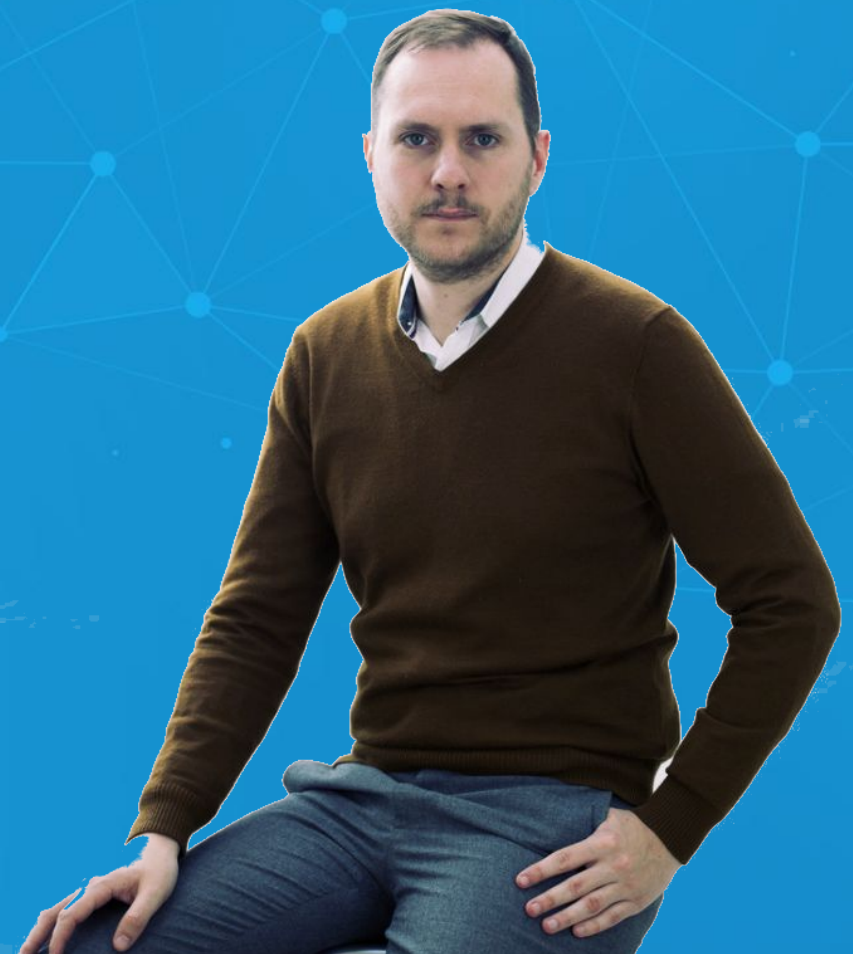
John Snow LABS

Legal NLP

Certification Trainings

Jan 25, 2023



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

Welcome - We have a lot of things ahead of us

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.Clause_Document_Classification.ipynb 04.1.Training_Legal_Binary_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 05.2.Clause_based_NER.ipynb 06.0.Relation_Extraction.ipynb
10 min	break	
50 min	Understand Entities in Context with Assertion Status. Question&Answering Models. Data Normalization, Mapping, Augmentation	06.1.Relation_Extraction_and_ZeroShotRE.ipynb 07.0.Understand_Entities_in_Context.ipynb 08.0.Answering_Questions_Legal_Texts.ipynb 09.0.Normalization_with_Entity_Resolution_Edgar.ipynb 10.0.Data_Augmentation_with_ChunkMappers.ipynb
10 min	break	
50 min	Deidentification. Financial Graph on a real use case. Integration with Visual NLP.	11.Deidentification.ipynb 80.0.WIP_Use_Case_Legal_Agreements.ipynb 90.0.Legal_Visual_Document_Understanding.ipynb 90.1.Layout_Classification_with_VisualNLP.ipynb

https://github.com/JohnSnowLabs/spark-nlp-workshop/tree/master/tutorials/Certification_Trainings_JSL/Legal

John Snow LABS

STATE OF THE ART

Introduction

Reviewing Spark NLP

John Snow Labs in 2022

John Snow LABS

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

Optimized, Tested, Supported Integrations

John Snow LABS

Spark NLP

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

Entity Recognition

I love Lucy PERSON

Information Extraction

They met Last week DATE -> 29-04-2020

Spelling & Grammar

She become the first... ->
She became the first

Text Classification

Translation

[je t'aime -> **i love you**]

Summarization

Question Answering

Q&A

Emotion Detection

Split Text

- Sentence Detector
- Tokenizer
- Normalizer
- nGram Generator
- Word Segmentation

Clean Text

- Spell Checker
- Grammar Checker
- Writing Style Checker
- Stopword Cleaner
- Summarization

Understand Grammar

- Stemmer
- Lemmatizer
- Part of Speech Tagger
- Dependency Parser
- Translation

Find in Text

- Text Matcher
- Regex Matcher
- Date Matcher
- Chunker
- Question Answering

10,000+

Pre-trained Pipelines, Models & Transformers

BERT | ELMO | GloVe
ALBERT | DeBERTa | USE
Longformer | ELECTRA
T5 | NMT | LaBSE
DistilBERT | RoBERTa
XLM-RoBERTa
S-BERT | XLNet

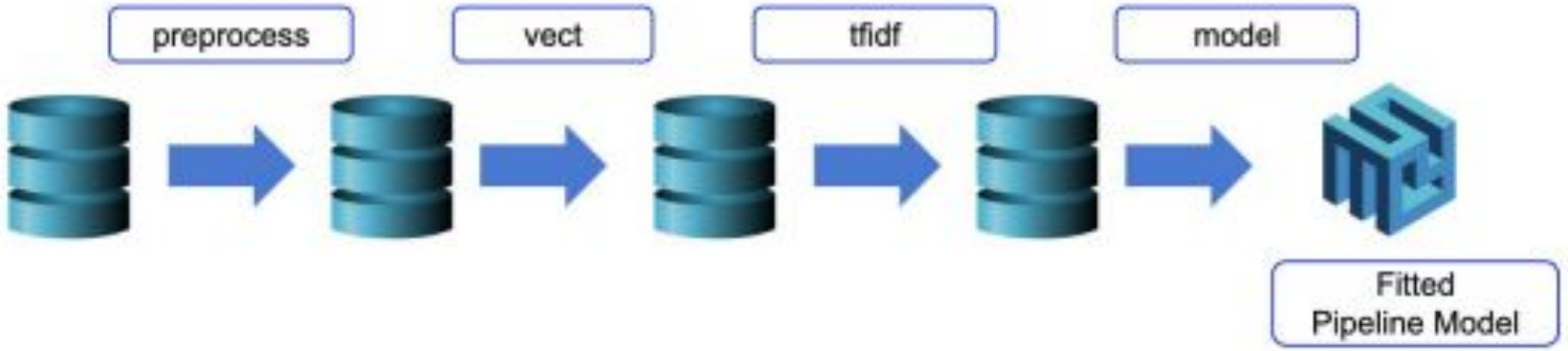
250+

Languages

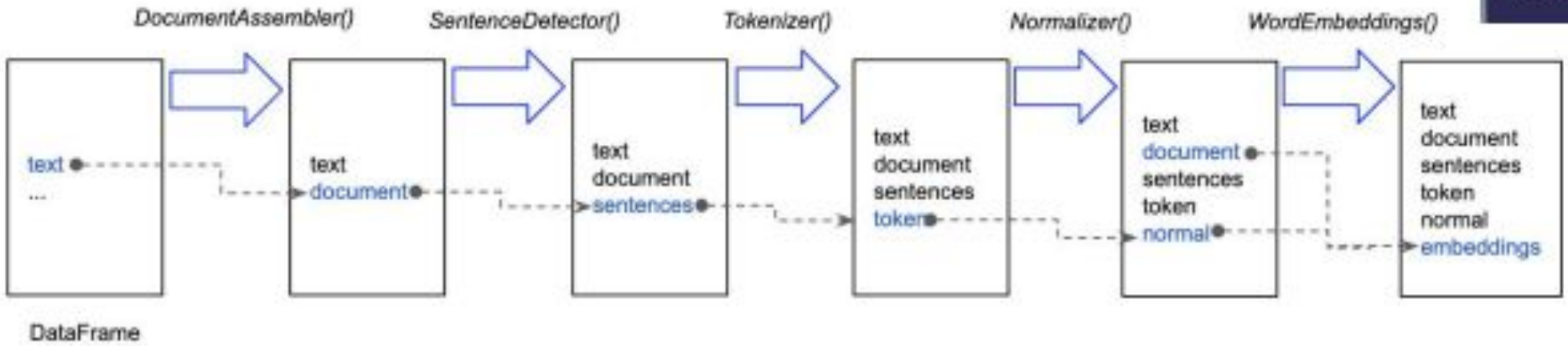
Trainable & Tunable	Scalable to a Cluster	Fast Inference	Hardware Optimized	Community
	Apache Spark ML Pipelines	LightPipeline	intel NVIDIA	NLP SUMMIT

Introducing Spark NLP

Pipeline of annotators




```
from pyspark.ml import Pipeline

document_assembler = DocumentAssembler()\
 .setInputCol("text")\
 .setOutputCol("document")

sentenceDetector = SentenceDetector()\
 .setInputCols(["document"])\
 .setOutputCol("sentences")

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

normalizer = Normalizer()\
 .setInputCols(["token"])\
 .setOutputCol("normal")

word_embeddings=WordEmbeddingsModel.pretrained()\
 .setInputCols(["document","normal"])\
 .setOutputCol("embeddings")

nlpPipeline = Pipeline(stages=[
 document_assembler,
 sentenceDetector,
 tokenizer,
 normalizer,
 word_embeddings,
 ])

nlpPipeline.fit(df).transform(df)
```



John Snow LABS

Spark NLP can be run both at **cluster level**, leveraging all the nodes, and a **master-only evel**, 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 MLib **Pipelines** and ***fit/transform***.

```
text ='''December 2007 SUBORDINATED LOAN AGREEMENT. THI

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

result = model.transform(df)
```

Result:

Spark Dataframe

scalable to millions of documents, slow for very few documents

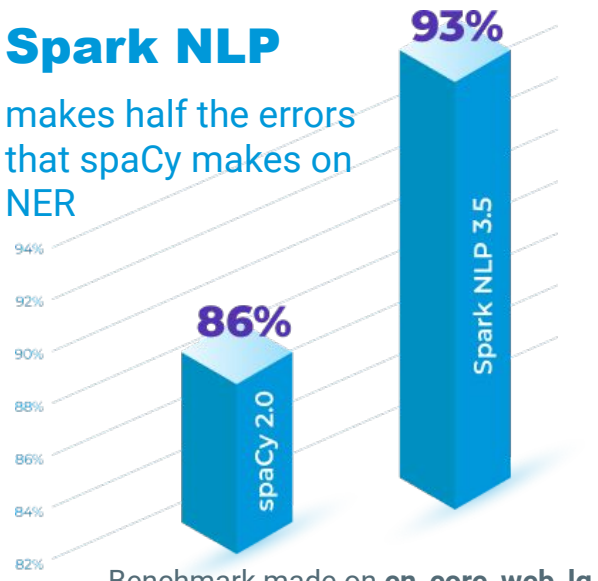
Trainable & Tunable	Scalable to a Cluster	Fast Inference	Hardware Optimized	Community
	Spark ML Pipelines	LightPipeline	intel NVIDIA	NLP SUMMIT

Spark NLP

is natively scalable and production-ready

Spark NLP

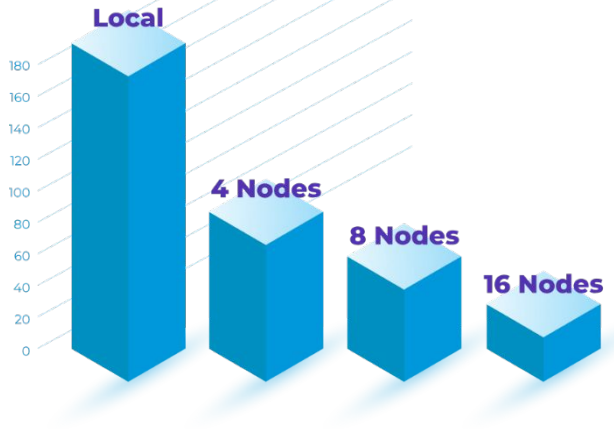
makes half the errors that spaCy makes on NER



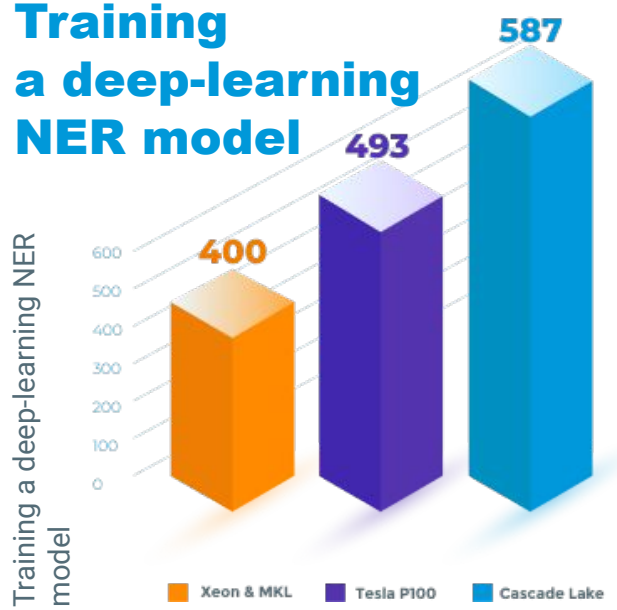
Benchmark made on **en_core_web_lg** dataset, micro-averaged F1 score

Speedup on Cluster

(less is better)



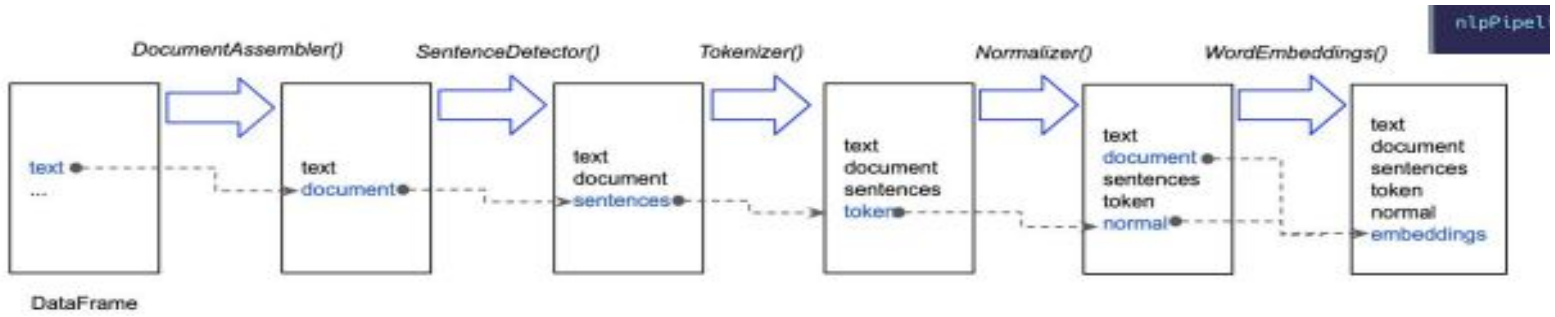
Training a deep-learning NER model



Accuracy | **Scalability** | **Speed**

John Snow LABS

nlp.LightPipeline



Thunder fast for few documents, not parallelizable

{;}
JSON

Json-friendly

John Snow LABS

Spark NLP can be run both at **cluster level**, leveraging all the nodes, and a **master-only evel**, 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 MLib **Pipelines** and ***fit/transform***.

```
text ='''December 2007 SUBORDINATED LOAN AGREEMENT. THI

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

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

