")

# Consider using SentenceDetector with rules/patterns to get smaller chunks from long sentences
sentence_detector = nlp.SentenceDetectorDLModel.pretrained("sentence_detector_dl", "xx")\
    .setInputCols(["document"])\
    .setOutputCol("sentence")

tokenizer = nlp.Tokenizer()\
    .setInputCols(["sentence"])\
    .setOutputCol("token")

embeddings = nlp.BertEmbeddings.pretrained("bert_embeddings_legal_bert_base_uncased", "en")\
    .setInputCols(["sentence", "token"])\
    .setOutputCol("embeddings")

ner_model = finance.NerModel.pretrained("finner_sec_conll", "en", "finance/models") \
    .setInputCols(["sentence", "token", "embeddings"])\
    .setOutputCol("ner")

ner_converter = finance.NerConverterInternal()\
    .setInputCols(["sentence", "token", "ner"])\
    .setOutputCol("ner_chunk")

nlpPipeline = nlp.Pipeline(stages=[

    documentAssembler,
    sentence_detector,
    tokenizer,
    embeddings,
    ner_model,
    ner_converter])

empty_data = spark.createDataFrame([[""]]).toDF("text")

model = nlpPipeline.fit(empty_data)
```

Cluster level: Uses Spark MLlib **Pipelines** and **fit/transform**.

```
text = '''December 2007 SUBORDINATED LOAN AGREEMENT. THE  
df = spark.createDataFrame([[text]]).toDF("text")  
result = model.transform(df)
```

Result:

Spark
Dataframe

scalable to millions of documents, slow for very few documents

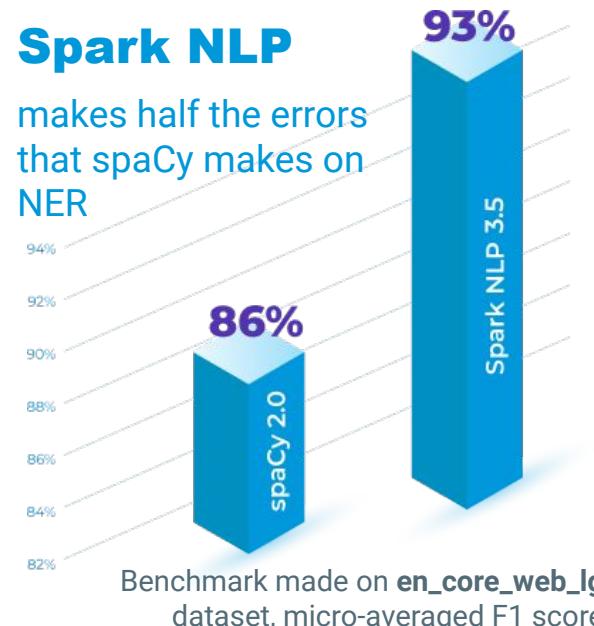


Spark NLP

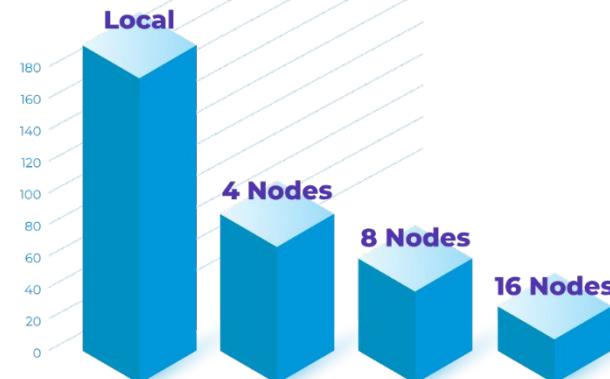
is natively scalable and production-ready

Spark NLP

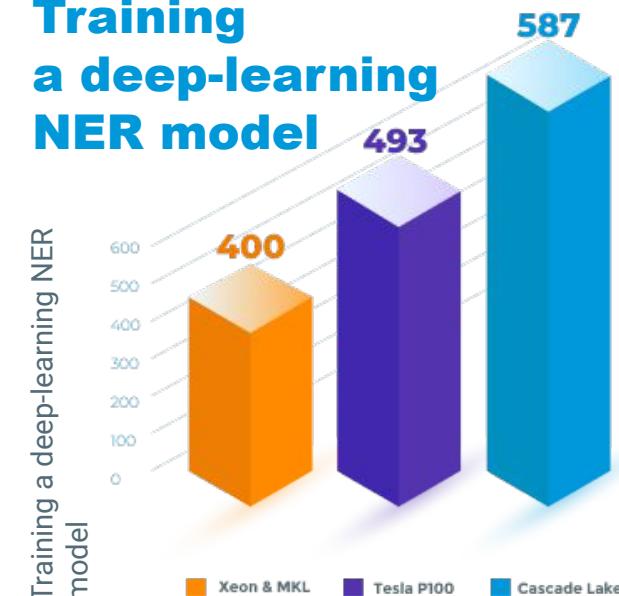
makes half the errors
that spaCy makes on
NER



Speedup on Cluster (less is better)



Training a deep-learning NER model

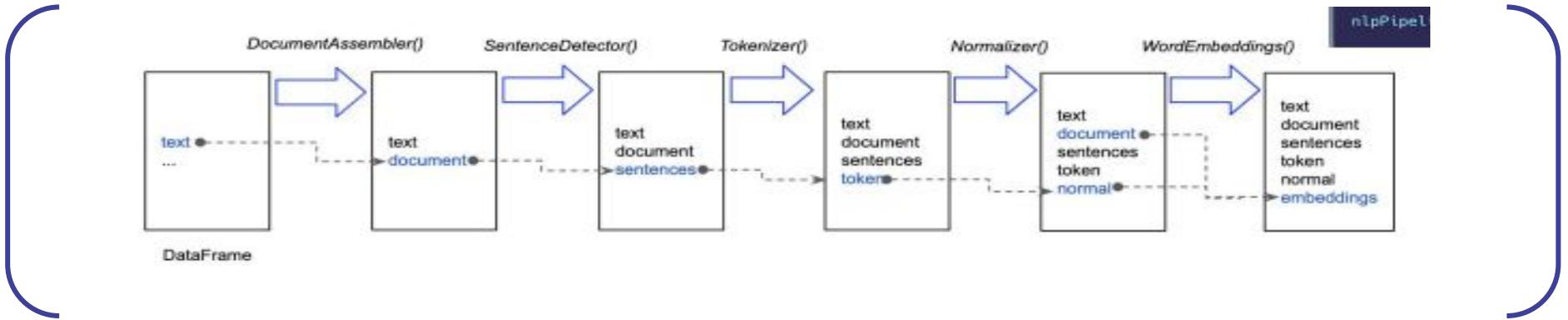


Accuracy

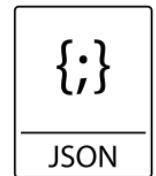
Scalability

Speed

nlp.LightPipeline



Thunder fast for few documents,
not parallelizable



Json-friendly

Spark NLP can be run both at **cluster level**, leveraging all the nodes, and a **master-only level**, working only in the driver machine (*1 node*)

```
documentAssembler = nlp.DocumentAssembler()\
    .setInputCol("text")\
    .setOutputCol("document")

# Consider using SentenceDetector with rules/patterns to get smaller chunks from long sentences
sentence_detector = nlp.SentenceDetectorDLModel.pretrained("sentence_detector_dl", "xx")\
    .setInputCols(["document"])\
    .setOutputCol("sentence")

tokenizer = nlp.Tokenizer()\
    .setInputCols(["sentence"])\
    .setOutputCol("token")

embeddings = nlp.BertEmbeddings.pretrained("bert_embeddings_legal_bert_base_uncased", "en")\
    .setInputCols(["sentence", "token"])\
    .setOutputCol("embeddings")

ner_model = finance.NerModel.pretrained("finner_sec_conll", "en", "finance/models")\
    .setInputCols(["sentence", "token", "embeddings"])\
    .setOutputCol("ner")

ner_converter = finance.NerConverterInternal()\
    .setInputCols(["sentence", "token", "ner"])\
    .setOutputCol("ner_chunk")

nlpPipeline = nlp.Pipeline(stages=[

    documentAssembler,
    sentence_detector,
    tokenizer,
    embeddings,
    ner_model,
    ner_converter])

empty_data = spark.createDataFrame([[""]]).toDF("text")

model = nlpPipeline.fit(empty_data)
```

Cluster level: Uses Spark MLlib **Pipelines** and **fit/transform**.

```
text = '''December 2007 SUBORDINATED LOAN AGREEMENT. THE  
df = spark.createDataFrame([[text]]).toDF("text")  
result = model.transform(df)
```

Result:

Spark
Dataframe

scalable to millions of documents, slow for very few documents

Driver-only: Used **LightPipelines** and **annotate/fullAnnotate**

```
light_model = nlp.LightPipeline(model)  
light_result = light_model.fullAnnotate(text)
```

Result:

json

thunder fast for few documents, not scalable

Introducing Spark NLP



Spark is like a locomotive racing a bicycle. The bike will win if the load is light, it is quicker to accelerate and more agile, but with a heavy load the locomotive might take a while to get up to speed, but it's going to be faster in the end.

Faster inference

```
from sparknlp.base import LightPipeline  
LightPipeline(someTrainedPipeline).annotate(someStringOrArray)
```

LightPipelines are Spark ML pipelines converted into a single machine but multithreaded task, becoming more than 10x times faster for smaller amounts of data (small is relative, but 50k sentences is roughly a good maximum).

New johnsnowlabs library

In 2022, we introduced the *johnsnowlabs* library, which allows you to get your environment ready with just a couple of lines.

How to run

Finance NLP is very easy to run on both clusters and driver-only environments using `johnsnowlabs` library:

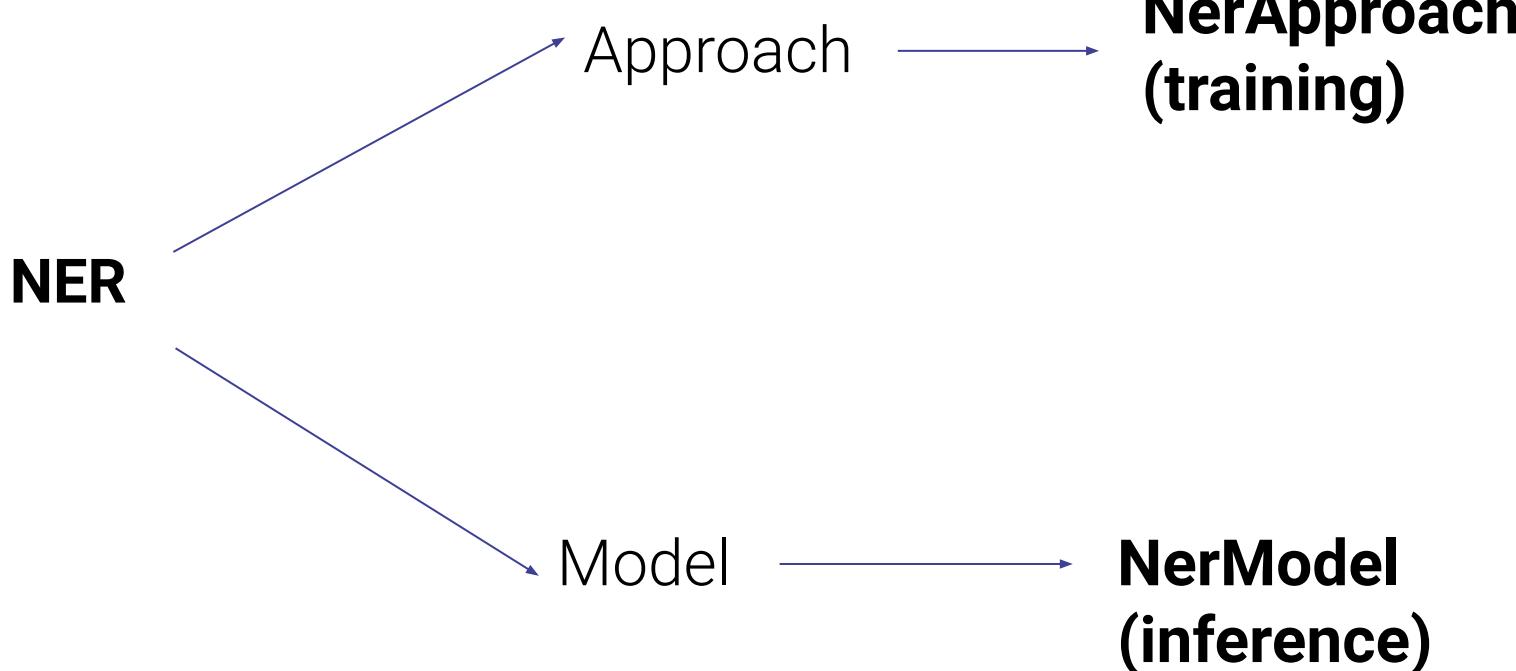
```
!pip install johnsnowlabs
```

```
nlp.install(force_browser=True)  
nlp.start()
```

Approach

Model

Training and Inference



This Named Entity recognition annotator allows to train generic NER model based on Neural Networks.

The architecture of the neural network is a Char CNNs - BiLSTM - CRF that achieves state-of-the-art in most datasets.

For instantiated/pretrained models, see NerDModel.

The training data should be a labeled Spark Dataset, in the format of [CoNLL 2003 IOB](#) with [Annotation](#) type columns. The data should have columns of type [DOCUMENT](#), [TOKEN](#), [WORD_EMBEDDINGS](#) and an additional label column of annotator type [NAMED_ENTITY](#). Excluding the label, this can be done with for example

- a [SentenceDetector](#)
- a [Tokenizer](#) and
- a [WordEmbeddingsModel](#) with clinical embeddings (any [clinical word embeddings](#) can be chosen).

For extended examples of usage, see the [Spark NLP Workshop](#) (sections starting with [Training a Clinical NER](#))

Input Annotator Types: DOCUMENT, TOKEN, WORD_EMBEDDINGS

Output Annotator Type: NAMED_ENTITY

Python API: [MedicalNerApproach](#) Scala API: [MedicalNerApproach](#)

Show Example

Python

Scala

Medical

Finance

Legal

```

from johnsnowlabs import *

# First extract the prerequisites for the NerDLApproach
documentAssembler = nlp.DocumentAssembler() \
.setInputCol("text") \
.setOutputCol("document")

sentence = nlp.SentenceDetector() \
.setInputCols(["document"]) \
.setOutputCol("sentence")

tokenizer = nlp.Tokenizer() \
.setInputCols(["sentence"]) \
.setOutputCol("token")

clinical_embeddings = nlp.WordEmbeddingsModel.pretrained('embeddings_clinical', "en", "clinical/models") \
.setInputCols(["sentence", "token"]) \
.setOutputCol("embeddings")