```
!pip install johnsnowlabs
```

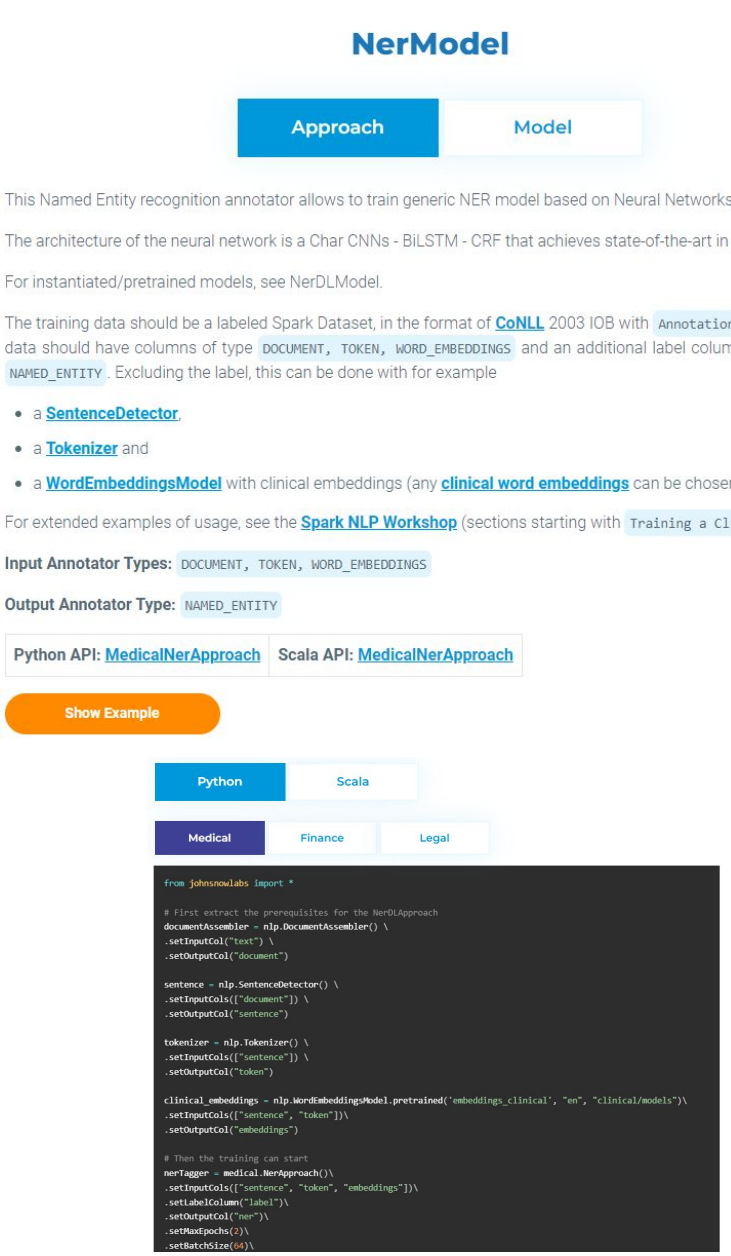
```
nlp.install(force_browser=True)
nlp.start()
```

Training and Inference

NER

Approach → **NerApproach (training)**

Model → **NerModel (inference)**



NerModel

Approach | Model

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 NerDLModel.

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

Legal NLP

Legal NLP

John Snow LABS

Legal Entity Recognition

This Loan Agreement dated as og November 17, 2014 (this «Agreement»), is made by and among Auxilium Pharmaceuticals, Inc., a corporation incorporated under the laws of the State of Delaware («U.S. Borrower»), Auxilium UK LTD, a private company limited by shares registered in England and Wales («UK Borrower» and, collectively with the U.S. Borrrower, the «Borrowers») and Endo Pharmaceuticals Inc., a corporation incorporated under the laws of the State of Delaware («Lender»).

Legal Entity Linking

UBS Group AG is a holding company, which engages in the provision of financial management solutions. It operates through the following segments: Global Wealth Management, Personal and Corporate Banking, Asset Management; Investment Bank and Corporate Center. The Global Wealth Management segment advises and offers financial services to wealthy private clients except those served by Wealth Management Americas which include banking and lending, wealth planning, and investment management. The Personal and Corporate segment offers financial products and services to private, corporate, and institutional clients in Switzerland. The Asset Management segment consists of investment management products and services, platform solutions and advisory support to institutions; wholesale intermediaries, and wealth management clients. The Investment Bank segment comprises investment advice, financial solutions, and capital markets access among corporate, institutional, and wealth management clients. The Corporate Center segment is involved in the services, group asset and ability management and non-core and legacy portfolio The company was founded on June 29, 1998 and is headquartered in.

Sector & Industry
Finance (4800) Major Banks (4805)

Fiscal Year End
December

Exchange/ISIN
SIX Swiss/CH0244767585

SEDOL
BRJL176

Investor Relations Contact
Martin A. Osinga

LEI
549300SZJ9VS8SGXANS1

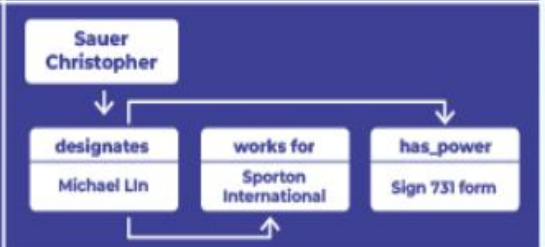
Assertion Status

Designates to sign the form of 731 and other documentation: → TRUE

Neither...nor...is subject to a denial...: → FALSE

...may require approval to... → POSSIBLE

Relation Extraction



- Legal Embeddings
- Document Splitting
- Knowledge Graphs
- Long Span Extraction with Question Answering
- Zero-shot Relation Extraction
- Clause Extraction
- Pattern Matching and Text Mining
- Deidentification

Trainable & Tunable

Scalable to a Cluster

Apache Spark ML Pipelines

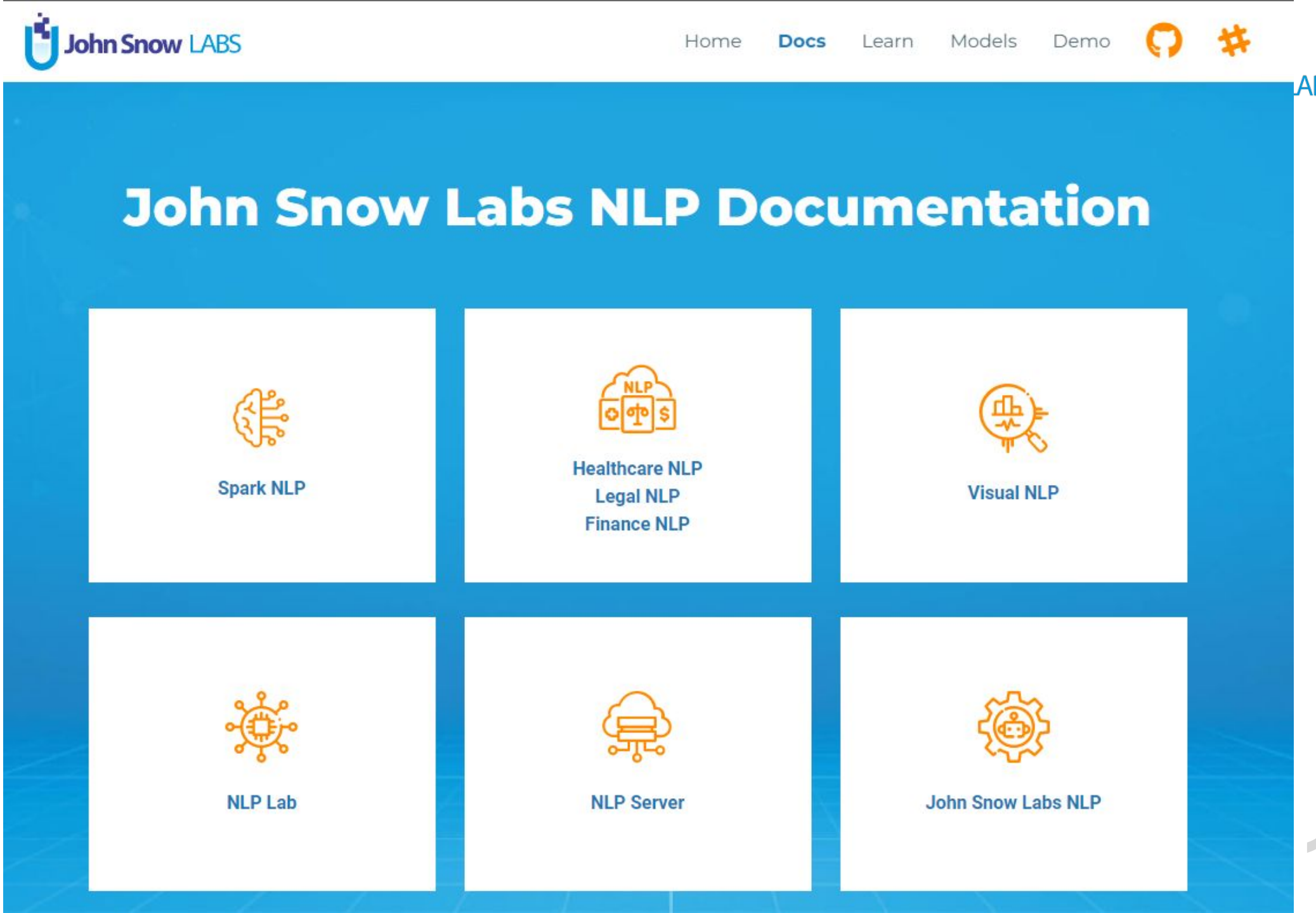
Transformers

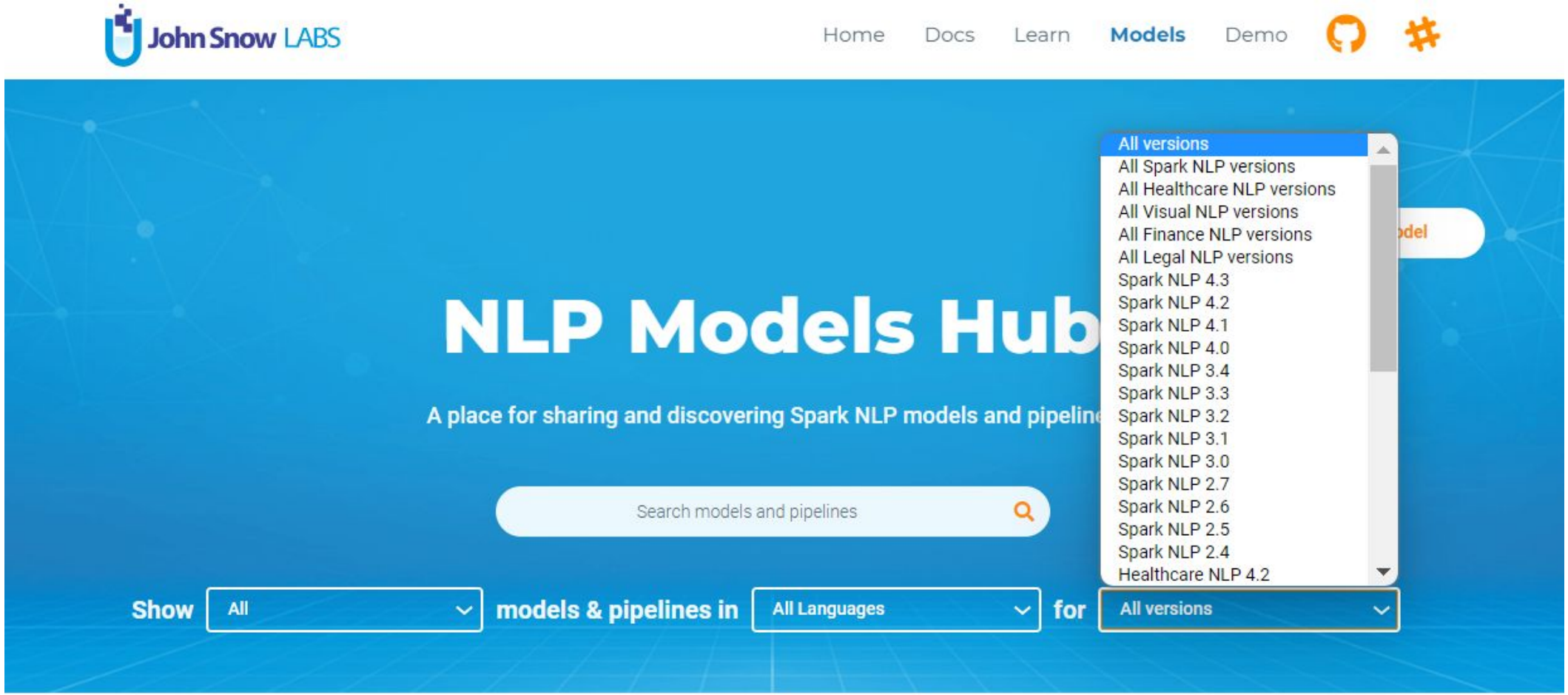
Fast Inference

LightPipeline

Hardware Optimized

intel NVIDIA


John Snow LABS
Home
Docs
Learn
Models
Demo
John Snow Labs NLP Documentation
Spark NLP
NLP
Healthcare NLP
Legal NLP
Finance NLP
Visual NLP
NLP Lab
NLP Server
John Snow Labs NLP


John Snow LABS
Home
Docs
Learn
Models
Demo
NLP Models Hub
A place for sharing and discovering Spark NLP models and pipeline
Search models and pipelines
All versions
All Spark NLP versions
All Healthcare NLP versions
All Visual NLP versions
All Finance NLP versions
All Legal NLP versions
Spark NLP 4.3
Spark NLP 4.2
Spark NLP 4.1
Spark NLP 4.0
Spark NLP 3.4
Spark NLP 3.3
Spark NLP 3.2
Spark NLP 3.1
Spark NLP 3.0
Spark NLP 2.7
Spark NLP 2.6
Spark NLP 2.5
Spark NLP 2.4
Healthcare NLP 4.2
Show
All
models & pipelines in
All Languages
for
All versions
13,728 Models & Pipelines Results:
Supported models only
SUPPORTED
Receipts Binary Classification
Date: 09.2022
task: Image Classification
Language: English
SUPPORTED
ESG Text Classification (Augmented, 26 classes)
Date: 09.2022
task: Text Classification
Language: English
SUPPORTED
Legal Zero-shot NER
Date: 09.2022
task: Named Entity Recognition
Language: English

ages

Radiology

Oncology

Resolve Entities to Terminology Codes

Databricks Solution Accelerators

Vaccines & Public Health

Mental Health

Risk Factors

Biomedical NLP

Genes, Variants, Phenotypes

Clinical Trials

Biomedical Research

Medical NLP: World Languages

German

Spanish

Financial Deidentification

Financial Document Splitting

Text Summarization

Legal NLP

Classify Legal Texts

Recognize Legal Entities

Understand Entities in Context

Extract Legal Relationships

Normalization & Data Augmentation

Legal Deidentification

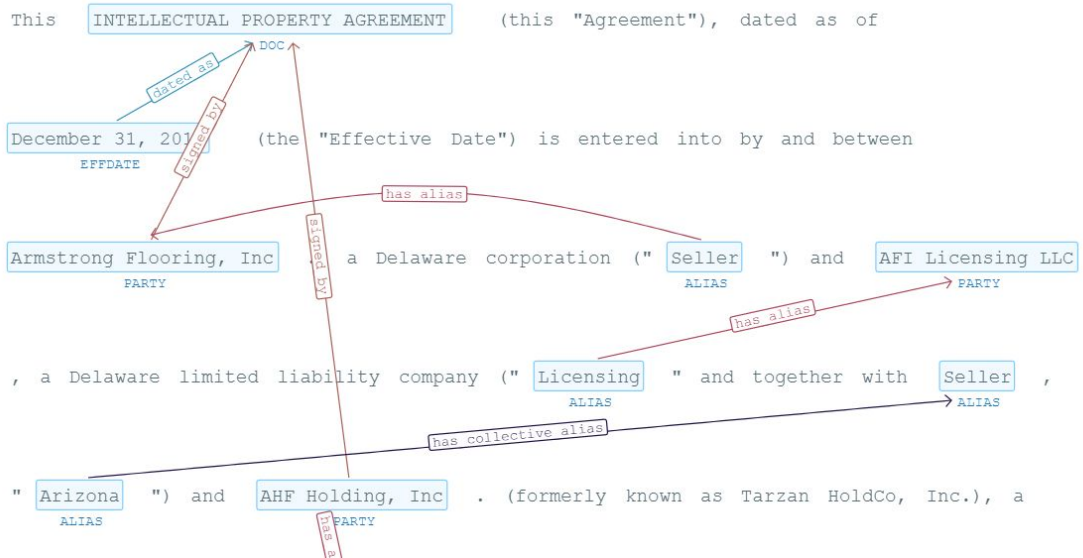
Long Document Splitting

John Snow LABS

Choose Sample Text

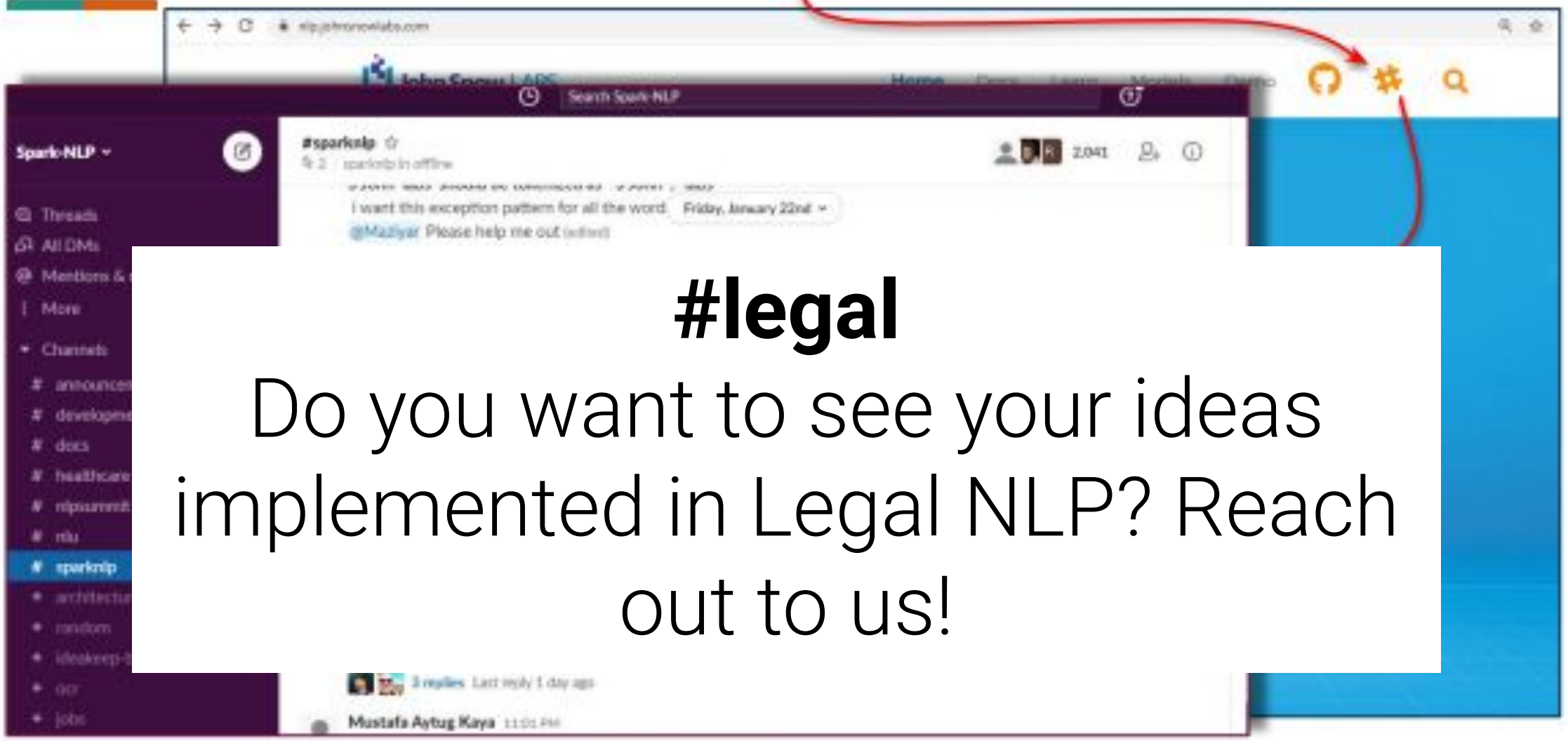
Exhibit 2.7 FORM OF TRADEMARK LICENSE AGREEMENT...

Text annotated with identified Named Entities



Spark-NLP Slack Channels

spark-nlp.slack.com



#legal

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

John Snow LABS

Medium-Spark NLP

Easy sentence similarity with BERT Sentence Embeddings using John Snow Labs NLU

1 Python line to Bert Sentence Embeddings and 5 more for Sentence similarity. Leverage your data to answer questions!

Christian Kasim Loan
Nov 20, 2020 · 4 min read

1 Python Line for ELMo Word Embeddings with John Snow Labs' NLU

Including Part of Speech, Named Entity Recognition, Emotion, and Sentiment Classification in the same line! With Bonus t-SNE plots and...

Christian Kasim Loan
Oct 24, 2020 · 7 min read

Turkish NER Model Training using Spark NLP, with the help of NLU.

The NLU miracle allows us to produce a perfect CoNLL file and a perfect CoNLL file makes the Turkish NER model perfect. This is the first...

Murat Guray
Oct 24, 2020 · 10 min read

Classification of Texts Written in Turkish Language Using Spark NLP

1 line of code for BERT, ALBERT, ELMO, ELECTRA, XLNET, GLOVE, Part of Speech with...

1 line to BERT Word Embeddings with NLU

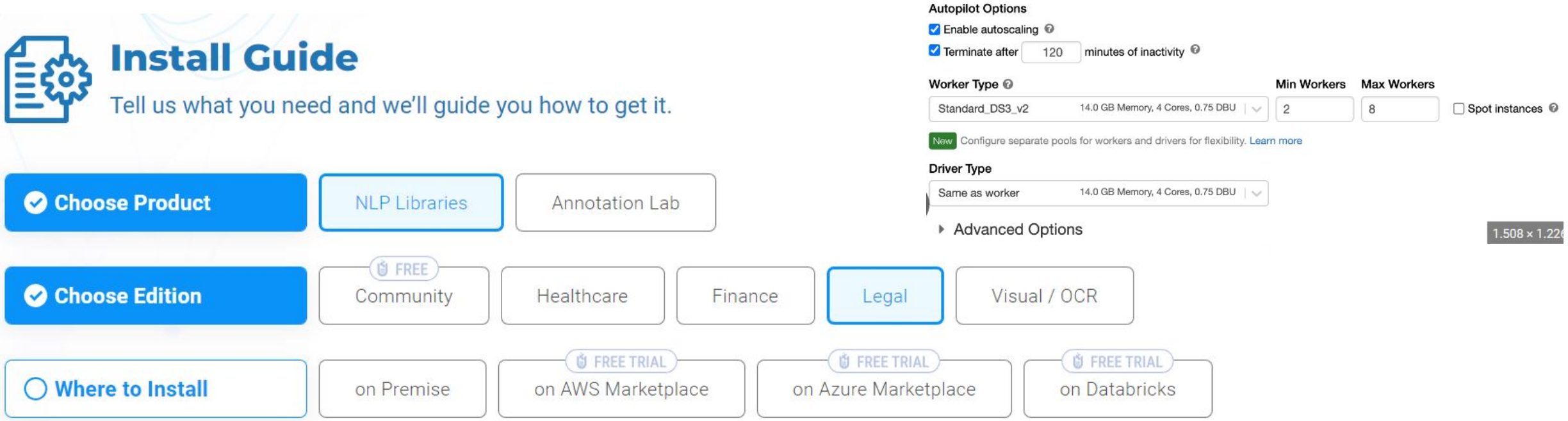
Including Part of Speech, Named Entity

Medium

Legal 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



John Snow LABS

STATE OF THE ART

Legal NLP, Visual NLP and NLP Lab

John Snow LABS

Projects
CUAD_legal_dataset
Analytics
Tasks
Train
Setup
Hub
Settings

j juan juan@johnsnowlabs.com

Undo [Ctrl+z] | Redo | Reset | Search | Instructions

Exhibit 10.2

Execution Version

INTELLECTUAL PROPERTY AGREEMENT DOC

This INTELLECTUAL PROPERTY AGREEMENT DOC (this "Agreement"), dated as of December 31, 2018 EFFDATE (the "Effective Date") is entered into by and between Armstrong Flooring, Inc PARTY., a Delaware corporation (" Seller ROLE ") and AFI Licensing LLC PARTY, a Delaware limited liability company (" Licensing ROLE " and together with Seller ROLE , " Arizona ROLE ") and AHF Holding, Inc PARTY. (formerly known as Tarzan HoldCo, Inc FORMER_PARTY_NAME .), a Delaware corporation (" Buyer ROLE ") and Armstrong Hardwood Flooring Company PARTY, a Tennessee corporation (the " Company ROLE " and together with Buyer ROLE the " Buyer Entities ROLE ") (each of Arizona ROLE on the one hand and the Buyer Entities ROLE on the other hand, a "Party" and collectively, the "Parties").

WHEREAS, Seller and Buyer have entered into that certain Stock Purchase Agreement, dated November 14, 2018 (the "Stock Purchase Agreement");
WHEREAS, pursuant to the Stock Purchase Agreement, Seller has agreed to sell and transfer, and Buyer has agreed to purchase and acquire, all of Seller's right, title and interest in and to Armstrong Wood Products, Inc., a Delaware corporation ("AWP") and its Subsidiaries, the Company and HomerWood Hardwood Flooring Company, a Delaware corporation ("HHFC," and together with the Company, the "Company Subsidiaries" and together with AWP, the "Company Entities" and each a "Company Entity") by way of a purchase by Buyer and sale by Seller of the Shares, all upon the terms and condition set forth therein;

View 3600 Characters Per Page | 1 2 3 4 5 ••• 14 | Page 1

[Ctrl+Space] Next

02:34:10

aleksei
- ID 512007
  Updated 6 months ago
  00:14:10

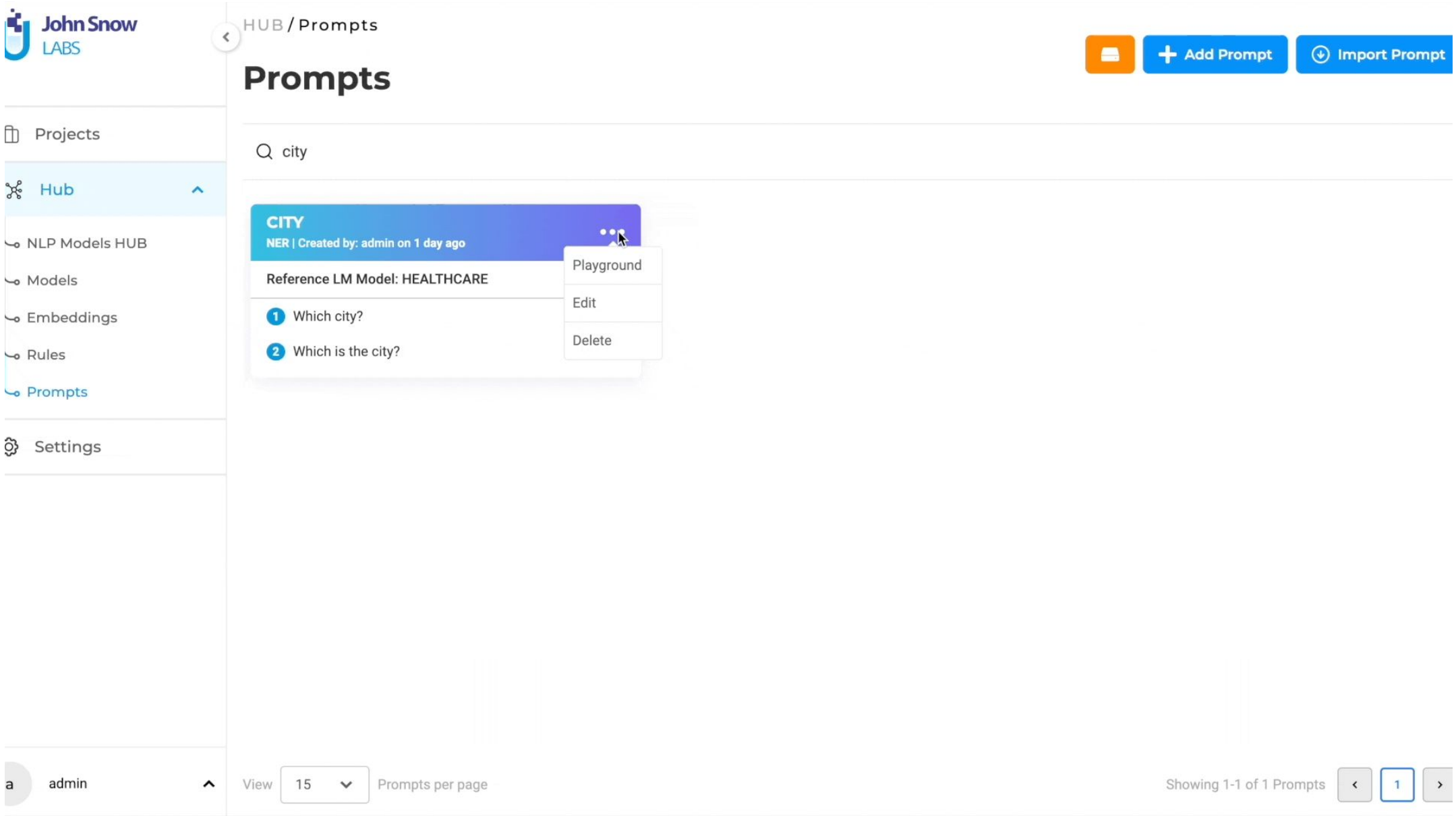
aubrey
- ID 512006
  Submitted 5 months ago
  01:13:21
- ID 512004
  Submitted 8 months ago
  00:02:59
- ID 512003
  Submitted 8 months ago
  00:05:26
- ID 512002
  Submitted 8 months ago
  01:21:58

juan
- ~~ID 512001~~

Predictions

No predictions

Zero-shot and Prompt engineering in NLP Lab



STATE OF THE ART

Text Splitting

Legal NLP

Splitting Legal 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 `setCustomBoundsStragegy()`.

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 disc
Processing Services and facilitate procurement of Merchant Process
without limitation by references to such pricing information and Me

2.2 Customer Agreements.

a) Subscriptions. Allscripts and its Affiliates may sell Subscriptions fo
years on a subscription basis to Persons who subsequently execute a
into Customer Agreements with terms longer than four (4) years wit
each instance in writing in advance, which consent will not be unrea