# Then the training can start
nerTagger = medical.NerApproach() \
.setInputCols(["sentence", "token", "embeddings"]) \
.setLabelColumn("label") \
.setOutputCol("ner") \
.setMaxEpochs(2) \
.setBatchSize(64)
  
```

NerModel.**pretrained**(...) loads a model trained with NerApproach and uploaded to ModelsHub.



STATE OF THE ART

Introduction Finance NLP

Finance NLP



Financial Entity Recognition	Financial Entity Linking	Assertion Status	Relation Extraction							
<p>BlackRock Energy and Resources Trust ORG (BGR) TICKER Ex-Dividend Date Scheduled for November 12, 2021 DATE</p> <p>There's A Lot To Like About ConnectOne Bancorp ORG's (NASDAQ:CNOB) TICKER Upcoming US\$0.13 AMOUNT Dividend</p>		<p>...the upcoming US\$10.3 dividend. → FUTURE</p> <p>...reported US\$1.4 benefits in 2022. → PAST</p> <p>...may end up signing a contract in 2025.. → POSSIBLE</p>	<p>ConnectOne Bancorp</p> <p>↓</p> <p>has_ticker NASDAQ: CNOB</p> <p>has_dividend US\$0.13 million</p> <p>has_date Now 12, 2021</p>							
<p>About ConnectOne Bank provides creative financial products and customized solutions</p> <p>Acquired by Center Bancorp 📍 Englewood Cliffs, New Jersey, United States 👤 251-500 💰 Post-IPO Debt 💻 Public 🌐 www.connectonebank.com/ 👤 25,400</p>	<p>Highlights</p> <table border="1"><tr><td>Stock Symbol NASDAQ:CNO > B</td><td>Acquisitions 1</td></tr><tr><td>Investments 2</td><td>Total Funding Amount \$50M</td></tr><tr><td>Contacts 203</td><td>Employee Profiles 3</td></tr></table>	Stock Symbol NASDAQ:CNO > B	Acquisitions 1	Investments 2	Total Funding Amount \$50M	Contacts 203	Employee Profiles 3	<p>Financial Embeddings</p> <p>Document Splitting</p> <p>Knowledge Graphs</p>	<p>Sentiment Analysis</p> <p>Deidentification</p> <p>Question & Answering</p>	<p>Text Classification</p> <p>Pattern Matching</p> <p>Table Understanding</p>
Stock Symbol NASDAQ:CNO > B	Acquisitions 1									
Investments 2	Total Funding Amount \$50M									
Contacts 203	Employee Profiles 3									
Trainable & Tunable	Scalable to a Cluster	Transformers	Fast Inference	Hardware Optimized						

Finance NLP Infrastructure

Free 30-days trial with guidance and support from our data scientists, software engineers and business experts!

- Install **on-premises (locally)**
 - **Fully compliant, air-gapped environments**
- Use it in the **cloud** with ready-to-use images in Databricks, AWS and Azure
- **Automatic scalability** in environments Databricks, AWS EMR and Azure HDInsight

Install Guide

Tell us what you need and we'll guide you how to get it.

Choose Product NLP Libraries Annotation Lab

Choose Edition Community Healthcare Finance Legal Visual / OCR

Where to Install on Premise **FREE TRIAL** on AWS Marketplace **FREE TRIAL** on Azure Marketplace **FREE TRIAL** on Databricks

Autopilot Options
 Enable autoscaling ?
 Terminate after minutes of inactivity ?

Worker Type New Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU | 2 8 Spot instances ?

Driver Type Same as worker 14.0 GB Memory, 4 Cores, 0.75 DBU |
 Advanced Options 1.508 x 1.226

John Snow Labs NLP Documentation



Spark NLP



Healthcare NLP
Legal NLP
Finance NLP



Visual NLP



NLP Lab



NLP Server



John Snow Labs NLP

NLP Models Hub

A place for sharing and discovering Spark NLP models and pipelines

Search models and pipelines 

Show All  models & pipelines in All Languages  for All versions 



13,728 Models & Pipelines Results:

Supported models only

<p>SUPPORTED</p> <p>Receipts Binary Classification</p> <p></p> <p>Date: 09.2022 task: Image Classification Language: English</p>	<p>SUPPORTED</p> <p>ESG Text Classification (Augmented, 26 classes)</p> <p></p> <p>Date: 09.2022 task: Text Classification Language: English</p>	<p>SUPPORTED</p> <p>Legal Zero-shot NER</p> <p></p> <p>Date: 09.2022 task: Named Entity Recognition Language: English</p>
--	--	---

Spark NLP in Action

Run 300+ live demos and notebooks

Clinical NLP

- De-Identification
- Diagnoses & Procedures
- Drugs & Adverse Events
- Labs, Tests, and Vitals
- Analyze Clinical Notes
- Radiology
- Oncology
- Resolve Entities to Terminology Codes
- Databricks Solution Accelerators
- Vaccines & Public Health
- Mental Health
- Risk Factors

Finance NLP

- Classify Financial Documents
- Recognize Financial Entities
- Understand Entities in Context
- Extract Financial Relationships
- Normalization & Data Augmentation
- Financial Deidentification
- Financial Document Splitting
- Text Summarization

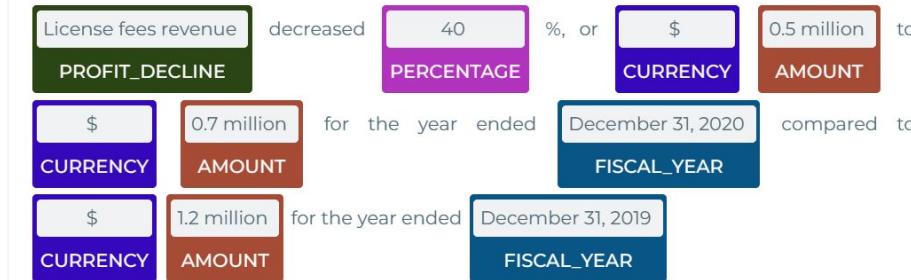
Legal NLP

- Classify Legal Texts

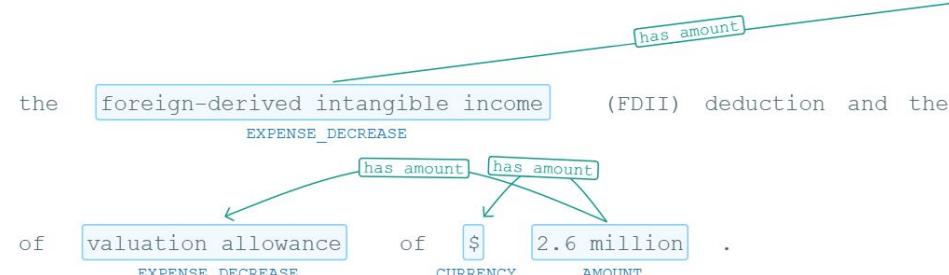
Choose Sample Text

License fees revenue decreased 40 %, or \$ 0.5 million to \$ 0.7 million for the year ended Dec...

Text annotated with identified Named Entities

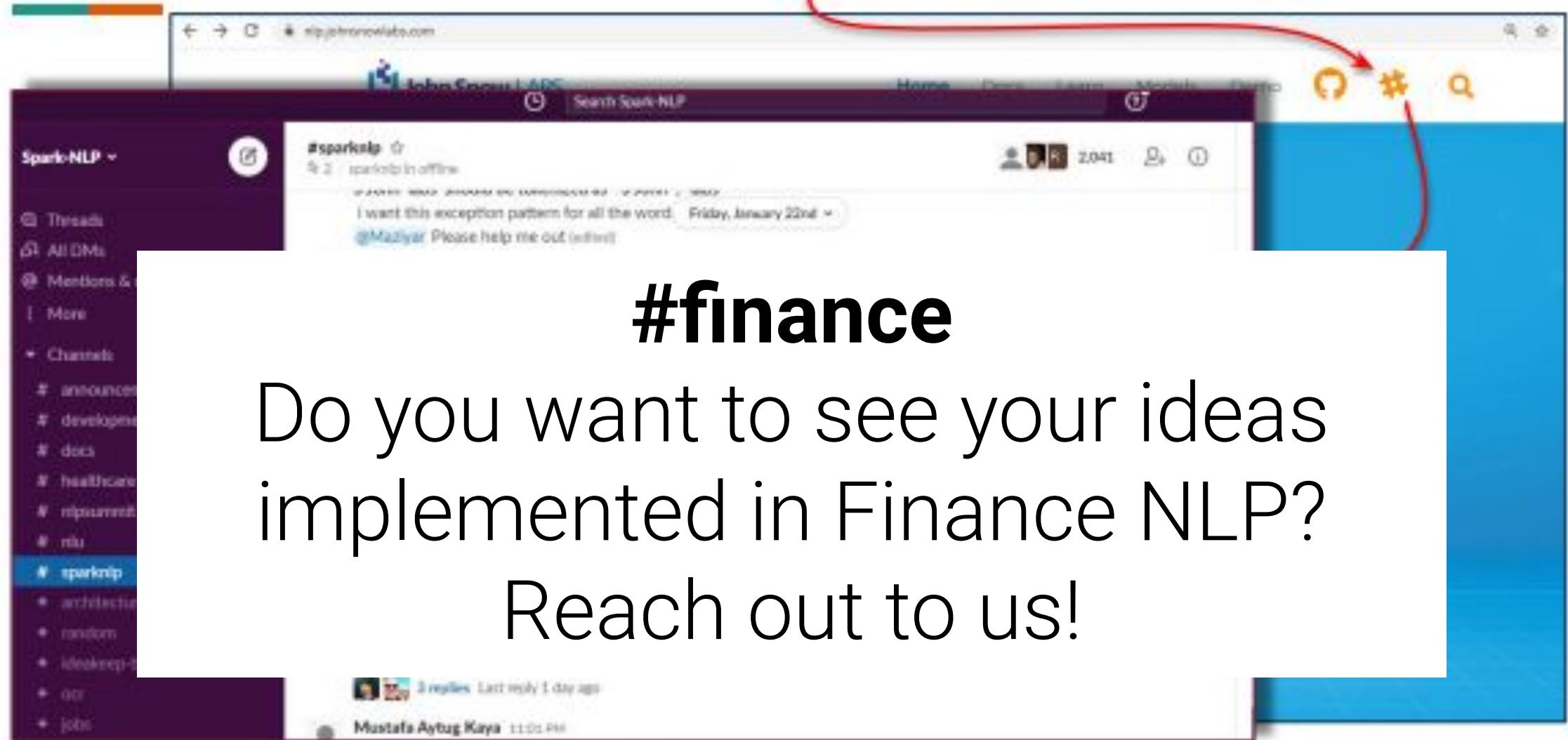


These benefits were partially offset by decreased benefits of



Spark-NLP Slack Channels

spark-nlp.slack.com



#finance

Do you want to see your ideas implemented in Finance NLP?
Reach out to us!

Mustafa Aytug Kaya 11:00 PM



Search Medium



Published in John Snow Labs



<https://medium.com/spark-nlp>



Jose Juan Martinez

Jan 19 · 4 min read · [Listen](#)



01/17/2023. Reuters. The Toro Company, formerly known as the **Toro Motor Company**, is an **American** company founded in **NATIONALITY** **FORMER_NAME**. The company was listed on **NASDAQ** in **August 2000**. It designs and operates **fawn mowers** **LISTING_DATE** **SECTOR** and **snow blowers** **SECTOR** and **irrigation system** **SECTOR** supplies.

Finance NLP 1.6.0: SEC Schedules, NASDAQ and Wikidata integration, and much more!



Credits to NexoCode for the image

We are happy to announce the Finance NLP 1.6.0 is out.

Finance NLP is a [John Snow Lab's](#) product, launched 2021. It is a state-of-the-art, autoscalable, domain-specific NLP on top of Spark.

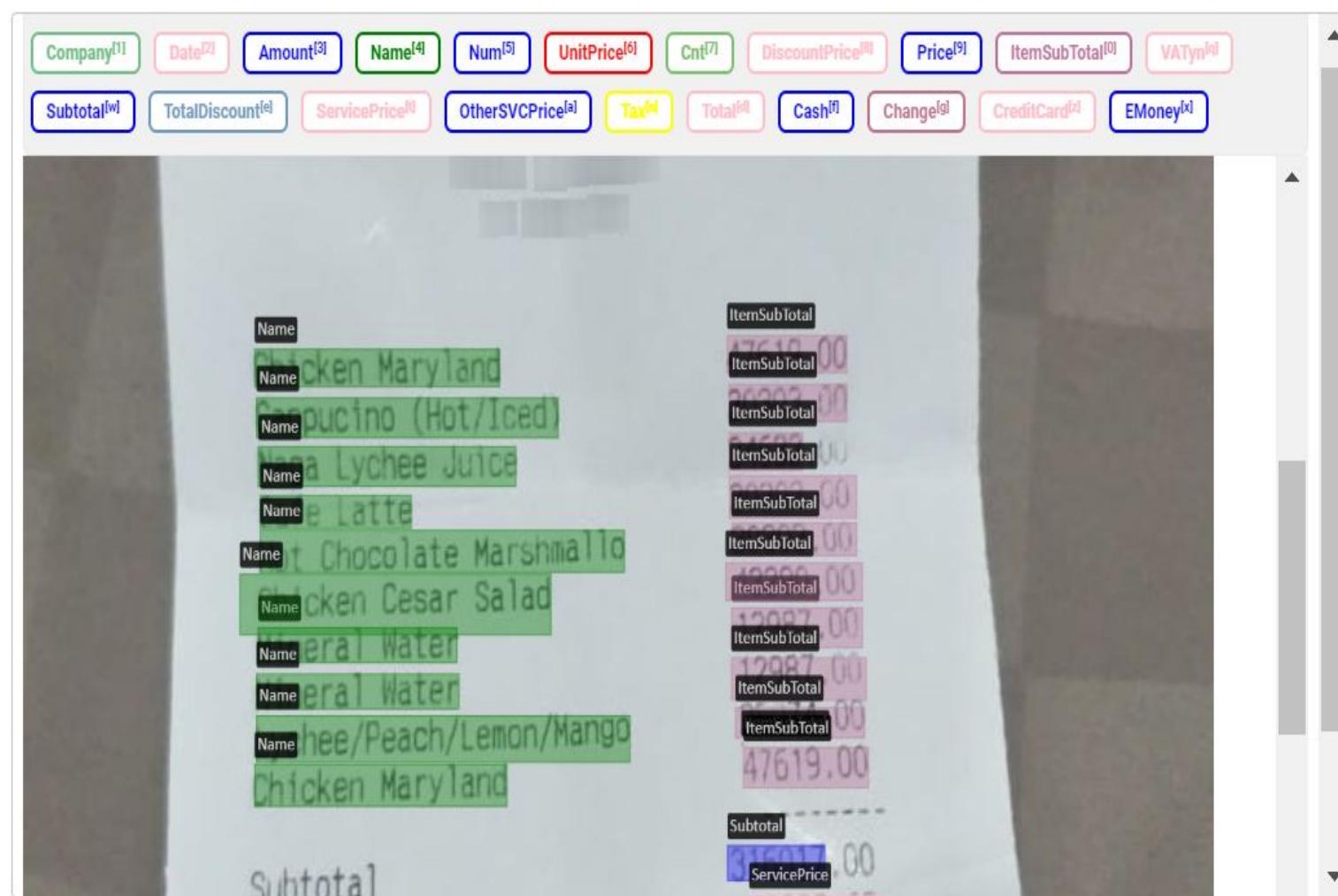
With more than 115 models, featuring Deep Learning and rule-based architectures, Finance NLP includes:

	precision	recall	f1-score	support
10-K	0.93	0.90	0.92	42
10-Q	0.95	0.95	0.95	38
3	0.62	0.61	0.62	33
4	0.82	0.78	0.80	54
8-K	0.86	0.91	0.88	33
S-8	0.93	0.96	0.95	28
other	1.00	1.00	1.00	238
schedule	0.94	0.96	0.95	50
accuracy			0.93	516
macro avg	0.88	0.88	0.88	516
weighted avg	0.93	0.93	0.93	516



STATE OF THE ART

**Finance NLP, Visual NLP and
NLP Lab**



[Ctrl+Space] Next ⏪

Annotations

Versions

LABELS

Sort

Name (10)

ItemSubTotal (11)

Subtotal (1)

ServicePrice (1)

Total (1)

CONNECTED WORDS (0)

No connected words

RELATIONS (0)

No relations added yet

Zero-shot and Prompt engineering in NLP Lab

The screenshot shows the John Snow LABS NLP Lab interface. The left sidebar includes sections for Projects, Hub (selected), NLP Models HUB, Models, Embeddings, Rules, Prompts (selected), and Settings. The main area is titled "HUB / Prompts" and "Prompts". It features a search bar with "Q city". A card for a prompt named "CITY" is displayed, created by "admin" one day ago, using a "HEALTHCARE" Reference LM Model. The card contains two numbered steps: "1 Which city?" and "2 Which is the city?". A context menu is open over the card, showing options: "Playground", "Edit", and "Delete". At the top right are buttons for "Add Prompt" and "Import Prompt". At the bottom, there are buttons for "View 15 Prompts per page", "Showing 1-1 of 1 Prompts", and navigation arrows.

John Snow LABS

HUB / Prompts

Prompts

Q city

CITY
NER | Created by: admin on 1 day ago

Reference LM Model: HEALTHCARE

1 Which city?
2 Which is the city?

Playground

Edit

Delete

View 15 Prompts per page

Showing 1-1 of 1 Prompts

admin

28



STATE OF THE ART

Text Splitting Finance NLP

Welcome - We have a lot of excited things ahead of us

Duration

Module

Notebooks

50 min

Introduction. Text Splitting. Tokenization. Embeddings. Binary, Multiclass, Multilabel Classification

01.Page_Splitting.ipynb
02.Sentence_Splitting_Tokenization.ipynb
03.Word_Sentence_EMBEDDINGS.ipynb
04.0.Document_Paragraph_Classification.ipynb
04.2.Training_Financial_Multiclass_Classifier.ipynb

10 min

break

60 min

Named Entity Recognition. Relation Extraction. Training. Zero-shot NER and RE.

05.0.NER_and_ZeroShotNER.ipynb
05.1.Training_Financial_NER.ipynb
06.0.Relation_Extraction.ipynb
06.2.ZeroShot_Relation_Extraction.ipynb

10 min

break

50 min

Understand Entities in Context with Assertion Status.
Question&Answering Models. Data Normalization, Mapping, Augmentation

07.0.Understand_Entities_in_Context.ipynb
08.0.Answering_Questions_Financial_Texts.ipynb
09.0.Normalization_with_Entity_Resolution_Edgar.ipynb
10.0.Data_Augmentation_with_ChunkMappers.ipynb

10 min

break

50 min

Financial Graph on a real use case. Deidentification. Integration with Visual NLP.

Databricks Solution Accelerator
11.0.Deidentification.ipynb
90.0.Financial_Visual_Classification.ipynb
90.1.Visual_and_Textual_Classification.ipynb

Splitting Financial texts

One of the first tasks when applying NLP to texts is **splitting**. Splitting means dividing the text into smaller chunks.

The main component to do that is **SentenceDetector**, a rule-based annotator, or **SentenceDetectorDL**, a pretrained, deep-learning based **Sentence Detector**. Don't get confused by the name, it could return whole **paragraphs or sections** as well using the setter `setCustomBounds()`. Other relevant setters: `setUseCustomBoundsOnly()` and `setCustomBoundsStrategy()`.

AGREEMENT

NOW, THEREFORE, for good and valuable consideration, and in consideration of the mutual covenants and conditions herein contained, the Parties agree as follows:

2. Definitions. For purposes of this Agreement, the following terms have the meanings ascribed thereto in this Section 1. 2.
Appointment as Reseller.

2.1 Appointment. The Company hereby [***]. Allscripts may also discontinue, suspend or terminate the provision of Processing Services and facilitate procurement of Merchant Processing Services from time to time, without limitation by references to such pricing information and Merchant Processing Services.

2.2 Customer Agreements.

a) Subscriptions. Allscripts and its Affiliates may sell Subscriptions for up to four (4) years on a subscription basis to Persons who subsequently execute a Customer Agreement. Allscripts may enter into Customer Agreements with terms longer than four (4) years with Persons who consent to such terms in each instance in writing in advance, which consent will not be unreasonably withheld by Allscripts.