```
text = """
4. GRANT OF KNOW-HOW LICENSE
4.1 Arizona Know-How Grant. Subject to the terms and conditions of this Agreement, Arizona hereby gr
4.2 Company Know-How Grant. Subject to the terms and conditions of this Agreement, the Company hereb
5. GRANT OF PATENT LICENSE
5.1 Arizona Patent Grant. Subject to the terms and conditions of this Agreement, Arizona hereby gran
"""
```

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

paragraphDetector = nlp.SentenceDetector()\
    .setInputCols(["document"])\
    .setOutputCol("paragraph")\
    .setCustomBounds(["\n[\d\.]+"])\
    .setCustomBoundsStrategy('prepend')\
```

Splitting Legal 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

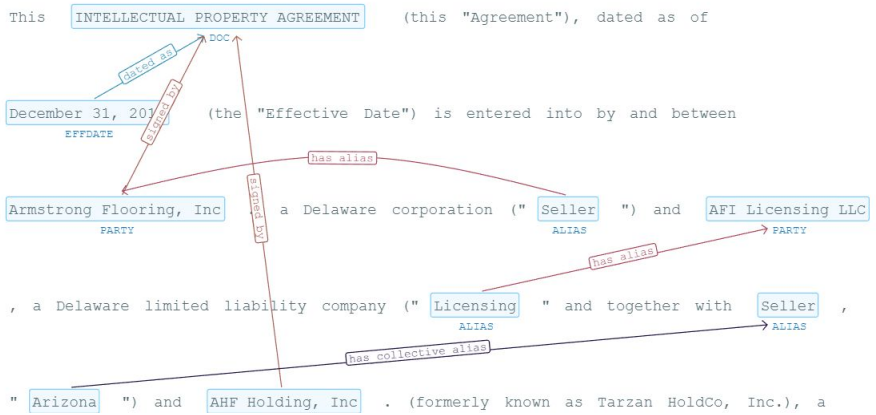
Splitting Legal texts

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.

WHEREAS, the Corporation wishes to provide:
a) investment advise;
b) management services;
c) administrative services
…

- For **Assertion**, as with Text Classification, you may want to send the model more than just a sentence.
- For **Relation Extraction**, is quite common entities are in different sentences, so you may want to split by paragraph



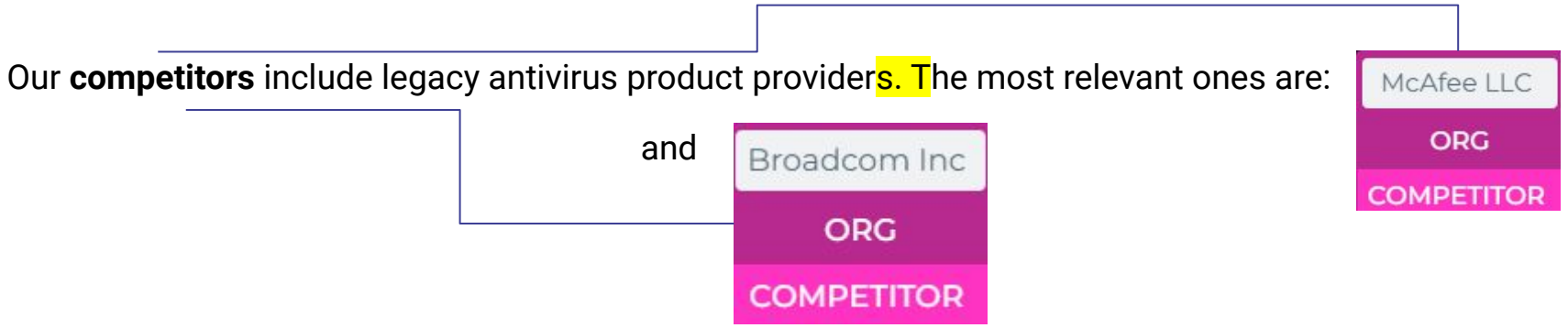
Splitting Legal texts

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.

WHEREAS, the Corporation wishes to provide:
a) investment advise;
b) management services;
c) administrative services
…

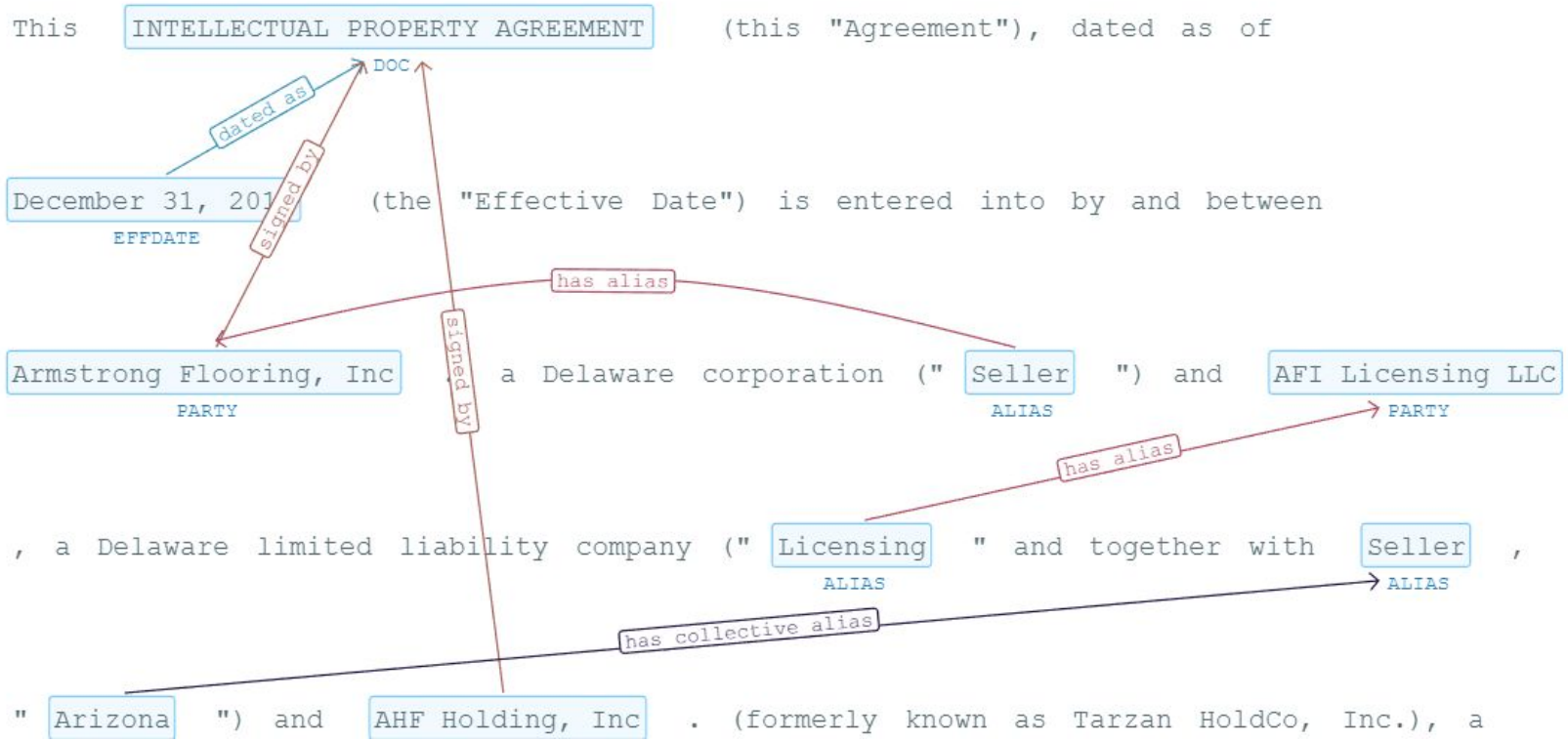
- For **Assertion**, as with Text Classification, you may want to send the model more than just a sentence.



Splitting Legal 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).

John Snow LABS

STATE OF THE ART

Language Models

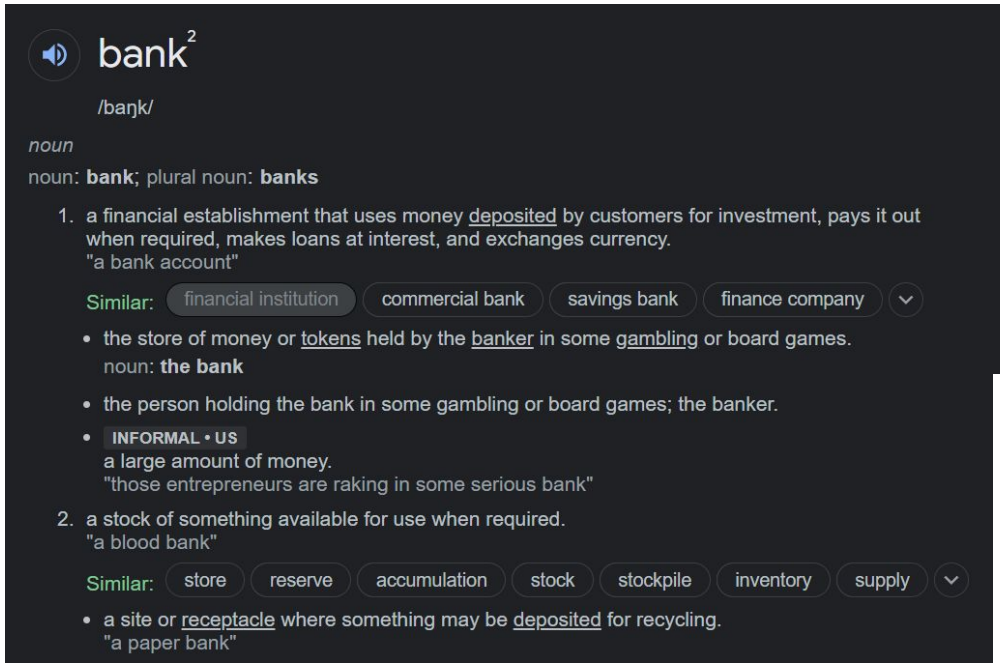
Legal NLP

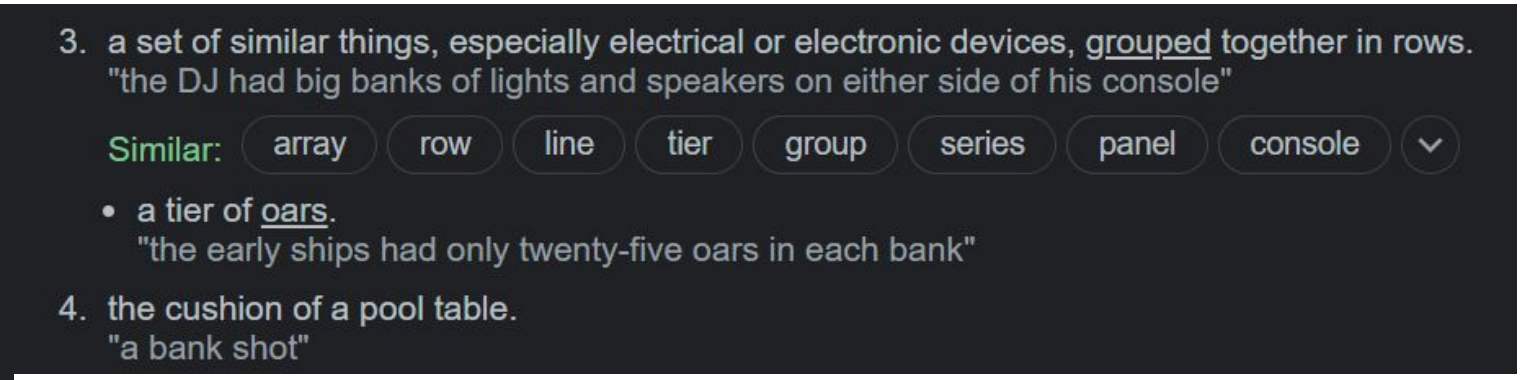
Language Models and Embeddings

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 it 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.



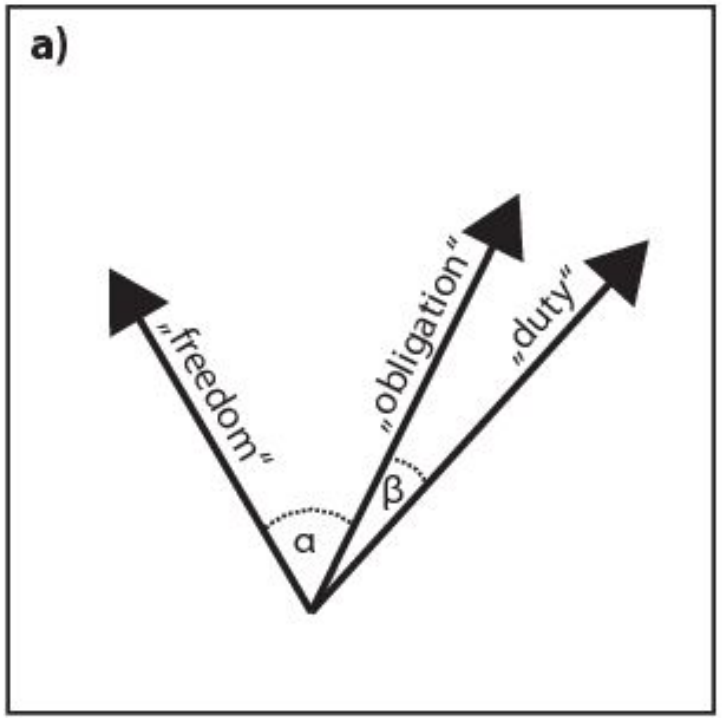


All of these will have different embeddings (numerical representations) in context!