```
text = """
4. GRANT OF KNOW-HOW LICENSE
4.1 Arizona Know-How Grant. Subject to the terms and conditions of this Agreement, Arizona hereby grants Allscripts a non-exclusive, transferable, worldwide license to use, copy, modify, and distribute the Know-How provided to Allscripts under this Agreement.
4.2 Company Know-How Grant. Subject to the terms and conditions of this Agreement, the Company hereby grants Allscripts a non-exclusive, transferable, worldwide license to use, copy, modify, and distribute the Know-How provided to the Company under this Agreement.
5. GRANT OF PATENT LICENSE
5.1 Arizona Patent Grant. Subject to the terms and conditions of this Agreement, Arizona hereby grants Allscripts a non-exclusive, transferable, worldwide license to use, copy, modify, and distribute the Patents provided to Allscripts under this Agreement.
"""

# Create a DocumentAssembler
documentAssembler = nlp.DocumentAssembler()\n    .setInputCol("text")\n    .setOutputCol("document")\n\n# Create a SentenceDetector\nsentenceDetector = nlp.SentenceDetector()\n    .setInputCols(["document"])\n    .setOutputCol("paragraph")\n    .setCustomBounds([\n        "\n[\d.]+"
    ])\n    .setCustomBoundsStrategy('prepend')\n\n# Create a Pipeline\npipeline = Pipeline(\n    stages=[documentAssembler, sentenceDetector])
```

```
documentAssembler = nlp.DocumentAssembler()\n    .setInputCol("text")\n    .setOutputCol("document")\n\n# Create a SentenceDetector\nsentenceDetector = nlp.SentenceDetector()\n    .setInputCols(["document"])\n    .setOutputCol("paragraph")\n    .setCustomBounds([\n        "\n[\d.]+"
    ])\n    .setCustomBoundsStrategy('prepend')\n\n# Create a Pipeline\npipeline = Pipeline(\n    stages=[documentAssembler, sentenceDetector])
```

```
documentAssembler = nlp.DocumentAssembler()\n    .setInputCol("text")\n    .setOutputCol("document")\n\n# Create a SentenceDetector\nsentenceDetector = nlp.SentenceDetector()\n    .setInputCols(["document"])\n    .setOutputCol("paragraph")\n    .setCustomBounds([\n        "\n[\d.]+"
    ])\n    .setCustomBoundsStrategy('prepend')\n\n# Create a Pipeline\npipeline = Pipeline(\n    stages=[documentAssembler, sentenceDetector])
```

Splitting Financial texts

One of the first tasks when applying NLP to texts is **splitting**. Splitting means dividing the text into smaller chunks.

The main component to do that is **SentenceDetector**, a rule-based annotator, or SentenceDetectorDL, a pretrained, deep-learning based **Sentence Detector**. Don't get confused by the name, it could return whole **paragraphs or sections** as well.

However, take into account that you may consider using **Visual NLP** to extract **Tabular information separately**, or you will lose layout information if you process it as a text. .

Transaction Activity						
In connection with repositioning our portfolio, and in furtherance of our real estate investment objectives, we have executed the following real estate transactions during 2020, 2019, and 2018. See Note 3, <i>Transactions</i> , of the accompanying consolidated financial statements for additional details.						
Acquisitions						
	Property	Location	% Acquired	Square Feet	Acquisition Date	Purchase Price (in thousands) ⁽¹⁾
2020						
Terminal Warehouse	New York, NY		8.65 %	1,200,000	March 13, 2020	\$ 40,048 ⁽²⁾
2019						
201 California Street	San Francisco, CA		100.00 %	252,000	December 9, 2019	\$ 238,900
101 Franklin Street ⁽³⁾	New York, NY		92.50 %	235,000	December 2, 2019	\$ 205,500
2018						
Lindbergh Center – Retail	Atlanta, GA		100.00 %	147,000	October 24, 2018	\$ 23,000
799 Broadway	New York, NY		49.70 %	182,000	October 3, 2018	\$ 30,200 ⁽²⁾

Splitting Financial texts

Sentence *Splitting* is very important for all the NLP use cases we are going to comment later on:

- For **Text Classification**, Sentence Detector will decide how much information will be sent to the Classifier.
 - Missing text could retrieve bad predictions
 - Passing too much may make the model ignore due to *token restrictions*, or get the *information mixed or deluded* (where you miss the key information in an ocean of other stuff).

'The IC and SoC design excellence requires technologies for custom IC, digital IC design and signoff, and functional verification, and leverages pre-built semiconductor IP. These tools, IP and associated services are specifically designed to meet the growing requirements of engineers designing increasingly complex chips across analog, digital and mixed-signal domains, and perform the associated verification tasks, including validation of low-level software running on the silicon model, thereby enabling design teams to manage complexity and verification throughput without commensurately increasing the team size or extending the project schedule, while reducing technical risks.\nThe second layer of our strategy centers around system innovation. It includes tools and services used for system design of the packages that encapsulate the ICs and the PCBs, system simulation which includes electromagnetic, electro-thermal and other multi-physics analysis necessary as part of optimizing the full system's performance, radio frequency ("RF") and microwave systems, and embedded software.\nThe third layer of our strategy addresses pervasive intelligence in new electronics. It starts with providing solutions and services to develop AI-enhanced systems and includes machine learning and deep learning capabilities being added to the Cadence\n\n technology portfolio to make IP and tools more automated and to produce optimized results faster.\nOur software and emulation products also support cloud access to address the growing computational needs of our customers.Recent Acquisitions During fiscal 2021, we continued to execute our Intelligent System Design strategy and expanded our product offerings and solutions into computational fluid dynamics ("CFD") with our acquisitions of Belgium-based NUMECA International, a leader in CFD technology, and Pointwise, Inc, a leading provider of CFD meshing technology. The addition of these technologies and talent broadens our System Design and Analysis portfolio and expertise. Chief Executive Officer Transition: On December 15, 2021, Anirudh Devgan assumed the role of President and Chief Executive Officer of Cadence, replacing Lip-Bu Tan. Prior to his role as Chief Executive Officer, Dr. Devgan served as President of Cadence. Concurrently, Mr. Tan transitioned to the role of Executive Chair.'