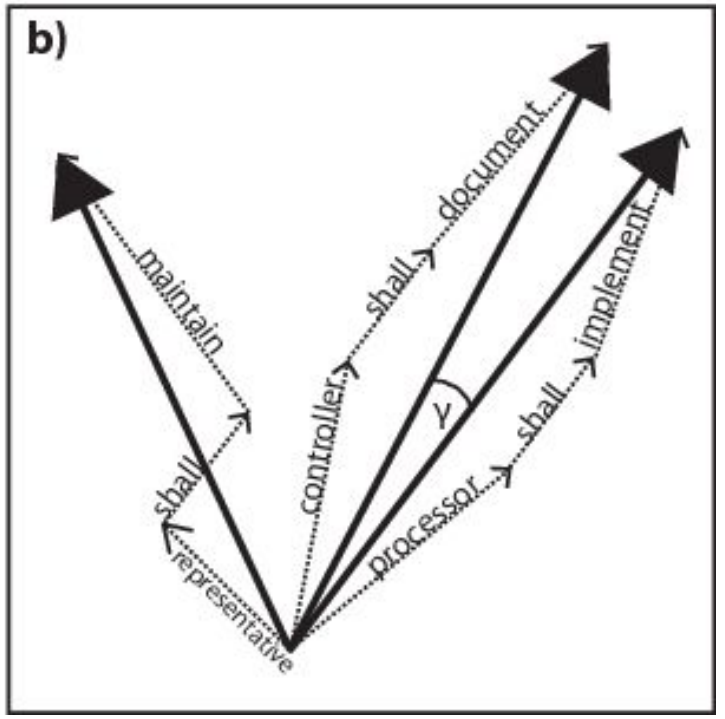
Language Models and Embeddings

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



Legal Word Embeddings



Legal Sentence Embeddings

Language Models and Embeddings

Domain specificity

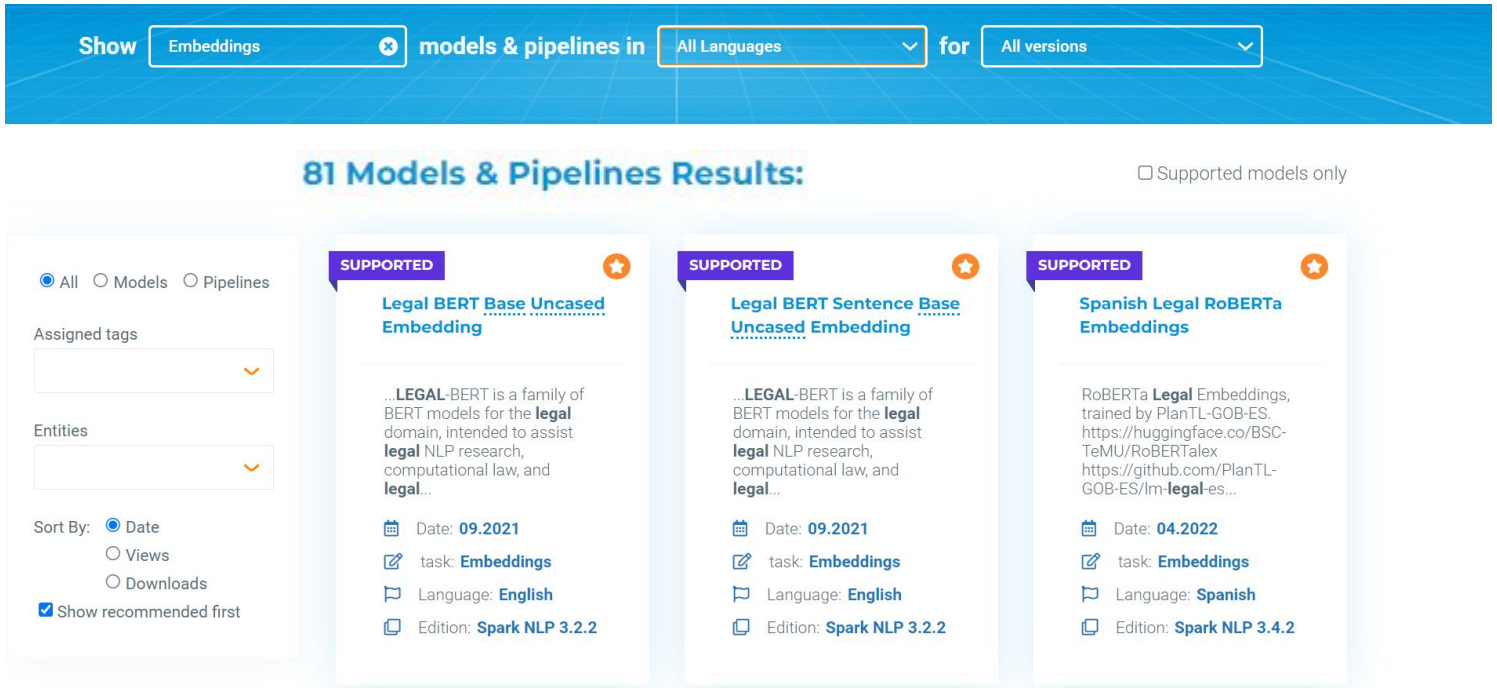
- As a consequence of their context-specificity, it's very important you use domain specific embeddings. Fortunately, we have **more than 30** Legal NLP Language Models in Models Hub, including English, Portuguese and Spanish.

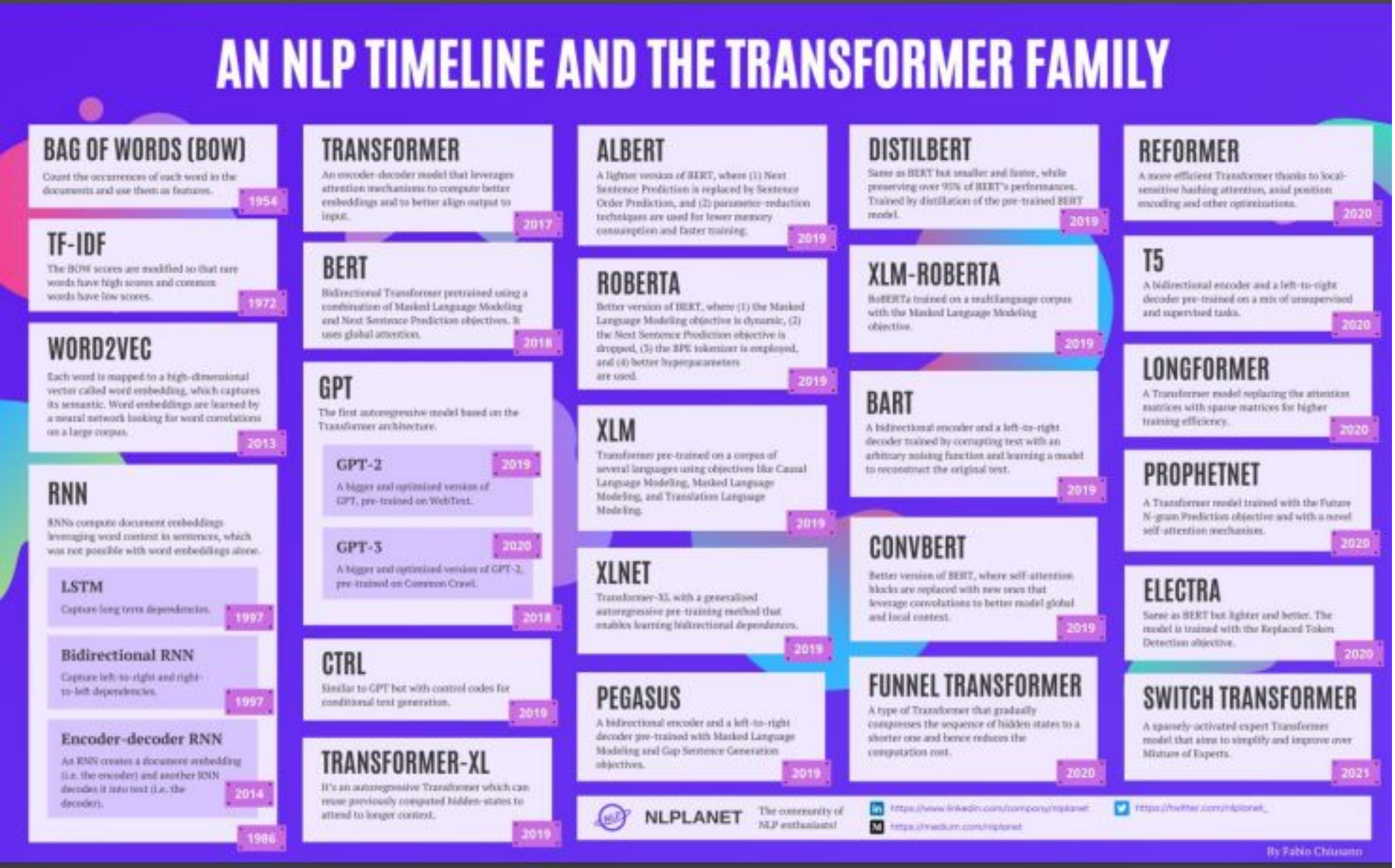
Word vs Sentence

- 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.




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
1986
TRANSFORMER
An encoder-decoder model that leverages attention mechanisms 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.
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
2018
CTRL
Similar to GPT but with control codes for conditional text generation.
2019
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 generalized autoregressive pre-training method that enables learning bidirectional dependences.
2019
PEGASUS
A bidirectional encoder and a left-to-right decoder pre-trained with Masked Language Modeling and Gap Sentence Generation objectives.
2019
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 multilanguage 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 noising 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
NLPLANET
The community of NLP enthusiasts!
https://www.linkedin.com/company/nlplanet
https://twitter.com/nlplanet_
https://medium.com/nlplanet
By Fabio Chiusano

Text Classification with Word & Sentence Embeddings & Transformers

Glove
ELMO
BERT
Electra
Longformer
Universal Sentence Encoder
Albert
XLNet
RoBerta
DistilBert
XlmRoBerta
Bert Sentence Embeddings
Word2Vec
Bert Token Classifier
DistilBert Token Classifier
Albert Token Classifier
RoBerta Token Classifier
RoBerta Token Classifier
XlmRoBerta Token Classifier
XLNet Token Classifier
LongFormer Token Classifier
Bert Sequence Classifier
DistilBert Sequence Classifier
DistilBert Sequence Classifier
Albert Sequence Classifier
RoBerta Sequence Classifier
XlmRoBerta Sequence Classifier
XLNet Sequence Classifier
Longformer Sequence Classifier
RoBerta Sentence Embeddings
XlmRoBerta Sentence Embeddings

Non Transformer Embedding
Transformer Embedding
Transformer Sequence Classifier
Transformer Token Classifier

	Sentence	label
0	a stirring , funny and finally transporting re imagining of beauty and the beast and 1930s	1
1	apparently reassembled from the cutting room floor of any given daytime soap	0
…	…	…
1,999	the movie is undone by a filmmaking methodology that 's just experimental enough	1

DistilBERT

Already (pre-)trained

	Sentence Embeddings				label
	0	1	…	767	
0	-0.215	-0.1402	…	0.261	1
1	-0.172	-0.144	…	0.371	0
…	…	…	…	…	…
1,999	0.124	0.014	…	0.274	1

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

Legal NLP

Classify clauses and whole documents

This agreement, or any term thereof, may be changed or waived only by written amendment, signed by the party against whom enforcement of such change or waiver is sought.

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

Classification Confidence: **99.92%**

We anticipate the value of our company to continue to rise for over the next few years, increasing dividends for shareholders.

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

Classification Confidence: **99.75%**

Each party warrants and represents that it has full capacity and authority, all necessary licenses, permits and consents to enter into and perform its obligations under the agreement.

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

Classification Confidence: **99.76%**

No loans shall be contracted on behalf of the company and no evidences of indebtedness shall be issued in its name unless authorized by a resolution of the board of managers

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

Classification Confidence: **100.0%**

The powers of the company shall be exercised by or under the authority of, and the business and affairs of the company shall be managed under the direction of, the member.

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

Classification Confidence: **99.97%**

Legal 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 400 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.	This is the most common type of classifiers in Legal NLP. 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 **Legal clauses**	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 **Legal 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.

Legal NLP Classification

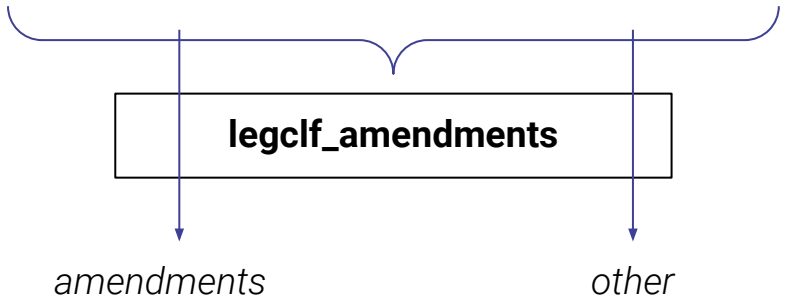
- By **output** type or **class assigned to the input text**

Binary Classifiers

Return *true* or *false* values. For example, our more than 300 Clause Binary classifiers, which return the **name of the clause** if it is classified as such, or **other** otherwise.

This agreement, or any term thereof, may be changed or waived only by written amendment, signed by the party against whom enforcement of such change or waiver is sought.

We anticipate the value of our company to continue to rise for over the next few years, increasing dividends for shareholders.



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).

<u>It's not suitable for:</u>

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

The Commission considered that a special feature in the present case was the fact that the applicant had chosen to express himself through poetry. However, even taking into account the prerogatives of a poet, it found that parts of the applicant's poems glorified armed rebellion against the Turkish State and martyrdom in that fight.

This text has been classified as : **COMMISSION/CHAMBER**

or **APPLICANT** or ...

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).

<u>It's not suitable for:</u>

- Big number of classes (more than 100)

(a) No failure or delay by the Administrative Agent or any Lender in exercising any right or power hereunder shall operate as a waiver thereof, nor shall any single or partial exercise of any such right or power, or any abandonment or discontinuance of steps to enforce such a right or power, preclude any other or further exercise thereof or the exercise of any other right or power

This text has been classified as: **waivers**, **amendments**

Legal NLP Classification

We have **thousands of different legal clauses** in documents, and often times, , they **are not disjoint**: a paragraph could have information about **some different legal clauses at the same time**.