```
is_acquisitions? NO  
is_work_experience? NO
```

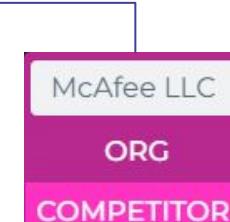
Splitting Financial texts

Sentence *Splitting* is very important for all the NLP use cases we are going to comment later on:

- For **NER**, in most cases, the information is contained in the **same sentence**, although in case of enumerations you may want to consider paragraph NER.
 - which are defined in the debt agreements, including:
 - limiting the ratio of secured debt,
 - requiring a fixed charge coverage ratio, and
 - limiting the ratio of debt.
- For **Assertion**, as with Text Classification, you may want to send the model more than just a sentence.

Our **competitors** include legacy antivirus product providers. The most relevant ones are:

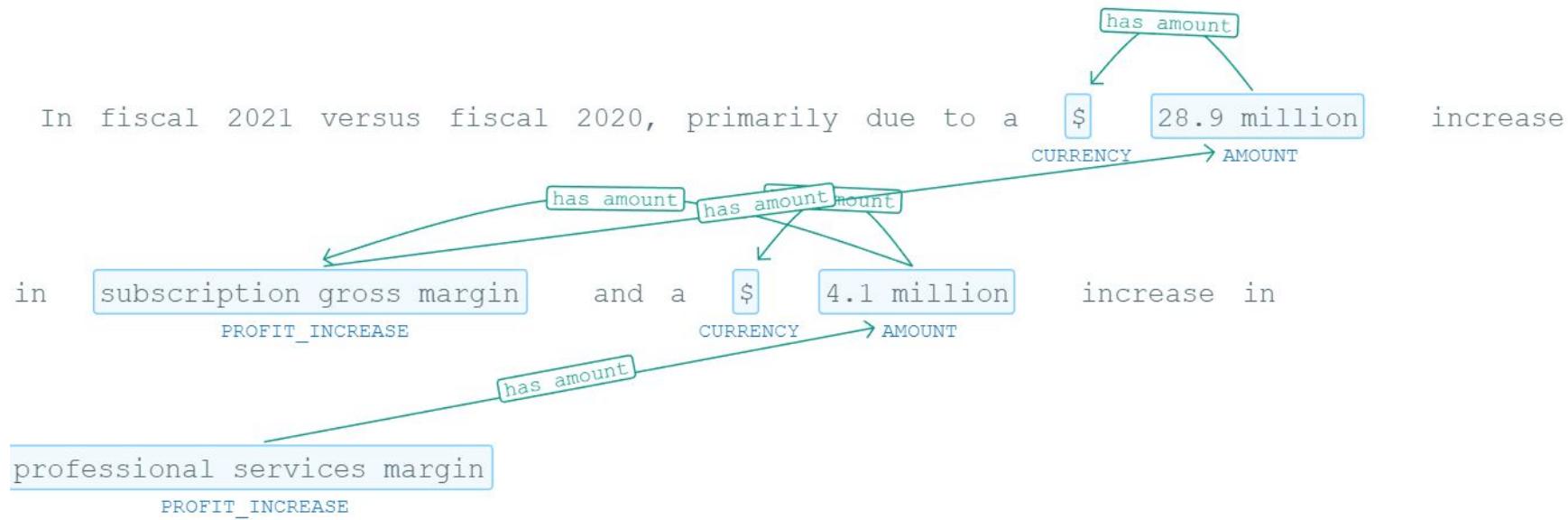
and



Splitting Financial texts

Sentence *Splitting* is very important for all the NLP use cases we are going to comment later on:

- For **Relation Extraction**, is quite common entities are in different sentences, so you may want to split by paragraph



Annotators

- **DocumentAssembler**: Assembles a Document Type from a text.
- **SentenceDetector**: Split Documents into sentences, pages, paragraphs, etc. using rules (regular expressions, characters, etc)
- **SentenceDetectorDL**: Deep Learning model (no rules, it's pretrained) to carry out sentence splitting exclusively.
- **Tokenizer**: Divides sentences into tokens (smaller pieces similar to words).
- **ChunkSentenceSplitter**: Uses detected entities in the document as boundaries to split documents (like headers and subheaders).



STATE OF THE ART

Language Models
Finance NLP

Welcome - We have a lot of excited things ahead of us

Duration

Module

Notebooks

50 min

Introduction. Text Splitting. Tokenization. Embeddings. Binary, Multiclass, Multilabel Classification

01.Page_Splitting.ipynb
02.Sentence_Splitting_Tokenization.ipynb
03.Word_Sentence_EMBEDDINGS.ipynb
04.0.Document_Paragraph_Classification.ipynb
04.2.Training_Financial_Multiclass_Classifier.ipynb

10 min

break

60 min

Named Entity Recognition. Relation Extraction. Training. Zero-shot NER and RE.

05.0.NER_and_ZeroShotNER.ipynb
05.1.Training_Financial_NER.ipynb
06.0.Relation_Extraction.ipynb
06.2.ZeroShot_Relation_Extraction.ipynb

10 min

break

50 min

Understand Entities in Context with Assertion Status.
Question&Answering Models. Data Normalization, Mapping, Augmentation

07.0.Understand_Entities_in_Context.ipynb
08.0.Answering_Questions_Financial_Texts.ipynb
09.0.Normalization_with_Entity_Resolution_Edgar.ipynb
10.0.Data_Augmentation_with_ChunkMappers.ipynb

10 min

break

50 min

Financial Graph on a real use case. Deidentification. Integration with Visual NLP.

Databricks Solution Accelerator
11.0.Deidentification.ipynb
90.0.Financial_Visual_Classification.ipynb
90.1.Visual_and_Textual_Classification.ipynb

Language Models and Embeddings

Language Models are Deep Learning objects you will use to process your texts. They are based on **Fill-mask** and **next-token prediction**, which means they learn the texts they see in training time and are able to predict a word if you mask it.

What we use from Language Models is not the fill-mask or next-token prediction, but the **numerical representation of the words** (or sentences), also called as **Embeddings**.

These numerical representations of words store information of their meaning in context.

The screenshot shows a dictionary entry for the word "bank".

bank²
/bæŋk/
noun
noun: bank; plural noun: banks

1. a financial establishment that uses money deposited by customers for investment, pays it out when required, makes loans at interest, and exchanges currency.
"a bank account"

Similar: financial institution, commercial bank, savings bank, finance company, ▾

• the store of money or tokens held by the banker in some gambling or board games.
noun: the bank

• the person holding the bank in some gambling or board games; the banker.

• INFORMAL • US
a large amount of money.
"those entrepreneurs are raking in some serious bank"

2. a stock of something available for use when required.
"a blood bank"

Similar: store, reserve, accumulation, stock, stockpile, inventory, supply, ▾

• a site or receptacle where something may be deposited for recycling.
"a paper bank"

3. a set of similar things, especially electrical or electronic devices, grouped together in rows.
"the DJ had big banks of lights and speakers on either side of his console"

Similar: array, row, line, tier, group, series, panel, console, ▾

• a tier of oars.
"the early ships had only twenty-five oars in each bank"

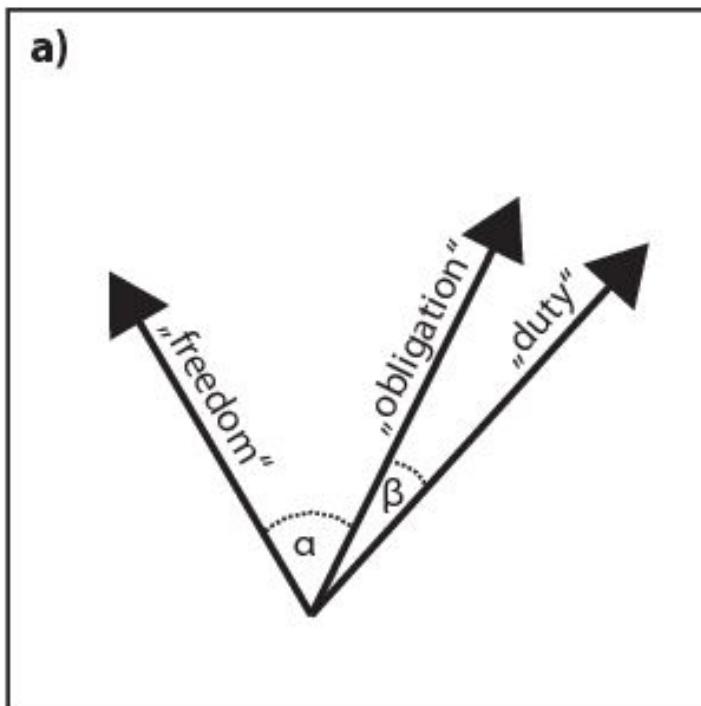
4. the cushion of a pool table.
"a bank shot"

All of these will have different embeddings (numerical representations) in context!

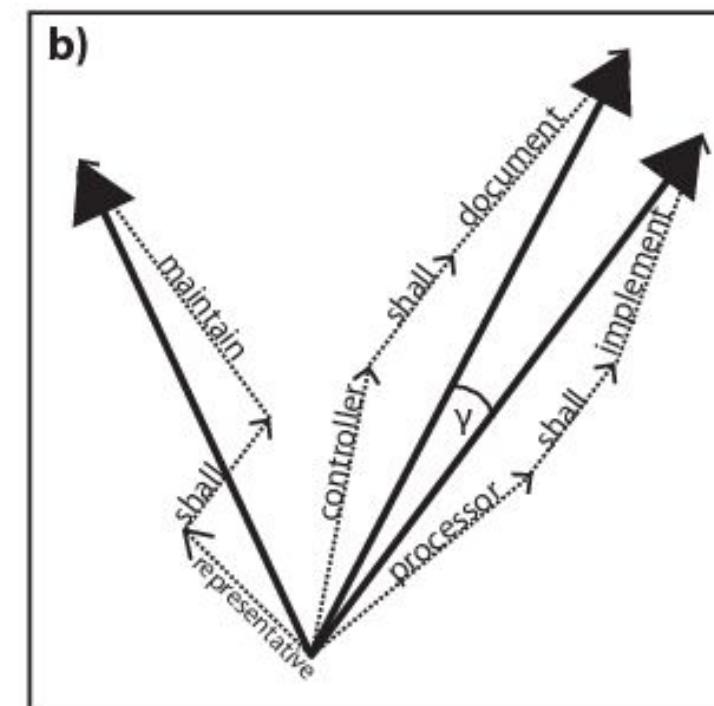
Language Models and Embeddings

We have two type of embeddings:

- **Word Embeddings**, for word-based NLP tasks, as:
 - Name Entity Recognition
 - Assertion Status
 - Relation Extraction, etc.
- **Sentence Embeddings**, for sentence/paragraph/document NLP tasks, as:
 - Text Classification
 - Entity Resolution



Finance Word Embeddings



Finance Sentence Embeddings

Language Models and Embeddings

Domain specificity

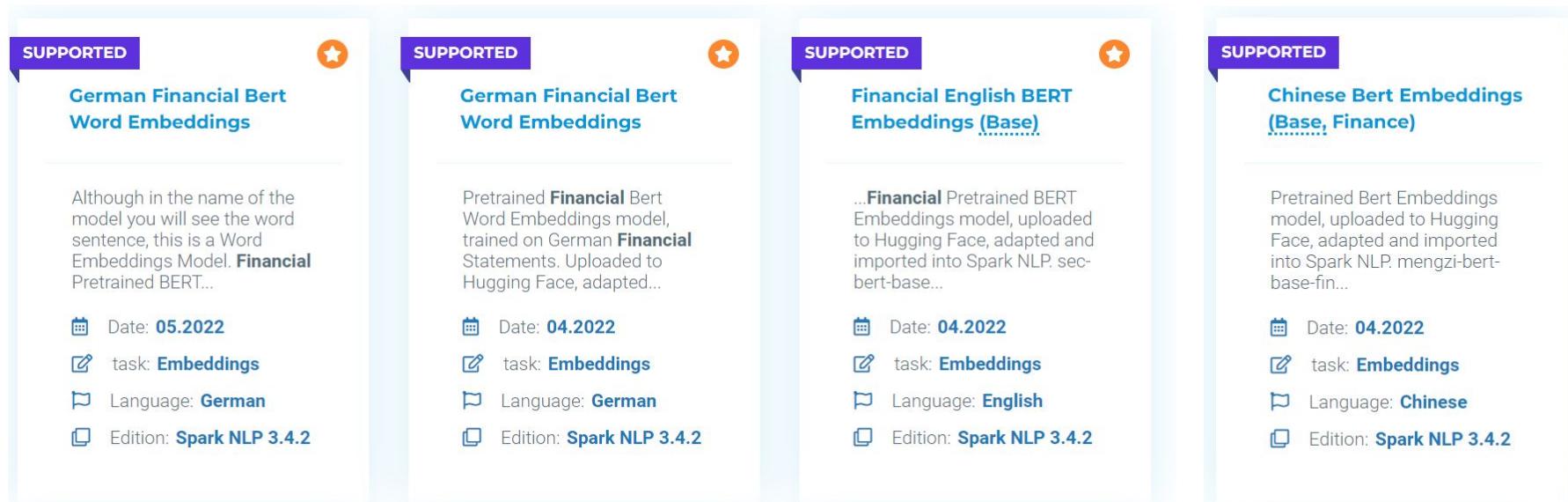
- As a consequence of their context-specificity, it's very important you use domain specific embeddings. Fortunately, we have **more than 15** Finance NLP Language Models in Models Hub, including English, German, Japanese and Chinese

Word vs Sentence

- If you don't find a proper Sentence Embeddings for you and you have a suitable Word Embeddings model, we provide with an **annotator called SentenceEmbeddings**, which will do the transformation for you.

Cased vs Uncased

- Please pay attention to the casing of the models. Some of them will require to lowercase the text first.



The image displays four cards from a Model Hub, each representing a different domain-specific language model:

- German Financial Bert Word Embeddings**: A "SUPPORTED" model. Description: "Although in the name of the model you will see the word sentence, this is a Word Embeddings Model. **Financial** Pretrained BERT...". Metadata: Date: 05.2022, task: Embeddings, Language: German, Edition: Spark NLP 3.4.2.
- German Financial Bert Word Embeddings**: A "SUPPORTED" model. Description: "Pretrained **Financial** Bert Word Embeddings model, trained on German **Financial** Statements. Uploaded to Hugging Face, adapted...". Metadata: Date: 04.2022, task: Embeddings, Language: German, Edition: Spark NLP 3.4.2.
- Financial English BERT Embeddings (Base)**: A "SUPPORTED" model. Description: "...**Financial** Pretrained BERT Embeddings model, uploaded to Hugging Face, adapted and imported into Spark NLP. sec-bert-base...". Metadata: Date: 04.2022, task: Embeddings, Language: English, Edition: Spark NLP 3.4.2.
- Chinese Bert Embeddings (Base, Finance)**: A "SUPPORTED" model. Description: "Pretrained Bert Embeddings model, uploaded to Hugging Face, adapted and imported into Spark NLP. mengzi-bert-base-fin...". Metadata: Date: 04.2022, task: Embeddings, Language: Chinese, Edition: Spark NLP 3.4.2.

AN NLP TIMELINE AND THE TRANSFORMER FAMILY

BAG OF WORDS (BOW)

Count the occurrences of each word in the documents and use them as features.

1954

TF-IDF

The BOW scores are modified so that rare words have high scores and common words have low scores.

1972

WORD2VEC

Each word is mapped to a high-dimensional vector called word embedding, which captures its semantic. Word embeddings are learned by a neural network looking for word correlations on a large corpus.

2013

RNN

RNNs compute document embeddings leveraging word context in sentences, which was not possible with word embeddings alone.

LSTM

Capture long term dependencies.

1997

Bidirectional RNN

Capture left-to-right and right-to-left dependencies.

1997

Encoder-decoder RNN

An RNN creates a document embedding (i.e. the encoder) and another RNN decodes it into text (i.e. the decoder).

2014

1966

TRANSFORMER

An encoder-decoder model that leverages attention mechanism to compute better embeddings and to better align output to input.

2017

BERT

Bidirectional Transformer pretrained using a combination of Masked Language Modeling and Next Sentence Prediction objectives. It uses global attention.

2018

GPT

The first autoregressive model based on the Transformer architecture.

2019

GPT-2

A bigger and optimized version of GPT, pre-trained on WebText.

2019

GPT-3

A bigger and optimized version of GPT-2, pre-trained on Common Crawl.

2020

CTRL

Similar to GPT but with control codes for conditional text generation.

2018

TRANSFORMER-XL

It's an autoregressive Transformer which can reuse previously computed hidden-states to attend to longer context.

2019

ALBERT

A lighter version of BERT, where (1) Next Sentence Prediction is replaced by Sentence Order Prediction, and (2) parameter-reduction techniques are used for lower memory consumption and faster training.

2019

ROBERTA

Better version of BERT, where (1) the Masked Language Modeling objective is dynamic, (2) the Next Sentence Prediction objective is dropped, (3) the BPE tokenizer is employed, and (4) better hyperparameters are used.

2019

XLM

Transformer pre-trained on a corpus of several languages using objectives like Causal Language Modeling, Masked Language Modeling, and Translation Language Modeling.

2019

XLNET

Transformer-XL, with a generalised autoregressive pre-training method that enables learning bidirectional dependencies.

2019

PEGASUS

A bidirectional encoder and a left-to-right decoder pre-trained with Masked Language Modeling and Gap Sentence Generation objectives.

2019



NLPLANET

The community of
NLP enthusiasts!

DISTILBERT

Same as BERT but smaller and faster, while preserving over 95% of BERT's performances. Trained by distillation of the pre-trained BERT model.

2019

XLM-ROBERTA

RoBERTa trained on a multilingual corpus with the Masked Language Modeling objective.

2019

BART

A bidirectional encoder and a left-to-right decoder trained by corrupting text with an arbitrary masking function and learning a model to reconstruct the original text.

2019

CONVBERT

Better version of BERT, where self-attention blocks are replaced with new ones that leverage convolutions to better model global and local context.

2019

FUNNEL TRANSFORMER

A type of Transformer that gradually compresses the sequence of hidden states to a shorter one and hence reduces the computation cost.

2020

REFORMER

A more efficient Transformer thanks to local-sensitive hashing attention, axial position encoding and other optimizations.

2020

T5

A bidirectional encoder and a left-to-right decoder pre-trained on a mix of unsupervised and supervised tasks.

2020

LONGFORMER

A Transformer model replacing the attention matrices with sparse matrices for higher training efficiency.

2020

PROPHETNET

A Transformer model trained with the Future N-gram Prediction objective and with a novel self-attention mechanism.

2020

ELECTRA

Same as BERT but lighter and better. The model is trained with the Replaced Token Detection objective.

2020

SWITCH TRANSFORMER

A sparsely-activated expert Transformer model that aims to simplify and improve over Mixture of Experts.

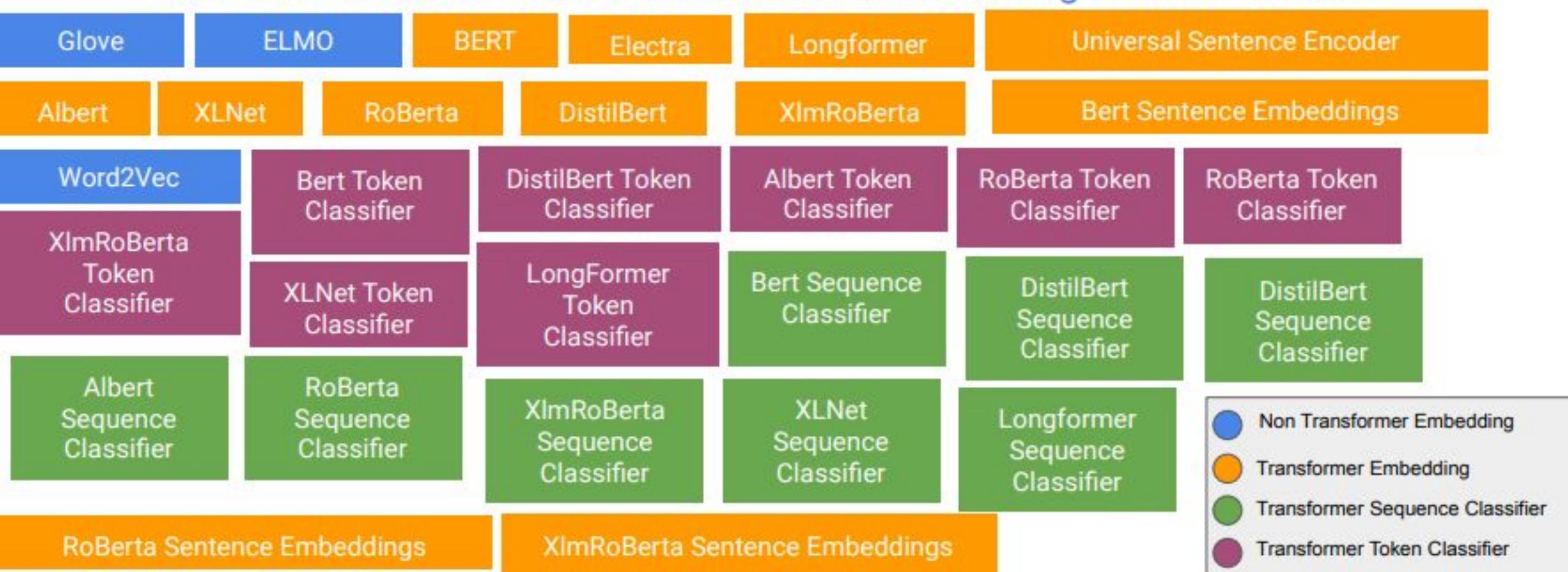
2021

<https://www.linkedin.com/company/nlplanet/>

<https://medium.com/nlplanet/>

https://twitter.com/nlplanet_

Text Classification with Word & Sentence Embeddings & Transformers



Spark NLP

- ClassifierDL
- SentimentDL
- MultiClassifierDL
- Sequence Classifier
- Token Classifier

Annotators

- **BertEmbeddings**: Gets BERT embeddings for each token using a pretrained Language Model.
- **RoBertaEmbeddings**: Gets RoBerta embeddings for each token using a pretrained Language Model.
- ...
- **UniversalSentenceEncoder**: Gets USE embeddings for a whole sentence / text.
- **BertSentenceEmbeddings**: Gets USE embeddings for a whole sentence / text.
- **SentenceEmbeddings**: Averages / Pools any Word Embedding model to get Sentence Embeddings



STATE OF THE ART

Classification Finance NLP

Welcome - We have a lot of excited things ahead of us

Duration

Module

Notebooks

50 min

Introduction. Text Splitting. Tokenization. Embeddings. Binary, Multiclass, Multilabel Classification

01.Page_Splitting.ipynb
02.Sentence_Splitting_Tokenization.ipynb
03.Word_Sentence_EMBEDDINGS.ipynb
04.0.Document_Paragraph_Classification.ipynb
04.2.Training_Financial_Multiclass_Classifier.ipynb

10 min

break

60 min

Named Entity Recognition. Relation Extraction. Training. Zero-shot NER and RE.

05.0.NER_and_ZeroShotNER.ipynb
05.1.Training_Financial_NER.ipynb
06.0.Relation_Extraction.ipynb
06.2.ZeroShot_Relation_Extraction.ipynb

10 min

break

50 min

Understand Entities in Context with Assertion Status.
Question&Answering Models. Data Normalization, Mapping, Augmentation

07.0.Understand_Entities_in_Context.ipynb
08.0.Answering_Questions_Financial_Texts.ipynb
09.0.Normalization_with_Entity_Resolution_Edgar.ipynb
10.0.Data_Augmentation_with_ChunkMappers.ipynb

10 min

break

50 min

Financial Graph on a real use case. Deidentification. Integration with Visual NLP.

Databricks Solution Accelerator
11.0.Deidentification.ipynb
90.0.Financial_Visual_Classification.ipynb
90.1.Visual_and_Textual_Classification.ipynb

Ecuador posted a trade surplus of 10.6 mln dls in the first four months of 1987 compared with a surplus of 271.7 ~~mln~~ in the same period in 1986...

Category: **finance, trade**

Classification Confidence: **99.86%**

I have been waiting over a week. Is the card still coming?

This sentence has been classified as : **Card_Arrival**

Classification Confidence: **99.29%**

Subadviser shall be compensated for the services it performs on behalf of the Fund in accordance with the terms set forth in Appendix A to this agreement.

Class: **Specific FLS**

Classification Confidence: **99.86%**

As filed with the SEC on July 27, 2016 Registration No. 333-
SECURITIES AND EXCHANGE COMMISSION WASHINGTON, D.C.
20549 FORM S-8
REGISTRATION

This document has been classified as : **S-8**

Classification Confidence: **99.99999%**

The company made an investment into Climate Vision, the science and research company that first identified a potential link between global warming and rising temperatures.

Class: **Environmental**

Classification Confidence: **99.96%**

Se'1 Sepi Sembel Lu'at (R)	20.000
Lidah Asap Sembel Lu'at (R)	30.000
Kast Putih	10.000
Jus Jeruk	12.000
Subtotal	72.000
PB 1 (10X)	7.200
Total	79.200
Tuna!	100.000
Kebab!!	20.000

This document has been classified as : **ticket**

Classification Confidence : **99.6%**

Text Classification

Finance NLP Classification

Text Classification is the NLP Task in charge of retrieving a **class/category** per input text.

- **Classification** require domain **Sentence Embeddings**. Remember, if you don't find proper sentence embeddings, you can use SentenceEmbeddings annotator to transform your word embeddings into SentenceEmbeddings.

We count on more than 30 Text Classifiers, which can be divided using 2 categorization systems:

- By **Input** type or type of **text splitting needed**

Sentences	Clauses / Paragraphs / Sections	Whole Documents
To do classification at sentence level. For example, detecting sentiment on a sentence, if a sentence talks about a specific topic , etc.	They can be used to identify if a piece of texts bigger than a sentence (a paragraph) is of a specific class. Very useful to detect Items in, for example, 10K filings	To carry out Document Classification. Bear in mind current NLP Models are not able to process big texts. The biggest amount of text we can process is using Longformers with 4096 tokens , or using Bert-based models with 512 . The rest of the text will be discarded. However, the good news is that in most cases, the information to classify a document is in the first page of it.

Finance NLP Classification

- By **output type or class assigned to the input text**

Binary Classifiers

Return *true* or *false* values. For example, our more binary classifiers, which return the **name of the clause** if it is classified as such, or **other** otherwise.

ITEM 1. BUSINESS
The following discussion, as well as other portions of this Form 10-K contain forward-looking statements that reflect our plans, estimates and beliefs. Any such forward-looking statements (including, but not limited to, statements to the effect that Tandy Leather Factory, Inc. ("TLFI") or its management "anticipates," "plans," "estimates," "expects," "believes," "intends," and other similar expressions) that are not statements of historical fact should be considered forward-looking statements and should be read in conjunction with our Consolidated Financial Statements and related notes contained elsewhere in this report. These forward-looking statements are made based upon management's current plans, expectations, estimates, assumptions and beliefs concerning future events impacting us. You should be read carefully because they involve risks and uncertainties. We are also obliged to specify certain forward-looking statements in accordance with rules required by law. Such forward-looking statements include statements regarding our forecasts of financial performance, share repurchases, store openings or store closings, capital expenditures and working capital requirements. Our actual results could materially differ from those discussed in such forward-looking statements. Factors that could cause or contribute to such differences include, but are not limited to, those discussed below and elsewhere in this Form 10-K and particularly in "Item 1A. Risk Factors" and "Item 7. Management's Discussion and Analysis of Financial Condition and Results of Operations." Unless the context otherwise indicates, references in this Form 10-K to "TLFI," "we," "our," "us," the "Company," "Tandy," or "Tandy Leather" mean Tandy Leather Factory, Inc., together with its subsidiaries.



Tandy Leather Factory, Inc.

(Exact name of registrant as specified in its charter)
Delaware (I.R.S. Employer Identification No.)
75-2543540

(State or other jurisdiction of incorporation or organization)
1900 Southeast Loop 820, Fort Worth, TX
76140 (Address of Principal Executive Offices and Zip Code)
817/872-3200 (Registrant's telephone number, including area code)

Securities registered pursuant to Section 12(b) of the Act:
Title of each class Name of each exchange on which registered
Common Stock NASDAQ Global Market

Securities registered pursuant to Section 12(g) of the Act:
None

finclf_10k_summary

10k

other

Multiclass classifiers

Returns 1 value from all the categories the model was trained on. Only works for models with a small number of categories (up to 100).

It's not suitable for:

- Big number of classes (more than 100)
- Non-disjoint classes (a text can be of several classes at the same time)

The company made an investment into Climate Vision, the science and research company that first identified a potential link between global warming and rising temperatures.

Class: Environmental
or Social or Governance

Multilabel classifiers

Returns n value from all the categories the model was trained on. Only works for models with a small number of categories (up to 100).

It's not suitable for:

- Big number of classes (more than 100)

Ecuador posted a trade surplus of 10.6 mln dls in the first four months of 1987 compared with a surplus of 271.7 mln in the same period in 1986...

Category: **finance, trade**

Classification Confidence: **99.86%**

Classifying Images

Sometimes you may have the image or a scanned pdf document and not the text. There are several ways you can go with the Spark NLP Suite.

If there is no layout or it is not relevant:

- 1) Use **Visual NLP to extract the text** and Use **Finance NLP Text Classifiers**.

If the layout information is important:

- 2) Use **Finance NLP Visual Transformers (ViT)** to train at image level.
- 3) Use **Visual NLP to use the text and the layout** of a document to train a classifier. No Finance NLP required.



Here there is no layout, so just extracting text and using a **Text Classifier** may be enough

I have been following your company and work for many years. I am pleased to discover that you are looking for an experienced Financial Services Associate to join your team. Not only I believe that the combination of my career history, field experience, and developed skills set makes me an ideal candidate for the role but I am also certain that it would be a great opportunity for me to grow my career.
My name is Rolien Gasner and I am the American University graduate with a bachelor's degree in Economics & Finance. I graduated in 2018 with a 3.9 GPA and was a member of the Dean's List. I am also a member of the Alpha Gamma Delta Honor Society and I also won the Dean's Award once for representing the school at multiple international economics competitions. My studies have allowed me to become an effective leader and helped me to acquire excellent analytical and communication skills.

Next, I worked as a Financial Services Associate at Viteo Financial, Ltd. for more than 2 years. There, I spent most of my time providing professional financial advice and recommendations to clients, identifying their needs and goals, and conducting financial portfolio analysis. On top of that, I completed yearly credit review and risk analysis for specific clients. I also spent time working on financial modeling projects. Many times that I am a pro-active and reliable person with the crucial ability to function well in deadline-driven and fast-paced environment. I have been able to prove myself a top-notch employee by the company executives for meeting all assigned tasks. Offering the experience and all skills that are pre-requisite necessary for the job, am a native Dutch speaker with a proficiency in English and a basic knowledge of French and German. Thank you for your time and consideration and I look forward to speaking with you in the near future.

Sincerely,
Rolien Gasner

PERSONAL FINANCIAL STATEMENT			
Section 1(a) - Personal Information		Date: 06/25/04	
Name: Joe T. Example	Birthdate: 05/05/55	SSN: 123-45-6789	
Address: Any Street	City: Any City	State: TX	Zip: 11111
Employer: Any Employer	Position: Any position	# of Years: 10	
Employer's Address: Any Employer's address	City: Any Employer's city	State: TX	Zip: 45678
Business Phone: 333-444-5555	Residence Phone: 222-333-4444	Drivers License #: 123456789	
Section 1(b) - Other Party Information			
Name: Jane T. Example	Birthdate: 06/06/56	SSN: 339-44-6666	
Address: Any Street	City: Any City	State: TX	Zip: 11111
Employer: Any Employer # 2	Position: Any position	# of Years: 10	
Employer's Address: Any Employer's address	City: Any Employer # 2 Address	State: TX	Zip: 32145
Business Phone: 111-222-3333	Residence Phone: 222-333-4444	Drivers License #: 9874565	
Section 2 - Statement of Financial Condition			
Assets	In Dollars	Liabilities	In Dollars
Cash: (Refer to Schedule A)	\$15,000.00	Notes Payable: (Refer to Schedule D)	\$45,000.00
Securities: (Refer to Schedule B)	\$1,500.00	Mortgages Payable: (Refer to Schedule C)	\$50,000.00
Real Estate: (Refer to Schedule C)	\$150,000.00	All Other Liabilities:	\$3,500.00
Automobiles:	\$38,000.00		
Restricted or Margin Accounts:	\$0.00		
Cash Value Life Insurance:	\$0.00		
Accounts and Notes Receivable:	\$0.00		
Household & Personal Assets:	\$0.00	Total Liabilities:	\$98,500.00
All Other Assets:	\$0.00	Net Worth: (Total Assets - Total Liabilities)	\$105,000.00
Total Assets:	\$204,500.00	Total Liabilities & Net Worth:	\$204,500.00
Section 3 - Annual Income		Annual Expenditures	
Federal Income Tax Information for the Year Ended 2003			
Salaries & Wages (Individual):	\$75,000.00	Instalment Payments (auto, credit cards)	\$10,200.00
Salaries & Wages (Spouse):	\$75,000.00	Lease Obligations:	\$0.00
Bonuses & Commissions:	\$0.00	Mortgage/Rental Payments:	\$0.00
Dividends & Interest Income:	\$0.00	Other Debt Service:	\$0.00
Net Real Estate Income:	\$0.00	Alimony, Child Support, etc:	\$0.00
Oil & Gas Income/Royalties:	\$0.00	Auto, Life & Health Insurance Premiums:	\$0.00
All Other Income:	\$0.00	All Other Expenditures:	\$0.00

But here the layout is super important, it's better to keep that information.

You can use:

- Finance NL Visual Transformers (ViT) to classify at **image** level
- Visual NLP to classify at **text+layout**

Annotators

Embeddings-based:

- **ClassifierDL**: A Deep Learning architecture (multilayer perceptron) using Sentence Embeddings as input. Produces binary, multiclass or multilabel models.
- **BertForSequenceClassification**: Same as ClassifierDL but using the BERT transformer.

Features-based:

- **GenericClassifier**: A multilayer perceptron but with a vector of features instead of sentence embeddings, as an input.
- **DocumentLogRegClassifier**: Similar to GenericClassifier but for Logistic Regression.

Images-based:

- **ViTForImageClassification**: Image (pixel only!) classification

[VISUAL NLP]

- **VisualDocumentClassifier**: Text + layout classification



STATE OF THE ART

**Named Entity Recognition
Finance NLP**

Welcome - We have a lot of excited things ahead of us

Duration

Module

Notebooks

50 min

Introduction. Text Splitting. Tokenization. Embeddings. Binary, Multiclass, Multilabel Classification

01.Page_Splitting.ipynb
02.Sentence_Splitting_Tokenization.ipynb
03.Word_Sentence_EMBEDDINGS.ipynb
04.0.Document_Paragraph_Classification.ipynb
04.2.Training_Financial_Multiclass_Classifier.ipynb

10 min

break

60 min

Named Entity Recognition. Relation Extraction. Training. Zero-shot NER and RE.

05.0.NER_and_ZeroShotNER.ipynb
05.1.Training_Financial_NER.ipynb
06.0.Relation_Extraction.ipynb
06.2.ZeroShot_Relation_Extraction.ipynb

10 min

break

50 min

Understand Entities in Context with Assertion Status.
Question&Answering Models. Data Normalization, Mapping, Augmentation

07.0.Understand_Entities_in_Context.ipynb
08.0.Answering_Questions_Financial_Texts.ipynb
09.0.Normalization_with_Entity_Resolution_Edgar.ipynb
10.0.Data_Augmentation_with_ChunkMappers.ipynb

10 min

break

50 min

Financial Graph on a real use case. Deidentification. Integration with Visual NLP.

Databricks Solution Accelerator
11.0.Deidentification.ipynb
90.0.Financial_Visual_Classification.ipynb
90.1.Visual_and_Textual_Classification.ipynb

Recognize Financial Entities in Documents

License fees revenue decreased 40 %, or \$ 0.5 million to \$ 0.7 million for the year ended PROFIT_DECLINE PERCENTAGE CURRENCY AMOUNT CURRENCY AMOUNT December 31, 2020 compared to \$ 1.2 million for the year ended December 31, 2019 FISCAL_YEAR FISCAL_YEAR

Borrowings under this facility were \$ 53.8 million and \$ 36.3 million as of May 31, 2016 and November 30, 2015, respectively.

ADDRESS

(Address of Principal Executive Offices including Zip Code)

(408) 727-1885

PHONE

(Registrant's Telephone Number, Including Area Code)

Securities registered pursuant to Section 12(b) of the Act:

Title of each class

Trading Symbol

Name of each exchange on which registered

COMMON STOCK , PAR VALUE \$.001 PER SHARE
TITLE_CLASS TITLE_CLASS_VALUE

EGHT
TICKER

New York Stock Exchange
STOCK_EXCHANGE

Common Stock The authorized capital of the Company is 200,000,000 common shares, par value \$ COMMONSTOCKSHARESAUTHORIZED

0.001 , of which 12,481,724 are issued or outstanding.

COMMONSTOCKPARORSTATEDVALUEPERSHARE COMMONSTOCKSHARESOUTSTANDING

Compliance: Policy Compliance (PC), Security Configuration Assessment (SCA), PCI Compliance (PCI), File Integrity Monitoring (FIM), Security Assessment Questionnaire (SAQ), Out-of-Band Configuration Assessment (OCA);

Finance NLP Named Entity Recognition



NER is the NLP task in charge of detecting relevant words / chunks in texts and categorize them.

- **NER** requires **Word embeddings**.
- As with Classification, **NER** also requires **splitting**. Usually, the split is done at the **sentence** level, but there may be cases where you would like to provide to the NER model more context than a sentence:

Sentences	Paragraphs
<p>To do NER at sentence level, after you split a text into sentences with SentenceDetector.</p> <p>Used in most of the cases, since the context of a relevant entity is found in the surroundings of its sentence.</p>	<p>To do NER at sentence level, after you split a text into paragraphs with SentenceDetector, not into sentences.</p> <p>We may need to do this in some exceptional cases:</p> <p>which are defined in the debt agreements, including:</p> <ul style="list-style-type: none">- limiting the ratio of secured debt,- requiring a fixed charge coverage ratio, and- limiting the ratio of debt.

Finance NLP Named Entity Recognition



We provide with **Finance NER** at **clause** and **document level**.

Clause Level

NER entities can be only found in some specific parts of the document. For example, **Ticker**, **Address**, **Title of Each Class**, **CIK**, **IRS** in a **10k filing**.



There is **no point** in applying the **10k summary NER** models to the whole document, since:

- It will affect drastically the performance
- It will retrieve more false positives / negatives

In order to carry out NER of specific clauses, please use first Text Classification, as described before, and if the specific class you have detected is relevant for your, apply its specific NER.

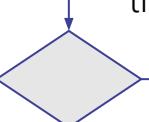


paragraph splitting

Indemnification and contribution. (a) The Company agrees to indemnify and hold harmless each Underwriter, the directors, officers, employees and agents of each Underwriter and each person who controls any Underwriter within the meaning of either the Act or the Exchange Act against any and all losses, claims, damages or liabilities, joint or several, to which they or any of them may become subject under the Act, the Exchange Act or other Federal or state statutory law or regulation, at common law or otherwise, insofar as such losses, claims, damages or liabilities result from any untrue statement of a material fact contained in the registration statement or any amendment thereto or in the Free Writing Prospectus, any Preliminary Final Prospectus or the Final Prospectus, or any statement or amendment filed in or supplement thereto or any statement or amendment filed in connection with the offering, if any, of the Securities, or in any documents or materials furnished by the Company to any Underwriter, or if any statement therein is untrue in any material respect or is omitted therein by reason of negligence on the part of the Company, or if any statement therein is misleading. The Company agrees to state therein a material fact required to be stated therein or necessary to make the statements therein not misleading, and agrees to reimburse each such indemnified party, as soon as practicable, for any amounts paid by it in respect of any such loss, claim, damage or liability. The Company shall not be liable for any such loss, claim, damage, liability or actions provided, however, that the Company will not be liable in any such case to the extent that any such loss, claim, damage or liability arises out of or is based upon any untrue statement or omission made in a document referred to in this paragraph if such untrue statement or omission was made therein in reliance upon and in conformity with written information furnished to the Company by or on behalf of any Underwriter through the representatives specifically for inclusion therein. This indemnity agreement will be in addition to any liability which the Company may otherwise have.

10k_summary ?

true?



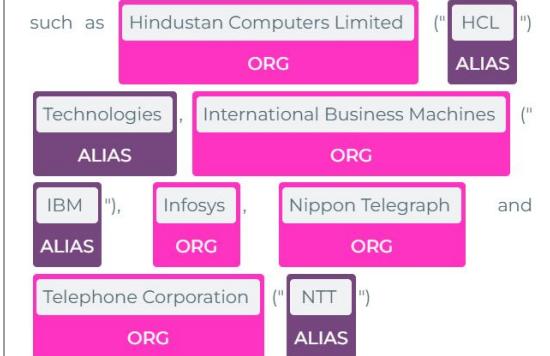
**10k_summary
NER**

Document Level

NER entities can be only found **all over** the document. For example, if you are looking for mention of companies, products, etc.

You can apply NER to the whole document.

Our channel partners include security consulting organizations, managed service providers and resellers,



Finance NLP Zero-shot NER



	Entity	Question
0	DATE	['When was the company acquisition?', 'When was the
1	ORG	['Which company was acquired?']
2	PRODUCT	['Which product?']
3	PROFIT_INCREASE	['How much has the gross profit increased?']
4	REVENUES_DECLINED	['How much has the revenues declined?']
5	OPERATING_LOSS_2020	['Which was the operating loss in 2020?']
6	OPERATING_LOSS_2019	['Which was the operating loss in 2019?']
7	EIN_NUMBER	['What is Employer Identification Number?']
8	NYSE_TICKER	['What is New York Stock Exchange Ticker Symbol?']

While our gross profit margin increased to
81.4% in 2020 from 63.1% in 2019, our
PROFIT_INCREASE
revenues declined approximately
27% in 2020 as compared to We reported an operating loss of approximately
\$8,048,581 million in 2020 as compared to an
OPERATING_LOSS_2020
operating loss of \$7,738,193 in 2019.
OPERATING_LOSS_2019 DATE

Usually, NLP models follow a **fit-transform** approach, where:

- 1) You **first** train a model, using what we call an Approach (*NerApproach* for NER), using training data.
- 2) And then, you **transform** (predict) on final data (*NerModel* for NER)

However, with the recent improvements in *Natural Language Inference*, we can use **Question Answering** models as well. The idea is quite simple:

- 1) You have a context document;
- 2) You have some *prompts* in form of *questions* or examples.

Using our **ZeroShotNER** annotator, those *questions (prompts)* can be asked to our NLI-based language model, and *retrieve the answers* in form of *predictions*, without a training step. And most importantly, **without any training data required**.

You can train in SparkNLP

- NerModel (Char CNNs - BiLSTM - CRF)
- ContextualParser (rule based)

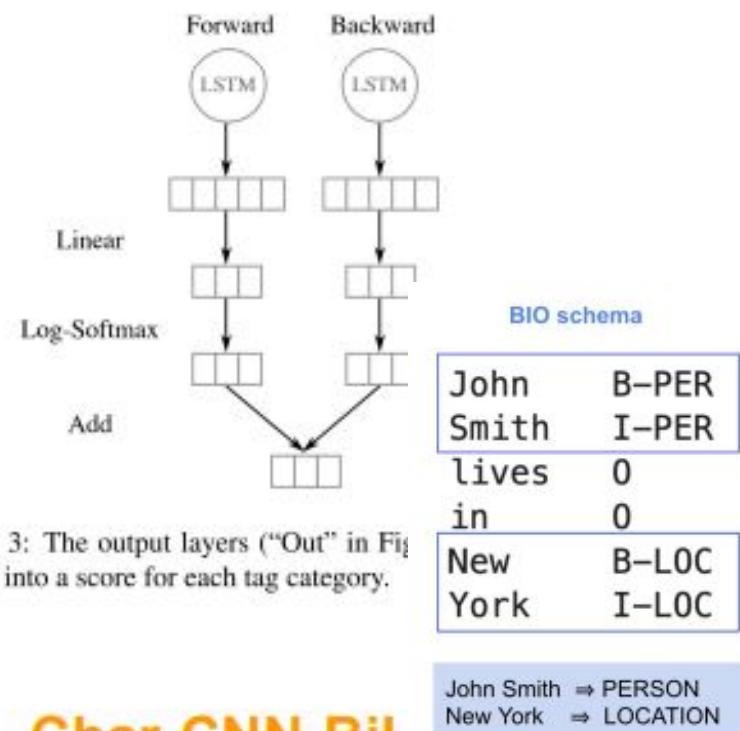
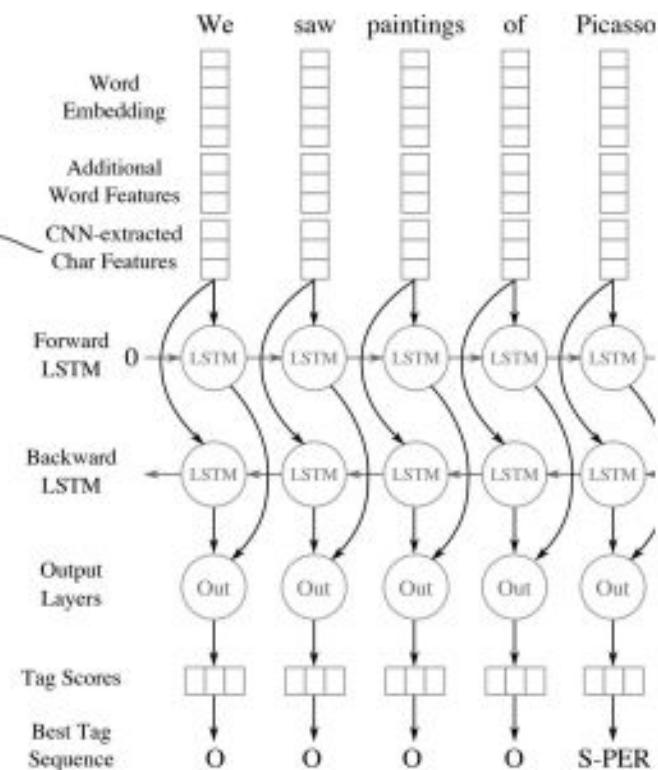
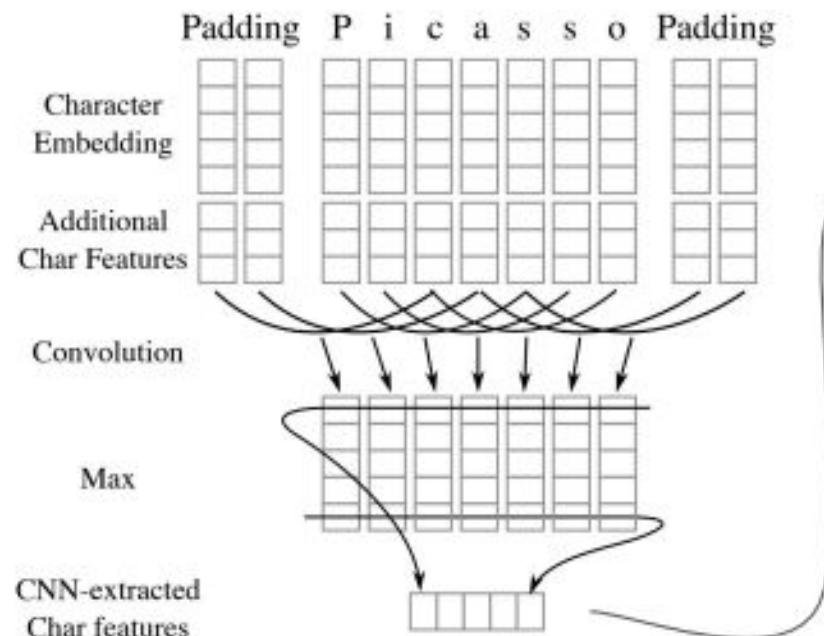
Actively used, we provide with templates to train in Hugging Face and import into Spark NLP

- BertForTokenClassification (transformer based)

Other available transformer-based

- RoBertaForTokenClassification
- CamemBertForTokenClassification
- DistilBertForTokenClassification
- LongformerForTokenClassification
- XlmRoBertaForTokenClassification
- XlnetForTokenClassification

NER-DL in Spark NLP



Char-CNN-BiL





STATE OF THE ART

**Relation Extraction
Finance NLP**

Welcome - We have a lot of excited things ahead of us

Duration

Module

Notebooks

50 min

Introduction. Text Splitting. Tokenization. Embeddings. Binary, Multiclass, Multilabel Classification

01.Page_Splitting.ipynb
02.Sentence_Splitting_Tokenization.ipynb
03.Word_Sentence_EMBEDDINGS.ipynb
04.0.Document_Paragraph_Classification.ipynb
04.2.Training_Financial_Multiclass_Classifier.ipynb

10 min

break

60 min

Named Entity Recognition. Relation Extraction. Training. Zero-shot NER and RE.

05.0.NER_and_ZeroShotNER.ipynb
05.1.Training_Financial_NER.ipynb
06.0.Relation_Extraction.ipynb
06.2.ZeroShot_Relation_Extraction.ipynb

10 min

break

50 min

Understand Entities in Context with Assertion Status.
Question&Answering Models. Data Normalization, Mapping, Augmentation

07.0.Understand_Entities_in_Context.ipynb
08.0.Answering_Questions_Financial_Texts.ipynb
09.0.Normalization_with_Entity_Resolution_Edgar.ipynb
10.0.Data_Augmentation_with_ChunkMappers.ipynb

10 min

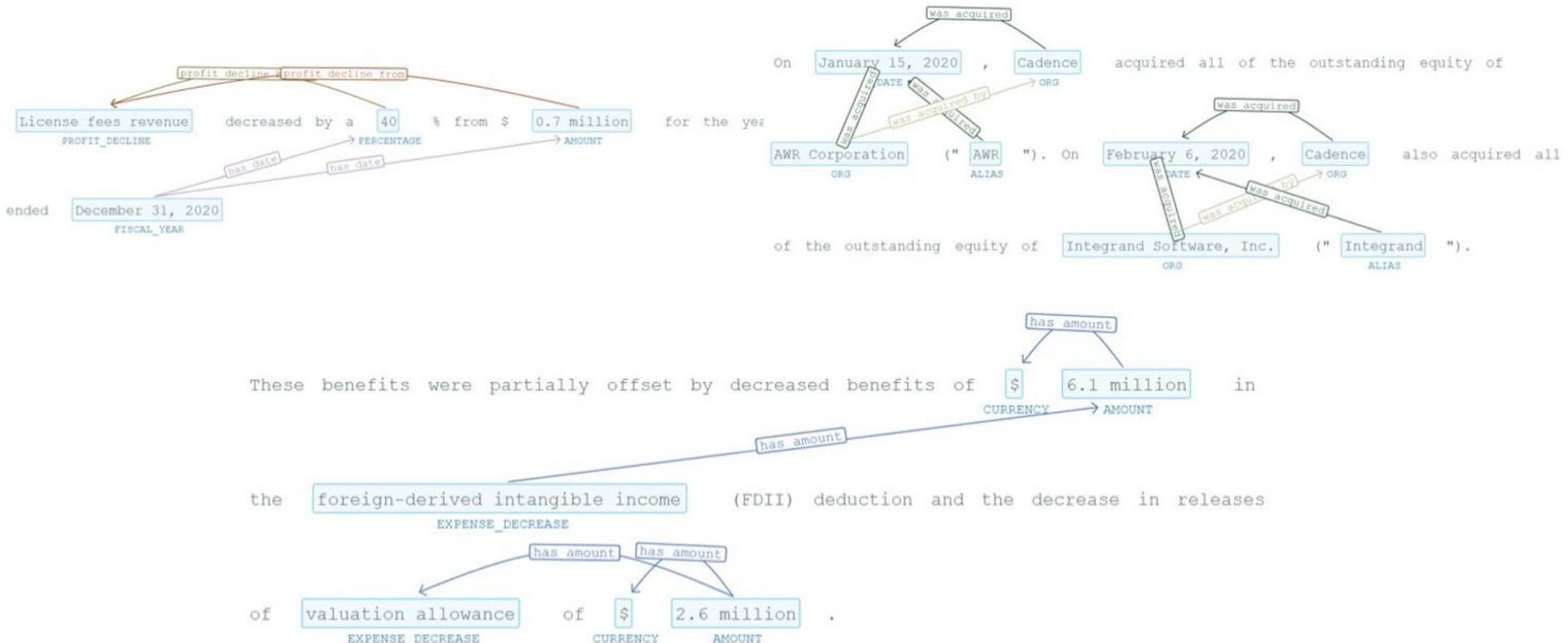
break

50 min

Financial Graph on a real use case. Deidentification. Integration with Visual NLP.

Databricks Solution Accelerator
11.0.Deidentification.ipynb
90.0.Financial_Visual_Classification.ipynb
90.1.Visual_and_Textual_Classification.ipynb

Understand relationships in Financial Documents



Relation Extraction

Relation Extraction is the NLP Task in charge of detecting if there is any relationship between 2 NER entities, and categorize them.

- Relation Extraction requires **Word embeddings**.
- As with Classification and NER, **Relation Extraction** also requires **splitting**. However, the main difference is that **entities may be in different sentences**, especially in Finance NLP, so it's recommended a bigger splitting than sentences. Usually **paragraph splitting** has good results, but you can also use **section** splitting.

Relation Extraction always goes after **Entity Recognition (NER)**, and tries to **categorize each pair of entities** retrieved in the same chunk,

What happens with texts with many entities?	What happens with long texts?
<p>RelationExtraction will try to understand if each combination of 2 entities is a category.</p> <p>This may be low performant or have undesired results, which you can prevent by:</p> <ul style="list-style-type: none">• Setting which combinations of entities may be checked.	<p>RelationExtraction will try to understand if each combination of 2 entities is a category.</p> <p>This may be low performant or have undesired results, which you can prevent by:</p> <ul style="list-style-type: none">• Set a maximum distance between entities.

Relation Extraction