Contribution 66k

Indemnification 424k

Indemnification and Contribution 40k

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 (or actions in respect thereof) arise out of or are based upon any untrue statement or alleged untrue statement of a material fact contained in the registration statement for the registration of the Securities as originally filed or in any amendment thereof, or in the Basic Prospectus, any Preliminary Final Prospectus or the Final Prospectus, or in any amendment thereof or supplement thereto, or arise out of or are based upon the omission or alleged omission 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 incurred, for any legal or other expenses reasonably incurred by them in connection with investigating or defending any such loss, claim, damage, liability or action; 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 such untrue statement or alleged untrue statement or omission or alleged omission 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.

indemnification - TRUE

contribution - TRUE

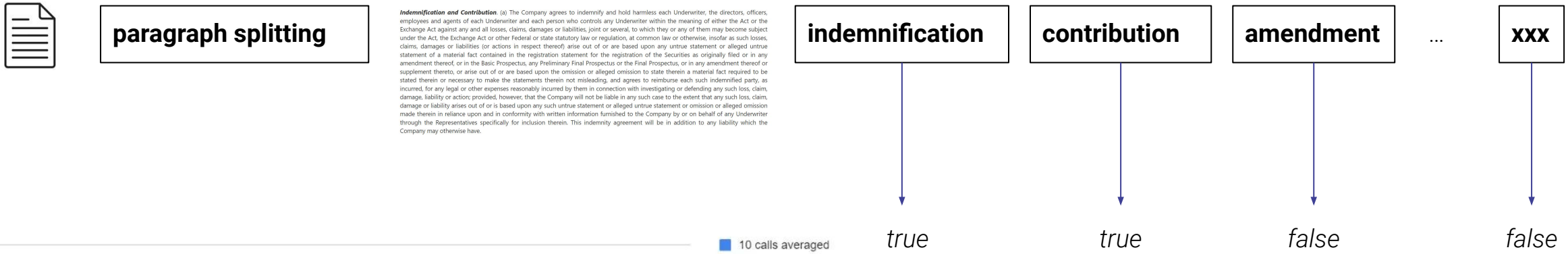
amendment - FALSE

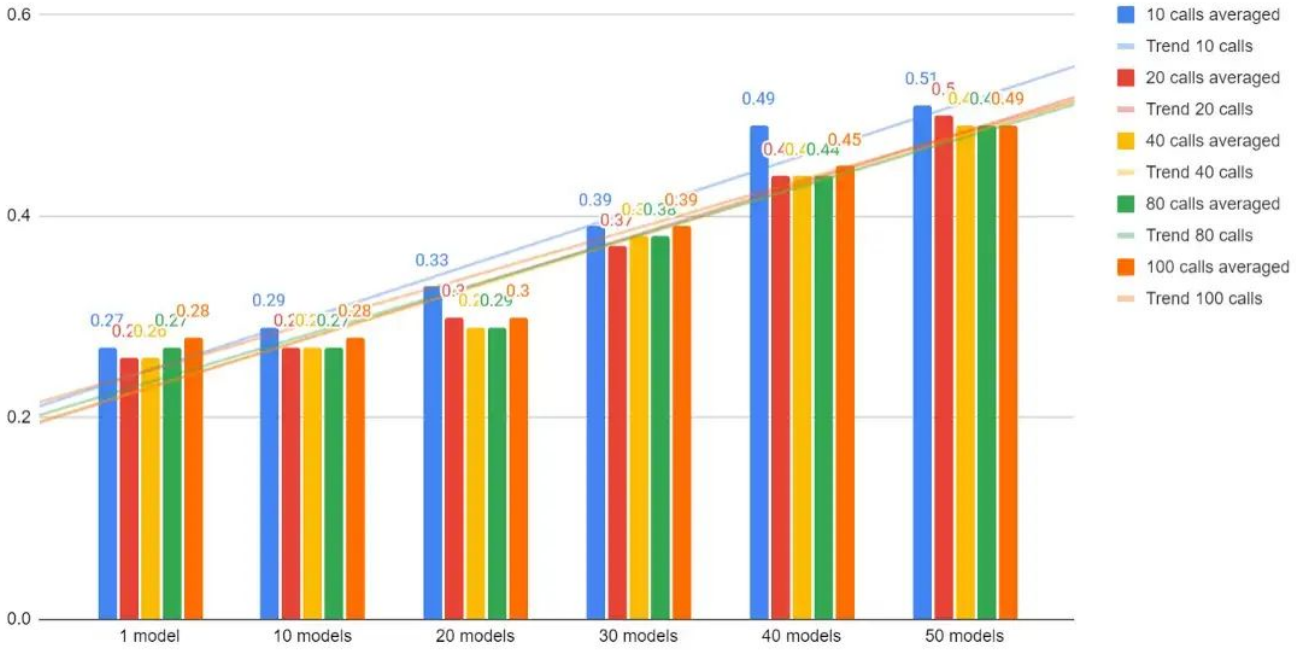
...

xxx - FALSE

Legal NLP Classification

We have **thousands of different legal clauses** in documents, and often times, , they **are not disjoint**: a paragraph could have information about **some different legal clauses at the same time**.





Performance-wise, these models are super light. You can include hundreds of them and have them predicting on a paragraph in less than 0.5 seconds. Check https://medium.com/p/a2f9b899de92

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 **Legal NLP Text Classifiers.**

If the layout information is important:

2) Use **Legal 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 Legal NLP required.

Rolien Gasner

yBank, LLC
Munich
Germany

555-555-5555
hello@kickresume.com

27/05/2020
Application for the position of Financial Services Associate

Dear Sir/Madam,

As I have been following your company and work for many years, I was 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.

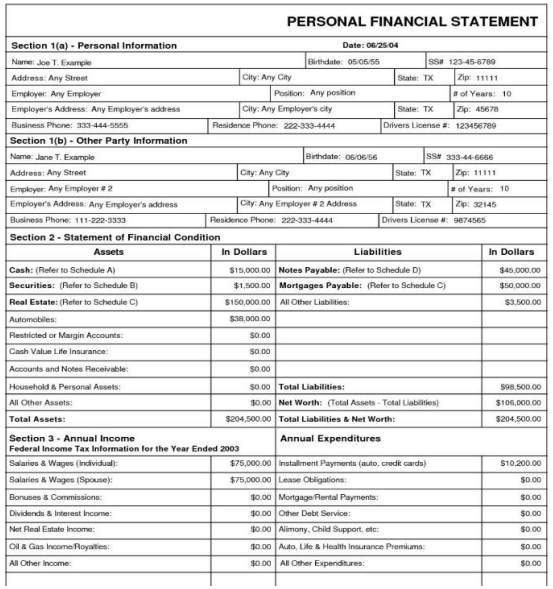
My name is Rolien Gasner and I am the American University graduate with a bachelor's degree in Economics & Finance. At the university, I was among the top 3% of students with the best academic results (3.98 GPA) and I also won the Dean's Award once for representing the school at multiple international economics competitions. My studies have allowed me to become an effective leader and helped me to acquire excellent analytical and problem-solving skills.

Next, I worked as a Financial Services Associate at Virtu 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 reviews, processed transactions, and opened new accounts. Throughout the years, I have proved numerous times that I am a pro-active and reliable person with the crucial ability to function well in deadline-driven and fast-paced environments. Furthermore, I have been recognized as a top achieving employee by the company executives for meeting all assigned targets. Offering the experience with all software programs necessary for the job, I 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

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



PERSONAL FINANCIAL STATEMENT

Section 1(a) - Personal Information				**Date: 06/25/04**	
Name: Joe T. Example		Birthdate: 05/05/55		SS# 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		SS# 333-44-6666	
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)	$106,000.00
Total Assets:	$204,500.00	**Total Liabilities & Net Worth:**	$204,500.00
Section 3 - Annual Income **Federal Income Tax Information for the Year Ended 2003**		**Annual Expenditures**	
Salaries & Wages (Individual):	$75,000.00	Installment 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

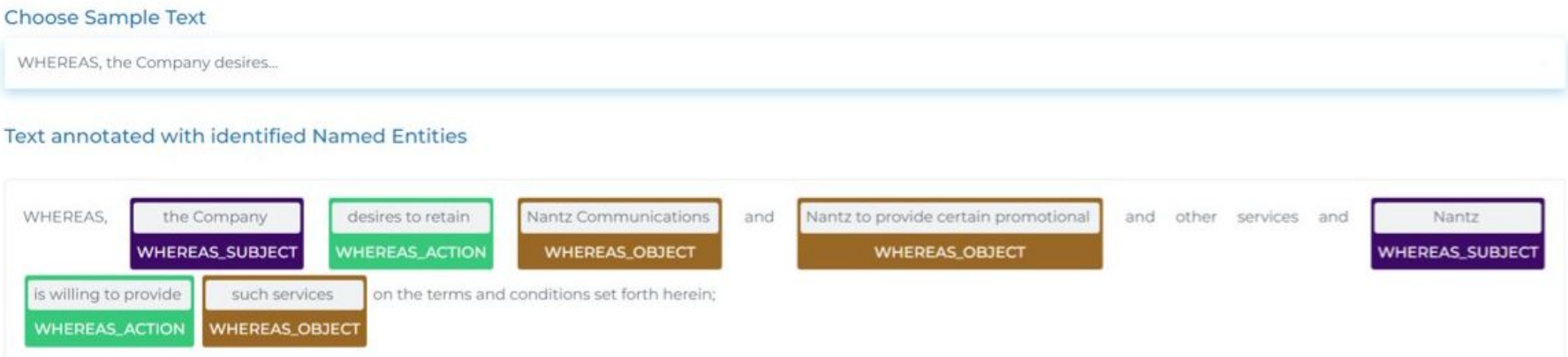
STATE OF THE ART

Named Entity Recognition

Legal NLP

Recognize Legal Entities in Documents

Extract who, what, to whom?



Extract main actions in an agreement

Legal NLP Named Entity Recognition

John Snow LABS

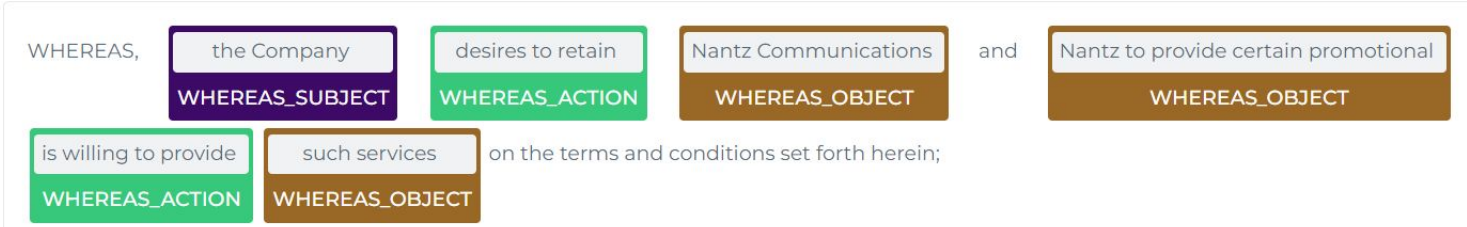
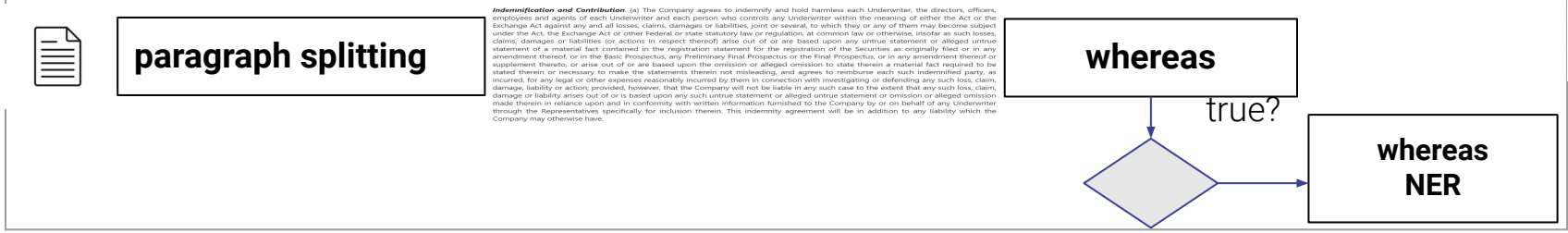
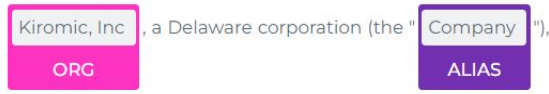
NER is the NLP task in charge of detecting relevant words / chunks in texts and categorize them.

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

Sentences	**Paragraphs**
To do NER at sentence level, after you split a text **into sentences with SentenceDetector.** Used in most of the cases, since the context of a relevant entity is found in the surroundings of its sentence.	To do NER at sentence level, after you split a text **into paragraphs with SentenceDetector, not into sentences.** We may need to do this in some exceptional cases: *WHEREAS, the Corporation wishes to provide:* *a) investment advise;* *b) management services;* *c) administrative services* *…*

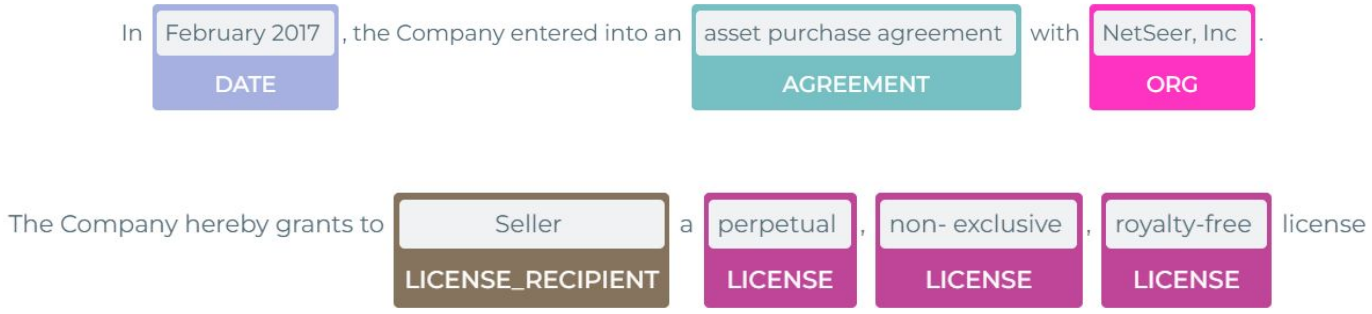
Legal NLP Named Entity Recognition

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

Clause Level	Document Level
NER entities can be only found in some specific parts of the document. For example, **whereas**, **indemnification**, **termination, the introduction of the parties in an agreement,** etc.  There is **no point** in applying these clause 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.**  	NER entities can be only found iall over the document. You can apply NER to the whole document. For example, **PARTY**, **ALIAS** (how the company is mention through the document), **OBLIGATIONS** (sentences describing what a PARTY has agreed to do)**, LAW, PERSON, DATE, COUNTRY, etc.** 

Legal NLP Zero-shot NER

	Entity	Question
0	DATE	['When was the company acquisition?', 'When was the comp
1	ORG	['Which company?', 'Which was the company acquisition?']
2	STATE	['Which state?']
3	AGREEMENT	['What kind of agreement?']
4	LICENSE	['What kind of license?']
5	LICENSE_REC	['To whom the license is granted?']
6	ALIAS	['Which is the alias?']



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.**

Annotators

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

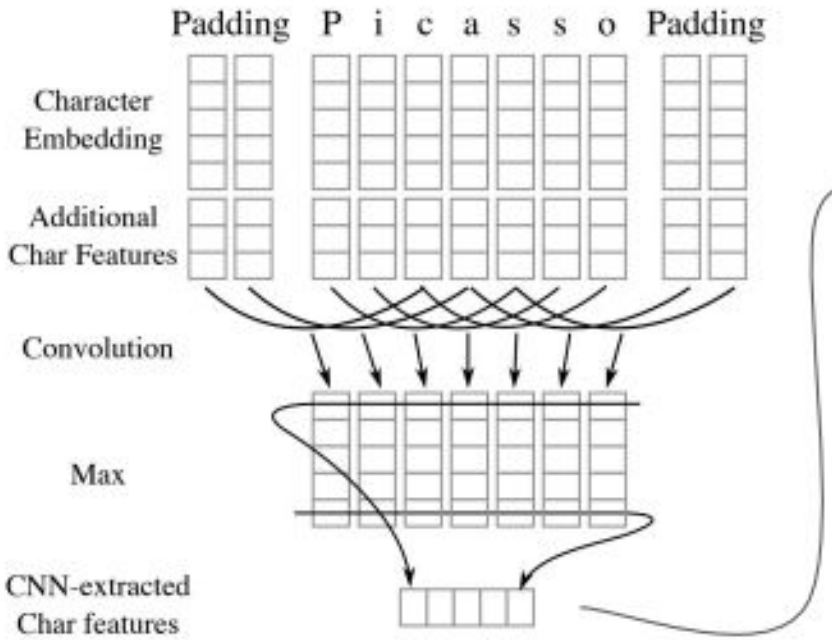


Figure 2: The convolutional neural network extracts character features from each word. The character embedding and (optionally) the character type feature vector are computed through lookup tables. Then, they are concatenated and passed into the CNN.

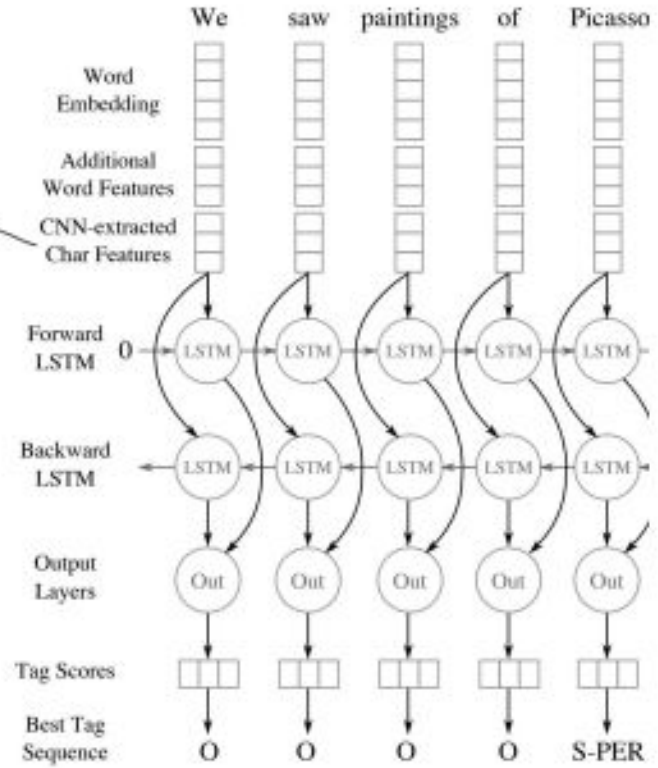


Figure 1: The (unrolled) BLSTM for tagging named entities. Multiple tables look up word-level feature vectors. The CNN (Figure 2) extracts a fixed length feature vector from character-level features. For each word, these vectors are concatenated and fed to the BLSTM network and then to the output layers (Figure 3).

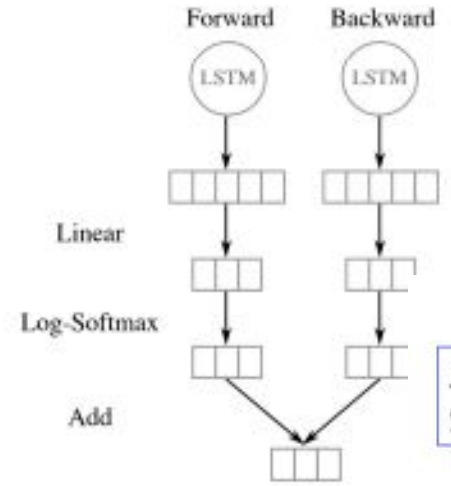


Figure 3: The output layers ("Out" in Fig output into a score for each tag category.

Char-CNN-BiL

John S

BIO schema

John	B-PER
Smith	I-PER
lives	O
in	O
New	B-LOC
York	I-LOC

John Smith ⇒ PERSON
New York ⇒ LOCATION

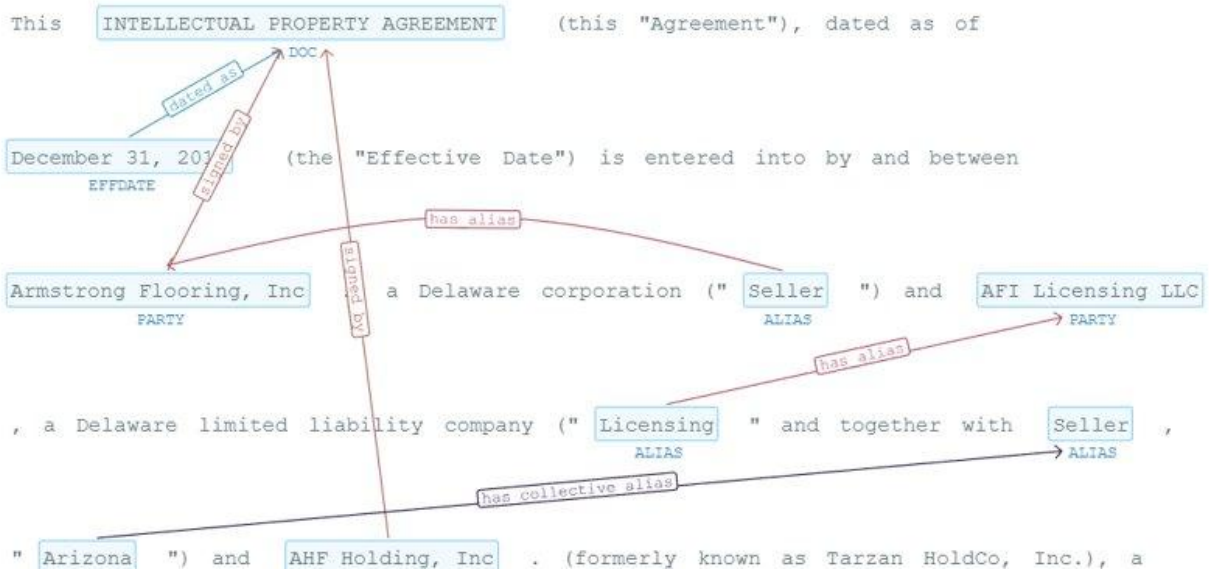
word	POS_tag	chunk_tag	NER_tag
She	PRP	O	B-person
presented	VBD	B-VP	O
with	IN	B-VP	O
left	JJ	B-NP	B-problem
upper	JJ	I-NP	I-problem
quadrant	NN	I-NP	I-problem
pain	NN	I-NP	I-problem
as	RB	O	O
well	RB	O	O
as	IN	B-VP	O
nausea	NN	B-NP	B-problem

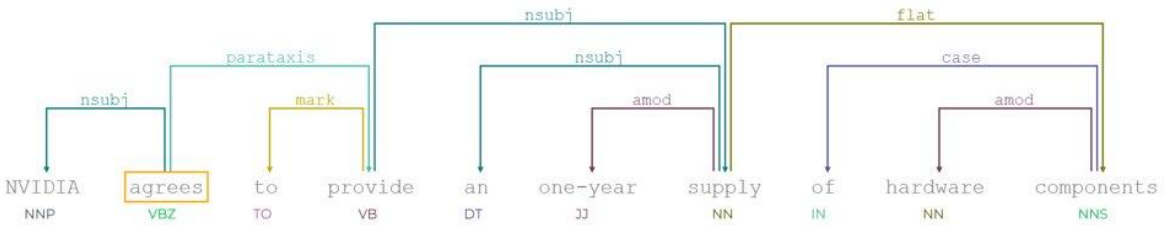
STATE OF THE ART

Relation Extraction

Legal NLP

Understand Relationships in Legal 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 Legal 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?
RelationExtraction will try to understand if each combination of 2 entities is a category. This may be **low performant or have undesired results,**which you can prevent by: ● Setting which **combinations** of entities may be checked.	RelationExtraction will try to understand if each combination of 2 entities is a category. This may be **low performant or have undesired results,**which you can prevent by: ● Set a **maximum distance** between entities.

Relation Extraction

```
"""
ONLY NEEDED IF YOU WANT TO FILTER RELATION PAIRS OR SYNTACTIC DISTANCE
pos_tagger = PerceptronModel()\
    .pretrained("pos_clinical", "en", "clinical/models") \
    .setInputCols(["document", "tokens"])\
    .setOutputCol("pos_tags")

dependency_parser = DependencyParserModel() \
    .pretrained("dependency_conllu", "en") \
    .setInputCols(["document", "pos_tags", "tokens"]) \
    .setOutputCol("dependencies")

Set a filter on pairs of named entities which will be treated as relation candidates
re_filter = RENerChunksFilter()\
    .setInputCols(["ner_chunks", "dependencies"])\
    .setOutputCol("re_ner_chunks")\
    .setMaxSyntacticDistance(7)\
    .setRelationPairs(['PARTY-ALIAS', 'DOC-PARTY', 'DOC-EFFDATE'])
"""
re_model = legal.RelationExtractionDLModel.pretrained("legre_contract_doc_parties", "en", "legal/models")\
    .setPredictionThreshold(0.5)\
    .setInputCols(["ner_chunk", "sentence"])\
    .setOutputCol("relations")

pipeline = nlp.Pipeline(stages=[
        document_assembler,
        sentence_detector,
        tokenizer,
        embeddings,
        ner_model,
        ner_converter,
        re_model
        ])
```

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

You can use:

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

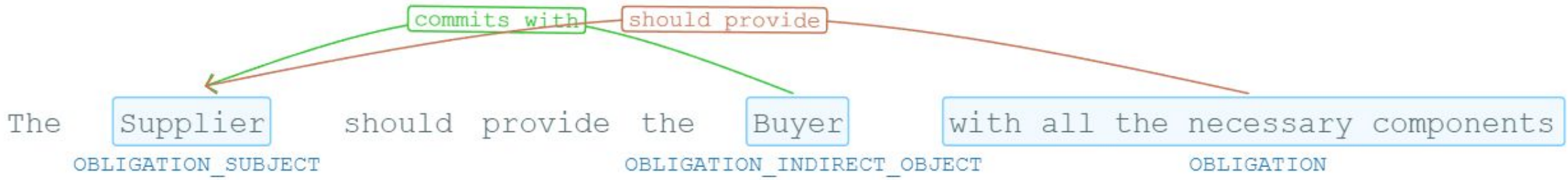
These steps are optional, as you can see in some examples they will just be commented out. In other cases it will be crucial due to false positives or negatives.

Zero-shot Relation Extraction

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

```
re_model = legal.ZeroShotRelationExtractionModel.pretrained("legre_zero_shot", "en", "legal/models")\
    .setInputCols(["ner_chunk", "document"]) \
    .setOutputCol("relations")

re_model.setRelationalCategories({
    "should_provide": ["{OBLIGATION_SUBJECT} will provide {OBLIGATION}", "{OBLIGATION_SUBJECT} should provide {OBLIGATION}"],
    "commits_with": ["{OBLIGATION_SUBJECT} to {OBLIGATION_INDIRECT_OBJECT}", "{OBLIGATION_SUBJECT} with {OBLIGATION_INDIRECT_OBJECT}"],
    "commits_to": ["{OBLIGATION_SUBJECT} commits to {OBLIGATION}"],
    "agree_to": ["{OBLIGATION_SUBJECT} agrees to {OBLIGATION}"],
})
```

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** (from NER) 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*.:

John Snow LABS

STATE OF THE ART

Assertion Status

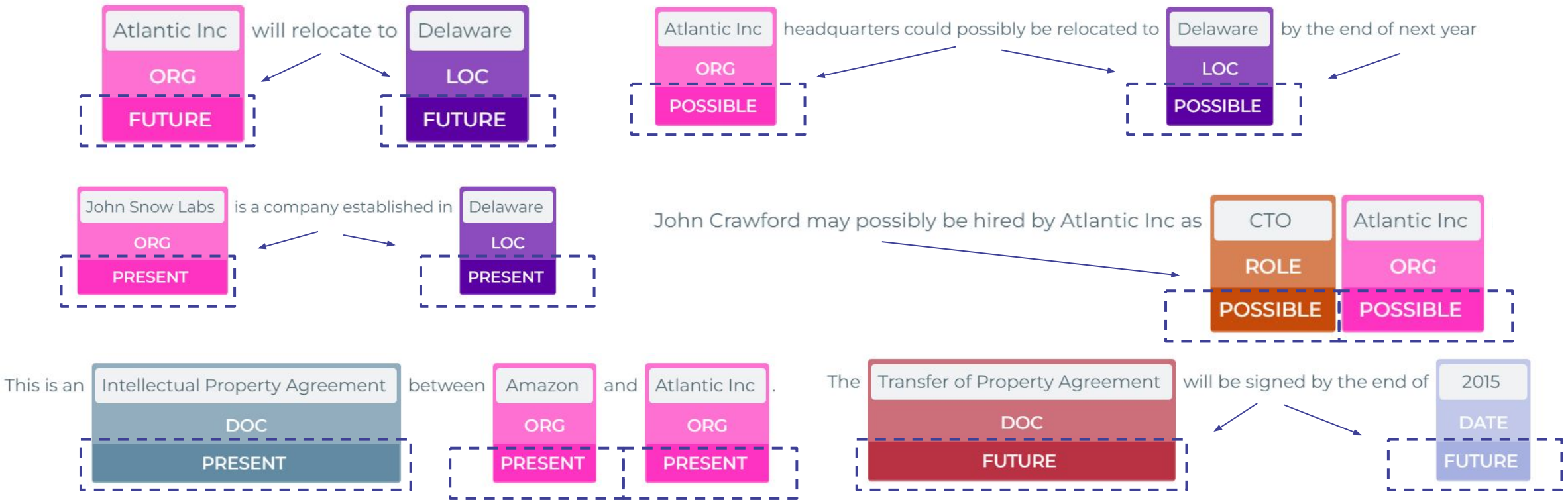
Legal NLP

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 wanto 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).**



Annotators

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

John Snow LABS

STATE OF THE ART

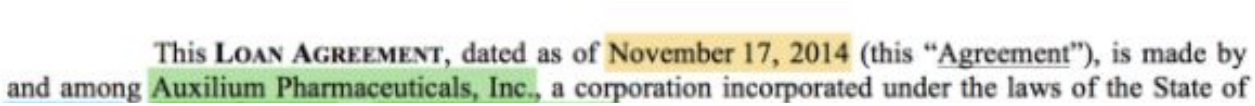
Entity Resolution

Legal NLP

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**



Auxilium Pharmaceuticals, Inc (from NER)

- Normalized name (Edgar): **AUXILIUM PHARMACEUTICALS INC**
- Unique ID (Edgar): **0001182128**