```
....  
ner_model = finance.NerModel.pretrained("finner_org_per_role_date", "en", "finance/models")\  
    .setInputCols(["sentence", "token", "bert_embeddings"])\  
    .setOutputCol("ner_orgs")  
  
ner_converter = NerConverter()\  
    .setInputCols(["sentence", "token", "ner_orgs"])\  
    .setOutputCol("ner_chunk")  
  
pos = PerceptronModel.pretrained()\  
    .setInputCols(["sentence", "token"])\  
    .setOutputCol("pos")  
  
dependency_parser = DependencyParserModel().pretrained("dependency_conllu", "en")\  
    .setInputCols(["sentence", "pos", "token"])\  
    .setOutputCol("dependencies")  
  
re_filter = finance.RENerChunksFilter()\  
    .setInputCols(["ner_chunk", "dependencies"])\  
    .setOutputCol("re_ner_chunk")\  
    .setMaxSyntacticDistance(5)  
    .setRelationPairs(["PERSON-ROLE", "PERSON-ORG", "ORG-ROLE", "DATE-ROLE"])\  
....  
  
reDL = finance.RelationExtractionDLModel()\  
    .pretrained('finre_work_experience_md', 'en', 'finance/models')\  
    .setInputCols(["re_ner_chunk", "sentence"])\  
    .setOutputCol("relations")
```

For doing that, we have a helper annotator called **RENNerChunksFilter**.

You can use:

- **setMaxSyntacticDistance**, to restrict the maximum distance between 2 entities.
- **setRelationPairs**, to allow only certain combination of NER types.

These steps are optional, as you can see in some examples they will just be commented out. In other cases it will be crucial due to false positives or negatives.

Zero-shot Relation Extraction

As with Zero-shot NER, we can carry out zero-shot Relation Extraction, using the following prompt syntax:

```
re_model = finance.ZeroShotRelationExtractionModel.pretrained("finre_zero_shot", "en", "finance/models")\
    .setInputCols(["ner_chunk", "sentence"]) \
    .setOutputCol("relations")

# Remember it's 2 curly brackets instead of one if you are using Spark NLP < 4.0
re_model.setRelationalCategories({
    "DECREASE": ["{PROFIT_DECLINE} decrease {AMOUNT}", "{PROFIT_DECLINE} decrease {PERCENTAGE}"],
    "INCREASE": ["{PROFIT_INCREASE} increase {AMOUNT}", "{PROFIT_INCREASE} increase {PERCENTAGE}"]
})
```

setRelationalCategories requires a dictionary, having as keys the relationship name, and as values a list of possible prompts which model those relations. In **brackets** you need to put the entity names involved in the relation.



Annotators

- **RelationExtractionDL**: a span-bert (transformer-based) Relation Extraction model. We provide notebooks to train and import it into Spark NLP.
- **RelationExtraction**: a feature-based multilayer-perceptron model.
- **ZeroShotRE** Zero Shot Relation Extraction. No data required, just *prompts*.



STATE OF THE ART

**Assertion Status
Finance NLP**

Welcome - We have a lot of excited things ahead of us

Duration

Module

Notebooks

50 min

Introduction. Text Splitting. Tokenization. Embeddings. Binary, Multiclass, Multilabel Classification

01.Page_Splitting.ipynb
02.Sentence_Splitting_Tokenization.ipynb
03.Word_Sentence_EMBEDDINGS.ipynb
04.0.Document_Paragraph_Classification.ipynb
04.2.Training_Financial_Multiclass_Classifier.ipynb

10 min

break

60 min

Named Entity Recognition. Relation Extraction. Training. Zero-shot NER and RE.

05.0.NER_and_ZeroShotNER.ipynb
05.1.Training_Financial_NER.ipynb
06.0.Relation_Extraction.ipynb
06.2.ZeroShot_Relation_Extraction.ipynb

10 min

break

50 min

**Understand Entities in Context with Assertion Status.
Question&Answering Models. Data Normalization, Mapping, Augmentation**

07.0.Understand_Entities_in_Context.ipynb
08.0.Answering_Questions_Financial_Texts.ipynb
09.0.Normalization_with_Entity_Resolution_Edgar.ipynb
10.0.Data_Augmentation_with_ChunkMappers.ipynb

10 min

break

50 min

Financial Graph on a real use case. Deidentification. Integration with Visual NLP.

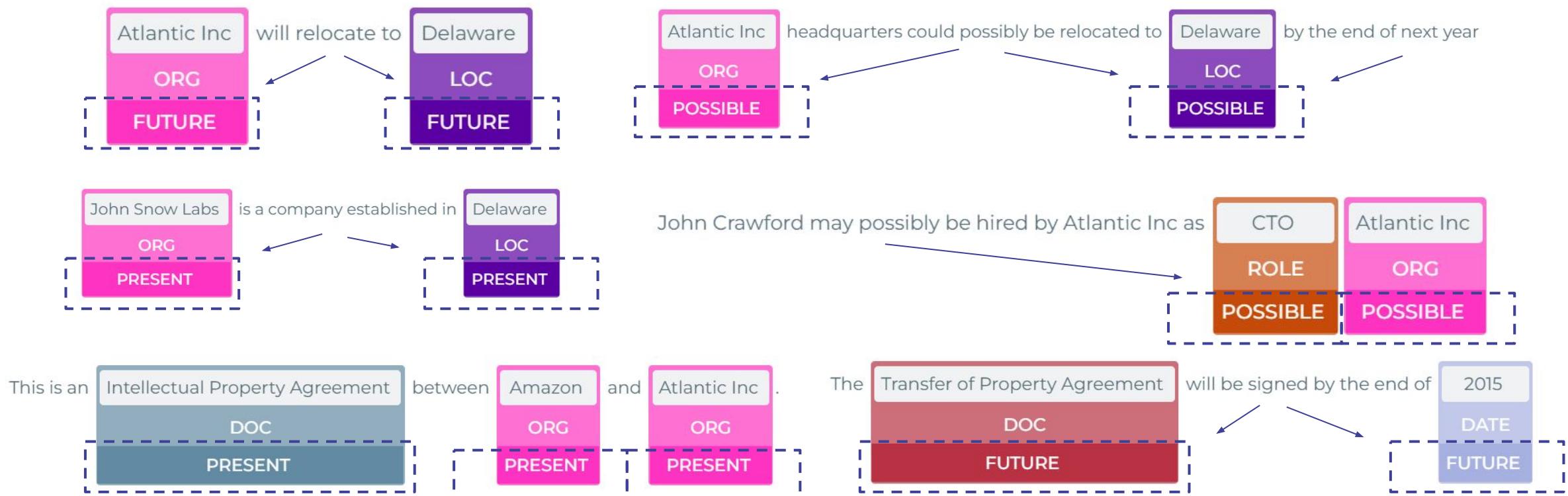
Databricks Solution Accelerator
11.0.Deidentification.ipynb
90.0.Financial_Visual_Classification.ipynb
90.1.Visual_and_Textual_Classification.ipynb

Understanding Entities in Context: Assertion Status

Assertion Status is the NLP Task in charge of **understanding entities in context**, and categorize them base on it. For example, it can detect if an entity is mentioned in a *Past*, *Future*, *Present* or *Possible* context.

- **Assertion Status** requires **Word embeddings**.
- **Assertion Status** also requires **splitting**. However, the main difference is that will need to decide if the context of the sentence of the entity is enough or you want to provide with more. That should be taken into account to decide either to go with **sentence splitting** or with **paragraph splitting**. Usually, sentence splitting should suffice.

Assertion Status always goes after **Entity Recognition (NER)**.



Understanding Entities in Context: Assertion Status

Much more than Negation, Temporality or Possibility.

Any NER entity which requires additional context to be sub-categorized, can be modelled with Assertion Status. For example:

- We extract **companies** as **ORG**;
- We analyse the context to understand if those **ORG** are mentioned to be my **competitors**;

In the customer management market, we compete with



and



A significant portion of our revenues in our Scores segment is attributable to the U.S. mortgage market, which includes, for conforming mortgages in that

market, a requirement of The



("Fannie Mae") and The



("Freddie

Mac") that U.S. lenders provide



for each mortgage delivered to them

Annotators

- **AssertionDL**: a multilayer perceptron model using embeddings..
- **AssertionLogReg**: a feature-based multilayer-perceptron model



STATE OF THE ART

**Entity Resolution
Finance NLP**

Welcome - We have a lot of excited things ahead of us

Duration

Module

Notebooks

50 min

Introduction. Text Splitting. Tokenization. Embeddings. Binary, Multiclass, Multilabel Classification

01.Page_Splitting.ipynb
02.Sentence_Splitting_Tokenization.ipynb
03.Word_Sentence_EMBEDDINGS.ipynb
04.0.Document_Paragraph_Classification.ipynb
04.2.Training_Financial_Multiclass_Classifier.ipynb

10 min

break

60 min

Named Entity Recognition. Relation Extraction. Training. Zero-shot NER and RE.

05.0.NER_and_ZeroShotNER.ipynb
05.1.Training_Financial_NER.ipynb
06.0.Relation_Extraction.ipynb
06.2.ZeroShot_Relation_Extraction.ipynb

10 min

break

50 min

**Understand Entities in Context with Assertion Status.
Question&Answering Models. Data Normalization, Mapping, Augmentation**

07.0.Understand_Entities_in_Context.ipynb
08.0.Answering_Questions_Financial_Texts.ipynb
09.0.Normalization_with_Entity_Resolution_Edgar.ipynb
10.0.Data_Augmentation_with_ChunkMappers.ipynb

10 min

break

50 min

Financial Graph on a real use case. Deidentification. Integration with Visual NLP.

Databricks Solution Accelerator
11.0.Deidentification.ipynb
90.0.Financial_Visual_Classification.ipynb
90.1.Visual_and_Textual_Classification.ipynb

Entity Resolution



Entity Resolution is the NLP Task in charge of, given an **NER chunk, retrieve the most semantically similar candidate** from a training set the model has been trained on. But it is much more than a *Text Similarity* task, **it can store unique IDs** so that, after the sentence similarity task, it retrieves not only the most similar **name, but also an ID**.

It requires **Sentence Embeddings**.

This has been widely used for retrieving **normalized versions** of, for example, **company names** (which can have many version as *INC, Inc., inc.*, different punctuation, etc) and their **unique ID**, as for example, their CIK in Edgar Database

Entity Resolution goes always after **Name Entity Recognition (NER)**.

About

Auxilium Pharmaceuticals (NASDAQ: AUXL) was founded in 1999 to develop and market pharmaceutical products that focus on ...

normalization and ID retrieval



Company name, ticker, CIK number or individual's name

AUXILIUM PHARMACEUTICALS INC (CIK 0001182129)

Entity resolution IS NOT a Deep Learning model, it carries out Semantic Search using a Language Model (embeddings).

Annotators

- **EntityResolver**: a multidimensional data structure storing embeddings and data associated to them, and able to be used at scale for semantic similarity operations..



STATE OF THE ART

Data Augmentation with Chunk Mappers Finance NLP

Welcome - We have a lot of excited things ahead of us

Duration

Module

Notebooks

50 min

Introduction. Text Splitting. Tokenization. Embeddings. Binary, Multiclass, Multilabel Classification

01.Page_Splitting.ipynb
02.Sentence_Splitting_Tokenization.ipynb
03.Word_Sentence_EMBEDDINGS.ipynb
04.0.Document_Paragraph_Classification.ipynb
04.2.Training_Financial_Multiclass_Classifier.ipynb

10 min

break

60 min

Named Entity Recognition. Relation Extraction. Training. Zero-shot NER and RE.

05.0.NER_and_ZeroShotNER.ipynb
05.1.Training_Financial_NER.ipynb
06.0.Relation_Extraction.ipynb
06.2.ZeroShot_Relation_Extraction.ipynb

10 min

break

50 min

**Understand Entities in Context with Assertion Status.
Question&Answering Models. Data Normalization, Mapping, Augmentation**

07.0.Understand_Entities_in_Context.ipynb
08.0.Answering_Questions_Financial_Texts.ipynb
09.0.Normalization_with_Entity_Resolution_Edgar.ipynb
10.0.Data_Augmentation_with_ChunkMappers.ipynb

10 min

break

50 min

Financial Graph on a real use case. Deidentification. Integration with Visual NLP.

Databricks Solution Accelerator
11.0.Deidentification.ipynb
90.0.Financial_Visual_Classification.ipynb
90.1.Visual_and_Textual_Classification.ipynb

Data augmentation with Chunk Mappers

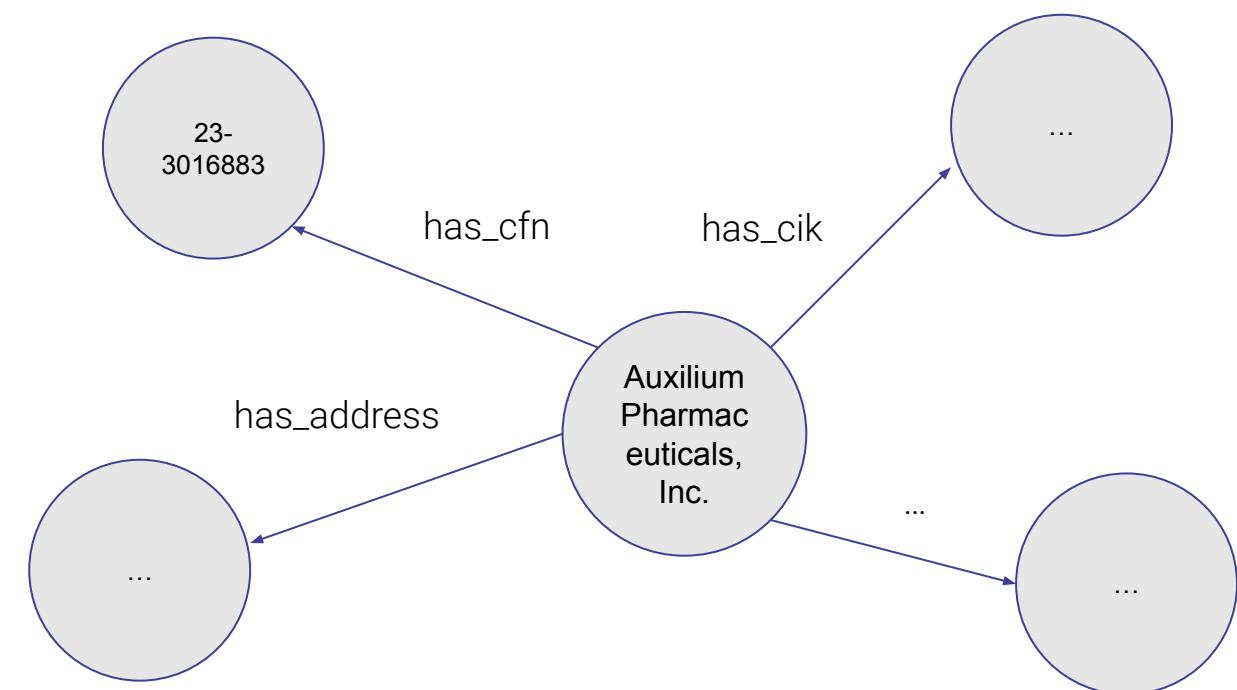
Given an **NER chunk extracted in NER**, and a **dictionary** in json format, you can use the NER chunks as a **key to retrieve the values from a dictionary** in form of relationships.

About

Example:

```
"mappings": [
  {
    "key": "Auxilium Pharmaceuticals.",
    "relations": [
      {
        "key": "has_cfn",
        "values" : ["23-3016883"]
      },
      ...
    ]
  }
]
```

Auxilium Pharmaceuticals (NASDAQ: AUXL) was founded in 1999 to develop and market pharmaceutical products that focus on ...



Data augmentation with Chunk Mappers

Given an **NER chunk extracted in NER, and a dictionary** in json format, you can use the NER chunks as a **key to retrieve the values from a dictionary** in form of relationships.

Chunk Mappers always go after **Entity Resolution**, because in your json file you should have unique keys. That means you should not save in a Chunk Mapper both *Auxilium Pharmaceuticals* and *Auxilium Pharmaceuticals Inc*, **you should only stored the normalized / official version** (AUXILIUM PHARMACEUTICALS INC, as per Edgar Database) in the json. And then, after NER, you carry out **normalization with Entity Resolvers**, and then Chunk Mapping to retrieve the rest of information.

Auxilium Pharmaceuticals (NASDAQ: AUXL) was founded in 1999 to develop and market pharmaceutical products that focus on ...

Normalization

AUXILIUM PHARMACEUTICALS INC

Augmentation

Company name, ticker, CIK number or individual's name

AUXILIUM PHARMACEUTICALS INC (CIK 0001182129)

crunchbase



AUXILIUM PHARMACEUTICALS INC

[+] Company Information

CIK:
1182129

State location:
PA

Business address:
640 LEE ROAD, CHESTERBROOK, PA, 19087
Phone: 404 321 5900

Filings:
1,257 EDGAR filings since August 19, 2002

EIN:
23-3016883

State of incorporation:
DE

Mailing address:
640 LEE ROAD, CHESTERBROOK, PA, 19087

SIC:
2834 - Pharmaceutical Preparations
(CF Office: Office of Life Sciences)

Fiscal year end:
December 31

Chunk Mapping IS NOT a Deep Learning model, it carries out scalable offline mapping in json structures.

Annotators

- **ChunkMapper:** a key-value data storage to be queried by keys and retrieve relations and values.



STATE OF THE ART

**Question & Answering
Finance NLP**

Welcome - We have a lot of excited things ahead of us

Duration

Module

Notebooks

50 min

Introduction. Text Splitting. Tokenization. Embeddings. Binary, Multiclass, Multilabel Classification

01.Page_Splitting.ipynb
02.Sentence_Splitting_Tokenization.ipynb
03.Word_Sentence_EMBEDDINGS.ipynb
04.0.Document_Paragraph_Classification.ipynb
04.2.Training_Financial_Multiclass_Classifier.ipynb

10 min

break

60 min

Named Entity Recognition. Relation Extraction. Training. Zero-shot NER and RE.

05.0.NER_and_ZeroShotNER.ipynb
05.1.Training_Financial_NER.ipynb
06.0.Relation_Extraction.ipynb
06.2.ZeroShot_Relation_Extraction.ipynb

10 min

break

50 min

**Understand Entities in Context with Assertion Status.
Question&Answering Models. Data Normalization, Mapping, Augmentation**

07.0.Understand_Entities_in_Context.ipynb
08.0.Answering_Questions_Financial_Texts.ipynb
09.0.Normalization_with_Entity_Resolution_Edgar.ipynb
10.0.Data_Augmentation_with_ChunkMappers.ipynb

10 min

break

50 min

Financial Graph on a real use case. Deidentification. Integration with Visual NLP.

Databricks Solution Accelerator
11.0.Deidentification.ipynb
90.0.Financial_Visual_Classification.ipynb
90.1.Visual_and_Textual_Classification.ipynb

Document Question Answering

Entity	Question
0 DATE	['When was the company acquisition?', 'When was the company purchase agreement?']
7 EIN_NUMBER	['What is Employer Identification Number?']
8 NYSE_TICKER	['What is New York Stock Exchange Ticker Symbol?']
6 OPERATING_LOSS_2019	['Which was the operating loss in 2019']
5 OPERATING_LOSS_2020	['Which was the operating loss in 2020']
1 ORG	['Which company was acquired?']
2 PRODUCT	['Which product?']
3 PROFIT_INCREASE	['How much has the gross profit increased?']

While our gross profit margin increased to **81.4%** in 2020 from 63.1% in 2019, our revenues declined approximately **27%** in 2020 as compared to 2019.

We reported an operating loss of approximately **\$8,048,581 million** in 2020 as compared to an operating loss of **\$7,738,193** in 2019.

2019 .
DATE

Entity	Question
0 DATE	['When was the company acquisition?', 'When was the company purchase agreement?']
1 ORG	['Which company was acquired?']
2 PRODUCT	['Which product?']
3 PROFIT_INCREASE	['How much has the gross profit increased?']
4 REVENUES_DECLINED	['How much has the revenues declined?']
5 OPERATING_LOSS_2020	['Which was the operating loss in 2020']
6 OPERATING_LOSS_2019	['Which was the operating loss in 2019']
7 EIN_NUMBER	['What is Employer Identification Number?']
8 NYSE_TICKER	['What is New York Stock Exchange Ticker Symbol?']

While our gross profit margin increased to **81.4%** in 2020 from 63.1% in 2019, our revenues declined approximately **27%** in 2020 as compared to 2019. We reported an operating loss of approximately **\$8,048,581 million** in 2020 as compared to an operating loss of **\$7,738,193** in 2019.

2019 .
DATE

Finance NLP Question Answering

Question Answering is the NLP Task in charge of, given a **question**, **retrieve an answer**. There are two main groups of QA models:

- **Open book**: We provide also with a context where to look.
- **Closed book**: The knowledge is stored in the Language Model and you don't give any example.

We use the *Open-book* approach, as **we want to retrieve answers in our specific documents**.

These models are **NLI**-based (*Natural Language Inference*). They use the question as a **hypotheses**, and try to find the maximum number of consequent tokens which maximize the probability to be an **answer** to that hypotheses.

Premise	Hypotheses	Inference Results
In 2017, the Company reported a profit decline of \$4 million dollars compared to 2016	The Company reported a profit decline in 2017.	Entailment
	The Company reported a profit increase in 2017.	Contradiction
	The Company is John Snow Labs, Inc.	Neutral

Entity	Question
DATE	['When was the company acquisition?', 'When was the
ORG	['Which company was acquired?']
PRODUCT	['Which product?']
PROFIT_INCREASE	['How much has the gross profit increased?']
REVENUES_DECLINED	['While our gross profit margin increased to
OPERATING_LOSS_2020	['V 81.4% in 2020 from 63.1% in 2019, our
OPERATING_LOSS_2019	['V PROFIT_INCREASE revenues declined approximately
EIN_NUMBER	['V 27% in 2020 as compared to 2019.
NYSE_TICKER	['V REVENUES_DECLINED

Finance NLP Question Answering for NER

Question Answering can be also used for retrieving specific NER entities you can't retrieve with other NER methods, either because your *model don't perform well, you don't have enough data to train, or any other reason.*

To do that, you can create your questions manually or even automatically generate questions if you have the SUBJECT and the VERB. To get them:

- Using **Part of Speech** and **Dependency Parser**;
- Using **ContextualParser** or **RegexMatcher**;
- If you have an NER model trained to detect subjects or objects, you can use **NerQuestionGeneration** to automatically generate those questions from you

```
+-----+  
| col |  
+-----+  
|{chunk, 4, 8, Buyer, {entity -> OBLIGATION SUBJECT, sentence -> 0, chunk -> 0, confidence -> 0.86514723}, []}|  
|{chunk, 10, 18, shall use, {entity -> OBLIGATION ACTION, sentence -> 0, chunk -> 1, confidence -> 0.9830627}, []}|  
+-----+
```

```
qagenerator = legal.NerQuestionGenerator()\\"  
.setInputCols(["ner_chunk"])\\"  
.setOutputCol("question")\\"  
.setQuestionMark(True)\\"  
.setQuestionPronoun("What")\\"  
.setEntities1(["OBLIGATION SUBJECT"])\\"  
.setEntities2(["OBLIGATION ACTION"])
```

```
+-----+  
| result |  
+-----+  
|[What Buyer shall use ?]|  
+-----+
```

The Buyer shall use such materials and supplies only in accordance with the present
agreement

Annotators

- **BertForQuestionAnswering:** NLI-based models to retrieve answers to questions in textual format using Bert.
- **RoBertaForQuestionAnswering:** NLI-based models to retrieve answers to questions in textual format using RoBerta.
- **DistilBertForQuestionAnswering:** NLI-based models to retrieve answers to questions in textual format using DistilBert.
- ...



STATE OF THE ART

Table Understanding Finance NLP

Welcome - We have a lot of excited things ahead of us

Duration

Module

Notebooks

50 min

Introduction. Text Splitting. Tokenization. Embeddings. Binary, Multiclass, Multilabel Classification

01.Page_Splitting.ipynb
02.Sentence_Splitting_Tokenization.ipynb
03.Word_Sentence_EMBEDDINGS.ipynb
04.0.Document_Paragraph_Classification.ipynb
04.2.Training_Financial_Multiclass_Classifier.ipynb

10 min

break

60 min

Named Entity Recognition. Relation Extraction. Training. Zero-shot NER and RE.

05.0.NER_and_ZeroShotNER.ipynb
05.1.Training_Financial_NER.ipynb
06.0.Relation_Extraction.ipynb
06.2.ZeroShot_Relation_Extraction.ipynb

10 min

break

50 min

**Understand Entities in Context with Assertion Status.
Question&Answering Models. Data Normalization, Mapping, Augmentation**

07.0.Understand_Entities_in_Context.ipynb
08.0.Answering_Questions_Financial_Texts.ipynb
09.0.Normalization_with_Entity_Resolution_Edgar.ipynb
10.0.Data_Augmentation_with_ChunkMappers.ipynb

10 min

break

50 min

Financial Graph on a real use case. Deidentification. Integration with Visual NLP.

Databricks Solution Accelerator
11.0.Deidentification.ipynb
90.0.Financial_Visual_Classification.ipynb
90.1.Visual_and_Textual_Classification.ipynb

Extracting Tables from Documents

Imagine you have a PDF or image document with **tables** and you want to extract information from it.

Any attempt to extract information from it using Visual NLP or any other tool for PDF Extraction or OCR will end up giving something like this:

Workforce38% 38%0102030405060
Workforce Management0%10%20%30%40%50%60%
Workforce53% 53%
Management42% 42%20% Sales Demographics (Global)
010203040
Workforce Management20% 20% Demographics* (U.S. Only)2021 Wc
Demographics (Global)
Jun 2020 Jun 2021
0102030405060
Gen Z Millennial Gen X Baby Boomer0%10%20%30%40%50%60%



PURCHASED ELECTRICITY CONSUMPTION 2019 – 2021

Where available, each site purchases electricity directly from independent electricity suppliers. Typically, these are large scale utility companies servicing thousands of clients across a national electricity grid. The energy we generate on site is not included in the purchased electricity consumption, but is included in the overall energy consumption. Where independent power is not available, our sites generate their own power.

In 2021, purchased electricity consumption decreased marginally. As with previous years, 80% of our purchased group electricity is generated by renewable energy.

PURCHASED GROUP ELECTRICITY CONSUMPTION (TJ)

	2019	2020	2021
Hydro	8 438	8 058	8 074
Coal	486	1306	1286
Nuclear	246	251	171
Oil	394	336	428
Natural Gas	246	321	213
Wind	187	223	189
Biofuels & Waste	82	60	52
Solar	44	61	57
Geothermal	4	6	6
Other	0	2	2

So instead do **Table Detection and Extraction** with tools as Visual NLP to get the dataframe / CSV representation of it.

Patch	ACES RGB			Display RGB			Display xyY		
	N1	1.8233	1.8233	1.8233	0.9243	0.8651	0.9013	0.3217	0.3377
N2	0.2753	0.2753	0.2753	0.5383	0.5038	0.5249	0.3217	0.3377	8.4552
N3	0.0898	0.0898	0.0898	0.2804	0.2625	0.2734	0.3217	0.3377	1.5514
R	0.4689	0.1193	0.0417	0.8046	0.2227	0.1795	0.6413	0.3307	6.4488
G	0.339	0.8088	0.0936	0.4335	0.8036	0.2434	0.3046	0.624	20.8422
B	0.2162	0.1133	0.8711	0.1707	0.1503	0.8215	0.1562	0.0692	2.3365
C	0.5187	0.9138	1.0432	0.4332	0.8028	0.8406	0.2269	0.3404	22.8164
M	0.58	0.2096	0.9086	0.808	0.2134	0.8294	0.333	0.1596	8.4349
Y	0.8237	0.9378	0.0855	0.8654	0.8096	0.2487	0.4338	0.5187	26.9923

Now, you can use Finance NLP Table Understanding, which already takes the information from the headers (columns) and the cells, keeping layout information and the whole meaning of the information of the table!

Finance NLP Table Question Answering

We include specific **Table Understanding Annotators**, based on the **Tapas Transformers**, to ask questions not to textual documents, but **to tables loaded into dataframes**.

name	money	age
Donald Trump	\$100,000,000	75
Elon Musk	\$20,000,000,000,000	55

```
queries = [
    "Who earns less than 200,000,000?",
    "Who earns 100,000,000?",
    "How much money has Donald Trump?",
    "How old are they?", ]
```

```
+-----+
|answer
+-----+
|Donald Trump, {question -> Who earns less than 200,000,000?, aggregation -> NONE, cell_positions -> [0, 0], cell_scores -> 0.9999999}
|Donald Trump, {question -> Who earns 100,000,000?, aggregation -> NONE, cell_positions -> [0, 0], cell_scores -> 0.9999999}
|$100,000,000, {question -> How much money has Donald Trump?, aggregation -> NONE, cell_positions -> [1, 0], cell_scores -> 0.9999998}
|AVERAGE > 75, 55, {question -> How old are they?, aggregation -> AVERAGE, cell_positions -> [2, 0], [2, 1], cell_scores -> 0.99999976, 0.9999995}
+-----+
```

Finance NLP Table Question Answering

We include specific **Table Understanding Annotators**, based on the **Tapas Transformers**, to ask questions not to textual documents, but **to tables loaded into dataframes**.

If your table is not digital, but it's in a scanned image, you can use **Visual NLP** to extract and save it into a dataframe, and then load it to do Table QA.



1. Table detection
Visual NLP

	Swimmer	Hopper	Walker
State space dim.	10	12	20
Control space dim.	2	3	6
Total num. policy params	364	4800	8206
Sim. steps per iter.	50K	1M	1M
Policy (pi)	0.01	0.01	0.01
Stepsize (\bar{D}_{KL})	0.01	0.01	0.01
Hidden layer size	30	50	50
Discount (γ)	0.99	0.99	0.99
Vine: rollout length	50	100	100
Vine: rollout per state	4	4	4
Vine: Q-values per batch	500	2500	2500
Vine: max. rollouts for sampling	16	16	16
Vine: len. rollouts for sampling	1000	1000	1000
Vine: computation time (minutes)	2	14	40

2. Table extraction
Visual NLP

	Swimmer	Hopper	Walker
State Space dim.	10	12	20
Control Space dim.	2	3	6
Total num. policy params	364	4800	8206
Sim. steps per trial	50K	1M	1M
Policy (pi)	0.01	0.01	0.01
Stepsize (SGD)	0.01	0.01	0.01
Hidden layer size	30	50	50
Discount (γ)	0.99	0.99	0.99
Vine: rollout length	50	100	100
Vine: rollout per state	4	4	4
Vine: Q-values per batch	500	2500	2500

3. Save / Load as **Spark** Dataframe

```
queries = [
    "Who earns less than 200,000,000?", 
    "Who earns 100,000,000?", 
    "How much money has Donald Trump?", 
    "How old are they?", 
]
```

4. Table Understanding
Finance NLP

```
+-----+  
|answer  
+-----+  
|Donald Trump, {question -> Who earns less than 200,000,000?, aggregation -> NONE, cell_positions -> [0, 0], cell_scores -> 0.9999999}  
|Donald Trump, {question -> Who earns 100,000,000?, aggregation -> NONE, cell_positions -> [0, 0], cell_scores -> 0.9999999}  
|$100,000,000, {question -> How much money has Donald Trump?, aggregation -> NONE, cell_positions -> [1, 0], cell_scores -> 0.9999998}  
|AVERAGE > 75, 55, {question -> How old are they?, aggregation -> AVERAGE, cell_positions -> [2, 0], [2, 1], cell_scores -> 0.99999976, 0.9999995} |  
+-----+
```

Annotators

- **TableQuestionAnswering**: NLI-based models to retrieve answers to questions from tables using TAPAS.



STATE OF THE ART

Deidentification Finance NLP

Welcome - We have a lot of excited things ahead of us

Duration

Module

Notebooks

50 min

Introduction. Text Splitting. Tokenization. Embeddings. Binary, Multiclass, Multilabel Classification

01.Page_Splitting.ipynb
02.Sentence_Splitting_Tokenization.ipynb
03.Word_Sentence_EMBEDDINGS.ipynb
04.0.Document_Paragraph_Classification.ipynb
04.2.Training_Financial_Multiclass_Classifier.ipynb

10 min

break

60 min

Named Entity Recognition. Relation Extraction. Training. Zero-shot NER and RE.

05.0.NER_and_ZeroShotNER.ipynb
05.1.Training_Financial_NER.ipynb
06.0.Relation_Extraction.ipynb
06.2.ZeroShot_Relation_Extraction.ipynb

10 min

break

50 min

Understand Entities in Context with Assertion Status.
Question&Answering Models. Data Normalization, Mapping, Augmentation

07.0.Understand_Entities_in_Context.ipynb
08.0.Answering_Questions_Financial_Texts.ipynb
09.0.Normalization_with_Entity_Resolution_Edgar.ipynb
10.0.Data_Augmentation_with_ChunkMappers.ipynb

10 min

break

50 min

Financial Graph on a real use case. Deidentification. Integration with Visual NLP.

Databricks Solution Accelerator
11.0.Deidentification.ipynb
90.0.Financial_Visual_Classification.ipynb
90.1.Visual_and_Textual_Classification.ipynb

De-Identification



DATE: 2020-02-01 10:00:00 AM

APPLICATION NUMBER: 1234567

26 Mar 2022
Hi Ritwik, your HDFC Bank Account 00017 has been credited with INR 300,000 on March 26: Info:IRM*USD4000@74.75 . The Available Balance is INR 3,39,000

Detect sensitive entities



DATE: 2020-02-01 10:00:00 AM

APPLICATION NUMBER: 1234567

26 Mar 2022
Hi Ritwik, your **HDFC Bank** Account 00017 has been credited with INR 300,000 on **March 26**: Info:IRM*USD4000@74.75 . The Available Balance is INR 3,39,000

Transform sensitive entities



DATE: 2020-02-01 10:00:00 AM

APPLICATION NUMBER: 1234567

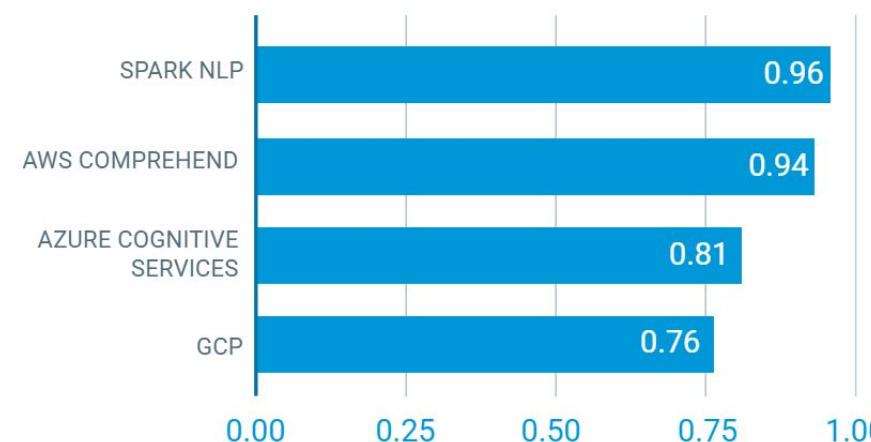
6 May 2021

Hi Ron, your **Bank of America** Account 03414 has been credited with INR 15,553 on **May, 1st**: Info:IRM*USD1241@114.75 . The Available Balance is INR 3,39,000

Transaction 1 on 26th Mar.

Transaction 1 on **26th Mar**

Transaction 1 on **26th Mar.**



Finance NLP Deidentification



Deidentification is the NLP task in charge of:

- 1) **Masking NER chunks or Obfuscating (faking) with synthetic data;**
- 2) **Returning an anonymized version** of the text;

It works on top of **NER** and **ContextualParser**, with specific **Deidentification** annotators which retrieve the NER chunks and mask / obfuscate them, all along with some other capabilities as *Language*, *Masking Technique*, *Date shift selection*, etc.

	Sentence	Masked	Masked with Chars	Masked with Fixed Chars	Obfuscated
0	CARGILL, INCORPORATED		[*****]	****	TURER INC
1	By: Pirkko Suominen	By:	By: [*****]	By: ****	By: SESA CO.
2	Name: Pirkko Suominen Title: Director, Bio Technology Development Center, Date: 10/19/2011	Name: : Center, Date:	Name: [*****]; [*****] Center, Date: [*****]	Name: ****: **** Center, Date: ****	Name: John Snow Labs Inc: Sales Manager Center, Date: 03/08/2025
3	BIOAMBER, SAS	,	[*****], [*]	****, ****	Clarus Ilc., SESA CO.
4	By: Jean-François Huc	By:	By: [*****]	By: ****	By: JAMES TURNER
5	Name: Jean-François Huc Title: President Date: October 15, 2011\n\nemail : jeanfran@gmail.com...	Name: : Date:\n\nemail :\n\ncphone : 0	Name: [*****]: [*****]Date: [*****]\n\nemail : [*****]\n\ncphone : ...	Name: ****: ****Date: ****\n\nemail :\n\ncphone : ****0	Name: MGT Trust Company, LLC: Business ManagerDate: 11/7/2016\n\nemail : Berneta@hotmail.com)\n...

Annotators

- **Deidentification:** Model that retrieves NER entities and masks, obfuscates them with default, custom vocabulary, applying date shifting and other consistency criteria.



STATE OF THE ART

Graph Creation
Finance NLP

Welcome - We have a lot of excited things ahead of us

Duration

Module

Notebooks

50 min

Introduction. Text Splitting. Tokenization. Embeddings. Binary, Multiclass, Multilabel Classification

01.Page_Splitting.ipynb
02.Sentence_Splitting_Tokenization.ipynb
03.Word_Sentence_EMBEDDINGS.ipynb
04.0.Document_Paragraph_Classification.ipynb
04.2.Training_Financial_Multiclass_Classifier.ipynb

10 min

break

60 min

Named Entity Recognition. Relation Extraction. Training. Zero-shot NER and RE.

05.0.NER_and_ZeroShotNER.ipynb
05.1.Training_Financial_NER.ipynb
06.0.Relation_Extraction.ipynb
06.2.ZeroShot_Relation_Extraction.ipynb

10 min

break

50 min

Understand Entities in Context with Assertion Status.
Question&Answering Models. Data Normalization, Mapping, Augmentation

07.0.Understand_Entities_in_Context.ipynb
08.0.Answering_Questions_Financial_Texts.ipynb
09.0.Normalization_with_Entity_Resolution_Edgar.ipynb
10.0.Data_Augmentation_with_ChunkMappers.ipynb

10 min

break

50 min

Financial Graph on a real use case. Deidentification. Integration with Visual NLP.

databricks_solutions_accelerators/company_ecosystem_graph/
11.0.Deidentification.ipynb
90.0.Financial_Visual_Classification.ipynb
90.1.Visual_and_Textual_Classification.ipynb

UNITED STATES SECURITIES AND EXCHANGE COMMISSION

Washington, D.C. 20549

FORM 10-K

(Mark One)

ANNUAL REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934

For the fiscal year ended January 1, 2022

OR

TRANSITION REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934

For the transition period from _____ to _____.

Commission file number 000-15867

cadence®

CADENCE DESIGN SYSTEMS, INC.

(Exact name of registrant as specified in its charter)

Delaware

(State or Other Jurisdiction of Incorporation or Organization)

00-0000000

(I.R.S. Employer Identification No.)

2655 Seely Avenue, Building 5, San Jose, California

95134

(Address of Principal Executive Offices)

(Zip Code)

(408)-943-1234

(Registrant's Telephone Number, including Area Code)

Securities registered pursuant to Section 12(b) of the Act:

Title of Each Class

Trading Symbol(s)

Names of Each Exchange on which Registe

Common Stock, \$0.01 par value per share

CDNS

Nasdaq Global Select Market

Securities registered pursuant to Section 12(g) of the Act:

None

Indicate by check mark if the registrant is a well-known seasoned issuer, as defined in Rule 405 of the Securities Act. Yes No

Indicate by check mark if the registrant is not required to file reports pursuant to Section 13 or Section 15(d) of the Act. Yes No

Indicate by check mark whether the registrant (1) has filed all reports required to be filed by Section 13 or 15(d) of the Securities Exchange Act of 1934 during the preceding 12 months (or for such shorter period that the registrant was required to file such reports), (2) has been subject to such filing requirements for the past 90 days. Yes No

Indicate by check mark whether the registrant has submitted electronically every Interactive Data File required to be submitted pursuant to Rule 405 of Regulation S-T (§ 232.405 of this chapter) during the preceding 12 months (or for such shorter period that the registrant was required to submit such files). Yes No

Indicate by check mark whether the registrant is a large accelerated filer, an accelerated filer, a non-accelerated filer, a smaller reporting company, or an emerging growth company. See the definitions of "large accelerated filer," "accelerated filer," "smaller reporting company," and "emerging growth company" in Rule 12b-2 of the Exchange Act.

Large Accelerated Filer

Accelerated Filer

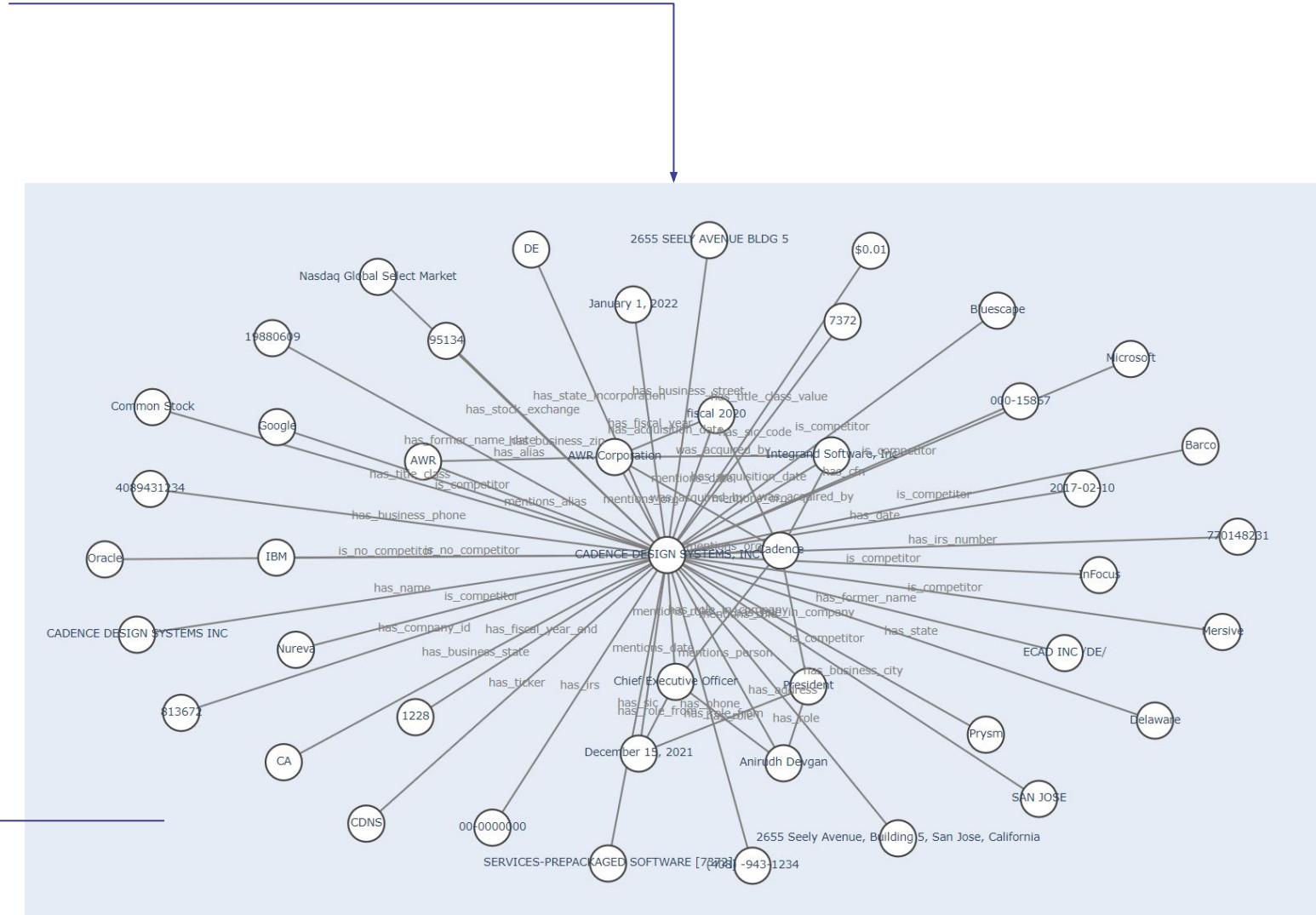
Non-accelerated Filer

Smaller Reporting Company

Emerging Growth Company

Graph Embeddings for company similarity, link prediction, etc?

- + Document splitting
 - + Paragraph Classification
 - + Name Entity Recognition on selected paragraphs
 - + Normalization and Data Augmentation
 - + Relation Extraction on Acquisitions, Subsidiaries, C-level managers, etc
 - + Assertion Status for Competitors vs No Competitors
 - + Temporality





STATE OF THE ART

Finance NLP and Visual NLP

Welcome - We have a lot of excited things ahead of us

Duration

Module

Notebooks

50 min

Introduction. Text Splitting. Tokenization. Embeddings. Binary, Multiclass, Multilabel Classification

01.Page_Splitting.ipynb
02.Sentence_Splitting_Tokenization.ipynb
03.Word_Sentence_EMBEDDINGS.ipynb
04.0.Document_Paragraph_Classification.ipynb
04.2.Training_Financial_Multiclass_Classifier.ipynb

10 min

break

60 min

Named Entity Recognition. Relation Extraction. Training. Zero-shot NER and RE.

05.0.NER_and_ZeroShotNER.ipynb
05.1.Training_Financial_NER.ipynb
06.0.Relation_Extraction.ipynb
06.2.ZeroShot_Relation_Extraction.ipynb

10 min

break

50 min

Understand Entities in Context with Assertion Status.
Question&Answering Models. Data Normalization, Mapping, Augmentation

07.0.Understand_Entities_in_Context.ipynb
08.0.Answering_Questions_Financial_Texts.ipynb
09.0.Normalization_with_Entity_Resolution_Edgar.ipynb
10.0.Data_Augmentation_with_ChunkMappers.ipynb

10 min

break

50 min

Financial Graph on a real use case. Deidentification. Integration with Visual NLP.

Databricks Solution Accelerator
11.0.Deidentification.ipynb
90.0.Financial_Visual_Classification.ipynb
90.1.Visual_and_Textual_Classification.ipynb

Visual classification



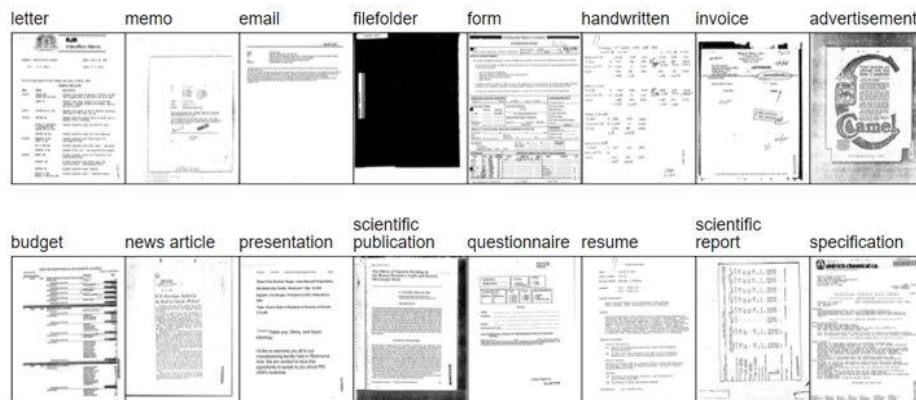
Sometimes textual information is not enough to classify a document. For example, let's suppose you have to classify 2 types of document with the same content, but only differing in the layout disposition of the information.

If we just get the text from them, and the contents are the same, Finance NLP may get confused. For this, we have 2 ways to go:

Finance NLP with Vision Transformers Image Level

Don't use text at all. Use **Visual Transformers** (**ViT models**) to transform only at image-level.

Characters consist of pixels, so they will be taken into account. Not a **language-level**, but a **pixel-level**.



Inconvenient: If you need the text to do NLP afterwards, maybe it's quicker to use the previous approach

Visual classification



Sometimes textual information is not enough to classify a document. For example, let's suppose you have to classify 2 types of document with the same content, but only differing in the layout disposition of the information.

If we just get the text from them, and the contents are the same, Finance NLP may get confused. For this, we have 2 ways to go:

Visual NLP
Text + Layout (HOCR)

Get from your documents from **images** or **pdf** both **textual and layout information** and use **Visual NLP** classifiers to predict the category based on them.

The diagram shows a document snippet with two versions of the word 'Gaitskell' highlighted: one in red box with a red dotted line to the first version in the image, and one in green box with a green dotted line to the second version in the image. Below the image is the corresponding HOCR XML code. A legend on the right identifies the colors: red for machine-printed printed text, green for machine-printed handwritten text, and blue for handwritten text.

A MOVE to stop Mr. **Gaitskell** from nominating any more Labour life Peers is to be made at the meeting of Labour M Ps tomorrow. Mr. Michael Foot has put down a resolution on the subject and he is to be backed by Mr. Will Griffiths, M P for Manchester Exchange.

Gaitskell

A MOVE to stop Mr. **Gaitskell** from

Gaitskell

Machine-Printed recognition of **printed text**

Machine-Printed recognition of **handwritten text**

```
<span id='line_1_4' title="bbox 400 376 2168 417">
    :
    <span id='word_1_10' title="bbox 791 378 865 408">Mr. </span>
    <span id='word_1_11' title="bbox 902 376 1085 408">Gaitskell</span>
    <span id='word_1_12' title="bbox 1101 377 1200 408">From</span>
    :
    </span>
    <span id='line_1_9' title="bbox 420 752 2084 841">
        :
        <span id='word_1_62' title="bbox 1199 760 1326 823">Mr. </span>
        <span id='word_1_63' title="bbox 1451 752 1835 829"><604&#39;an</span>
        <span id='word_1_64' title="bbox 1910 767 2084 840">onm</span>
        :
        </span>
```

Inconvenient: This approach uses OCR and tables, handwritten text, images, etc. may be ignored

10
4

Visual NLP: Summary



Document Classification

Classified Image

Classification

This document has been classified as: **Form**
Classification Confidence: **99.6%**

From images, pdf, docx, ppt...

... to plain text

UNITED STATES
SECURITIES AND EXCHANGE COMMISSION
Washington, D.C. 20549
FORM 10-K
 ANNUAL REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE
SECURITIES EXCHANGE ACT OF 1934
For the fiscal year ended October 31, 2021

TRANSITION REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE
SECURITIES EXCHANGE ACT OF 1934
For the transition period from _____ to _____

Commission File Number: 1-2402
HORMEL FOODS CORPORATION
(Exact name of registrant as specified in its charter)

Extracted Tables:

	Swimmer	Hopper	Walker
State space dim.	10	12	20
Control space dim.	2	3	6
Total num. policy params	364	4806	8206
Sim. steps per iter.	50K	1M	1M
Policy iters.	200	200	200
Stepsize (\bar{D}_{KL})	0.01	0.01	0.01
Hidden layer size	30	50	50
Discount (γ)	0.99	0.99	0.99
Vine: rollout length	50	100	100
Vine: rollouts per state	4	4	4
Vine: Q-values per batch	500	2500	2500
Vine: num. rollouts for sampling	16	16	16
Vine: len. rollouts for sampling	1000	1000	1000
Vine: computation time (minutes)	2	14	40

	Swimmer	Hopper	Walker
State space dim.	10	12	20
Control space dim.	2	3	6
Total num. policy params	364	4806	8206
Sim. steps per iter.	50K	1M	1M
Policy iters.	200	200	200
Stepsize (\bar{D}_{KL})	0.01	0.01	0.01
Hidden layer size	30	50	50
Discount (γ)	0.99	0.99	0.99
Vine: rollout length	50	100	100
Vine: rollouts per state	4	4	4
Vine: Q-values per batch	500	2500	2500
Vine: num. rollouts for sampling	16	16	16
Vine: len. rollouts for sampling	1000	1000	1000
Vine: computation time (minutes)	2	14	40

Table #1

Table, Signature extraction



Thank you!