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

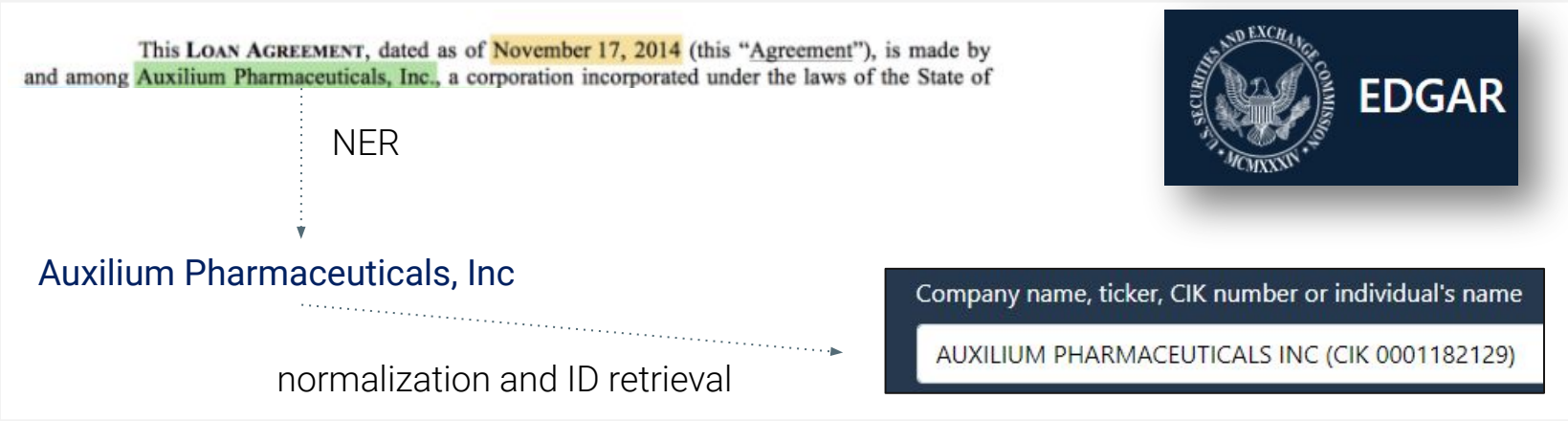
Entity Resolution

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).**



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

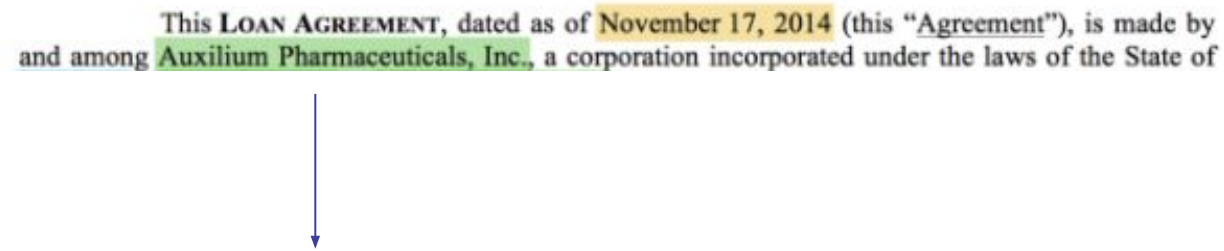
Data Augmenation with Chunk Mappers

Legal NLP

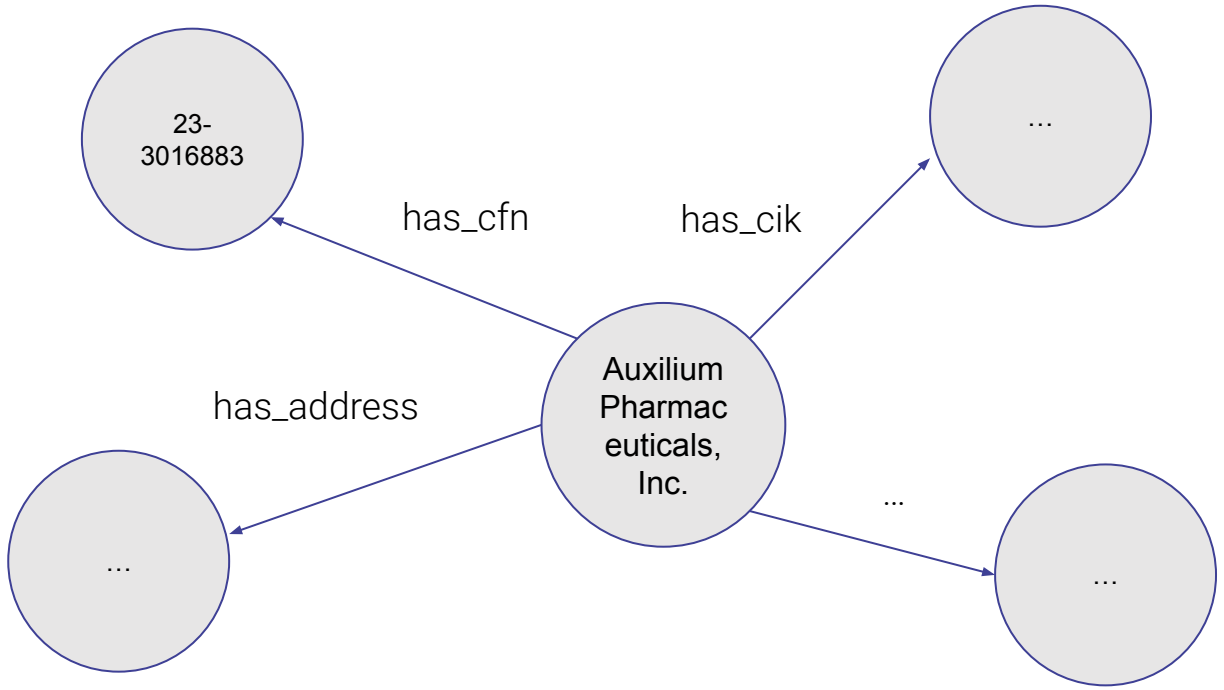
Data augmentation with Chunk Mappers

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**.**

Example**:**




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



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.

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

Graph Creation

Legal NLP

Table of Contents

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	**00-0000000**
(State or Other Jurisdiction of Incorporation or Organization)	(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:

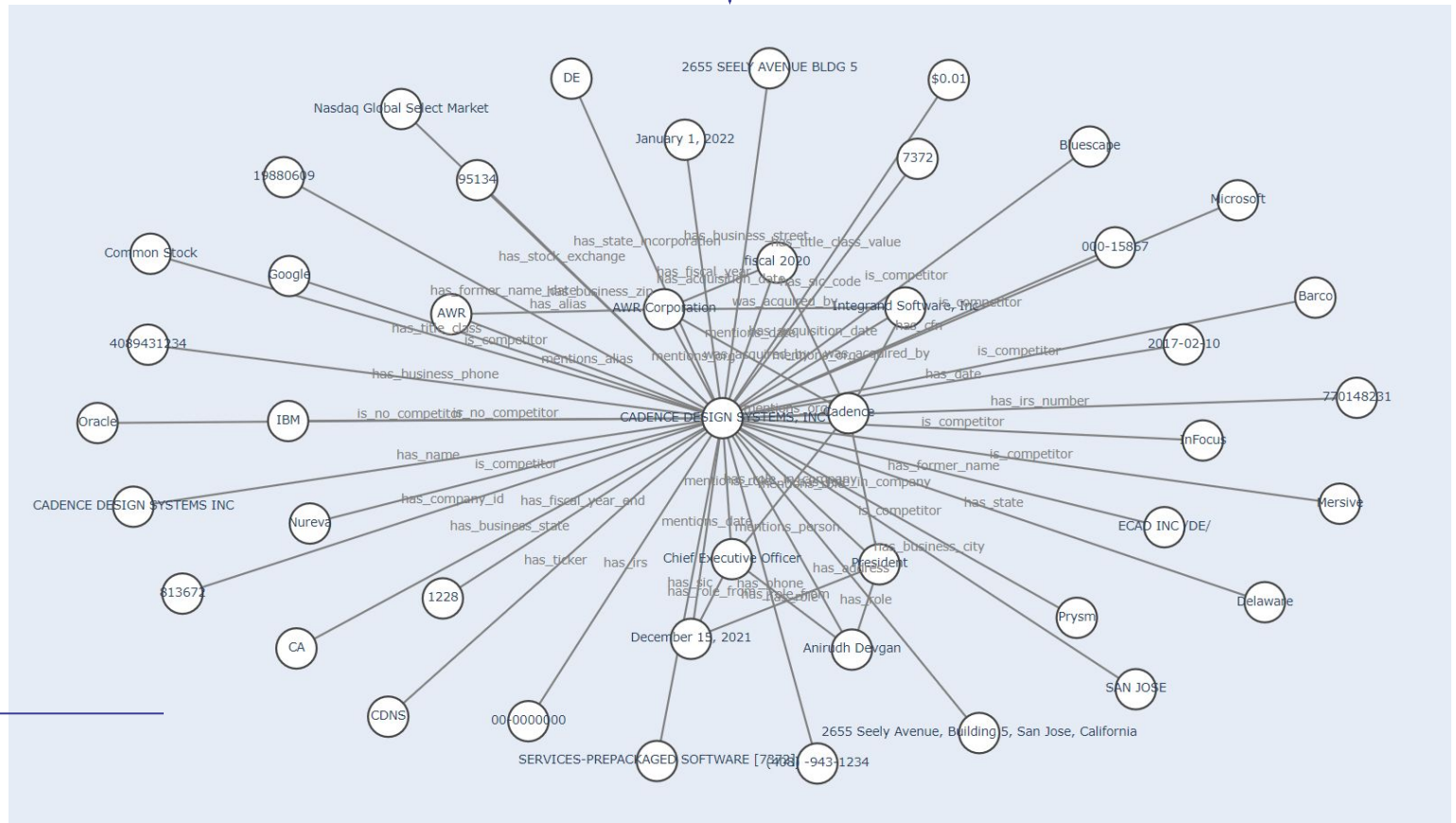
Title of Each Class	Trading Symbol(s)	Names of Each Exchange on which Registered
Common Stock, $0.01 par value per share	**CDNS**	**Nasdaq Global Select Market**

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), and (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	☐

+ 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



Graph Embeddings for company similarity, link prediction, etc?

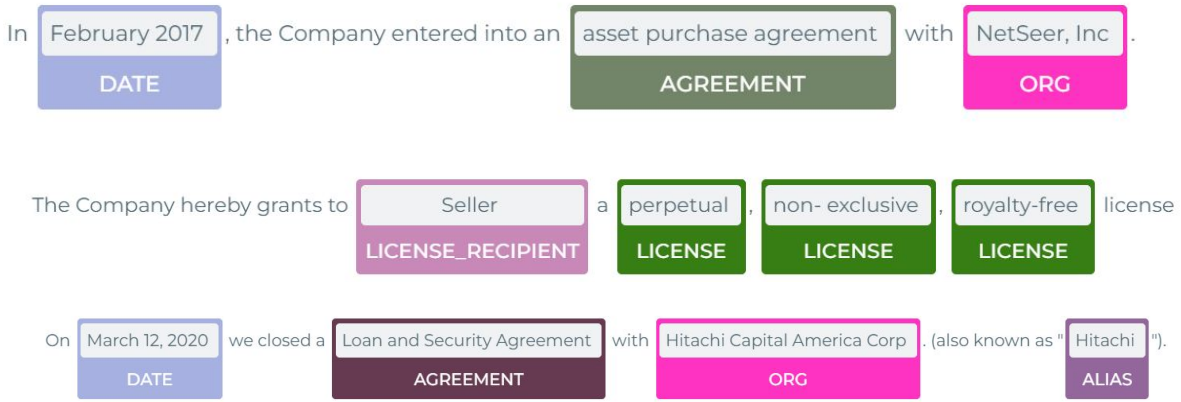
STATE OF THE ART

Question&Answering

Legal NLP

Legal NLP Question Answering

	Entity	Question
0	DATE	['When was the company acquisition?', 'When was the company purchase agreement?'
1	ORG	['Which company?', 'Which was the company acquisition?']
2	STATE	['Which state?']
3	AGREEMENT	['What kind of agreement?']
4	LICENSE	['What kind of license?']
5	LICENSE_REC	['To whom the license is granted?']
6	ALIAS	['Which is the alias?']



Legal 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 February 2017, the Company entered into an asset purchase agreement with NetSeer, Inc.	The Agreement is an Asset Purchase Agreement.	Entailment
	The Company entered into agreement in March 2020.	Contradiction
	The Company is John Snow Labs, Inc.	Neutral

	Entity	Question
0	DATE	['When was the company acquisition?', 'When was the company purchase agreement?'
1	ORG	['Which company?', 'Which was the company acquisition?']
2	STATE	['Which state?']
3	AGREEMENT	['What kind of agreement?']
4	LICENSE	['What kind of license?']
5	LICENSE_REC	['To whom the license is granted?']
6	ALIAS	['Which is the alias?']

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

Prompt Generation

Legal NLP

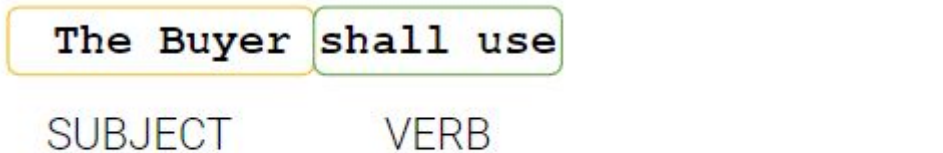
Legal NLP NER QA-based: Automatic Question Generation

One of the main restrictions of Zero-shot NER or QA is that you need the question before hand. Fortunately, there is a way you can generate **questions (prompt) on the fly.**

1. Imagine you want to extract:

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

2. ***NerQuestionGenerator*** annotator can generate questions for you if you have an NER with retrieves **SUBJECT** and **VERB**. which is much easier to train (for example, **legner_obligations**)




```
qagenerator = legal.NerQuestionGenerator()\
  .setInputCols(["ner_chunk"])\
  .setOutputCol("question")\
  .setQuestionMark(False)\
  .setQuestionPronoun("What")\
  .setEntities1(["OBLIGATION_SUBJECT"])\
  .setEntities2(["OBLIGATION_ACTION"])
```

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

we send it to QA

```
qa =nlp.BertForQuestionAnswering.pretrained("legqa_bert_large","en", "legal/models")
  .setInputCols(["question", "document"]) \
  .setOutputCol("answer") \
  .setCaseSensitive(True)
```

`such materials and supplies only in accordance with the present agreement`

Annotators

- **NerQuestionGenerator**: Takes the output of 2 NER models and creates prompts to be used in Question Answering or even Zero-shot NER.

STATE OF THE ART

Deidentification

Legal NLP

De-Identification

John Snow LABS

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

AGREEMENT NUMBER: 1234567

26 Mar 2022

IN WITNESS WHEREOF, the Parties have duly executed this Agreement as of the date first written above. ARMSTRONG FLOORING, INC.

By: /s/

Donald R. Maier Title: President and Chief Executive Officer

Detect sensitive entities

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

AGREEMENT NUMBER: 1234567

26 Mar 2022

IN WITNESS WHEREOF, the Parties have duly executed this Agreement as of the date first written above. ARMSTRONG FLOORING, INC.

By: /s/

Donald R. Maier Title: President and Chief Executive Officer

Transform sensitive entities

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

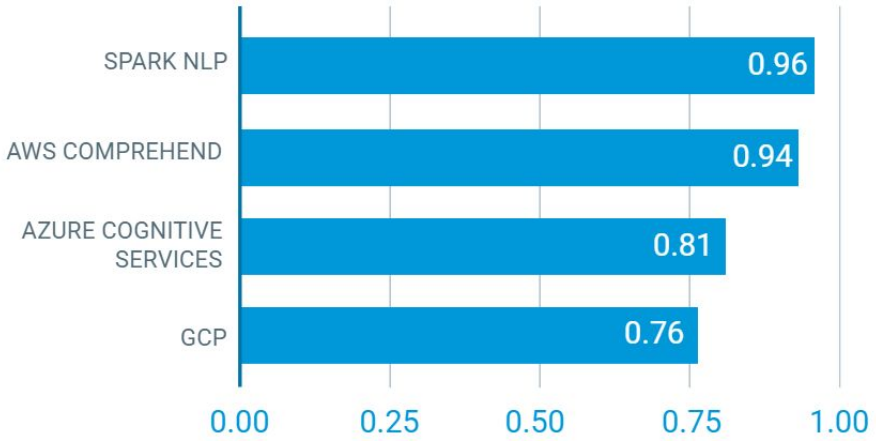
AGREEMENT NUMBER: 1234567

16 Jan 2019

IN WITNESS WHEREOF, the Parties have duly executed this Agreement as of the date first written above. AUXILIUM, INC.

By: /s/

David Bill Title: Director and Chief Accounting Officer



Legal NLP Deidentification

John Snow LABS

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 llc., 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 : \nphone : 0	Name: [**********]: [*******]Date: [**********]\n\nemail : [**********]\...	Name: ****: ****Date: ****\n\nemail : ****\nphone : ****0	Name: MGT Trust Company, LLC.: Business ManagerDate: 11/7/2016\n\nemail : Berneta@hotmail.com\...

Annotators

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

John Snow LABS

STATE OF THE ART

Text and Layout information Legal NLP and Visual NLP

Visual classification

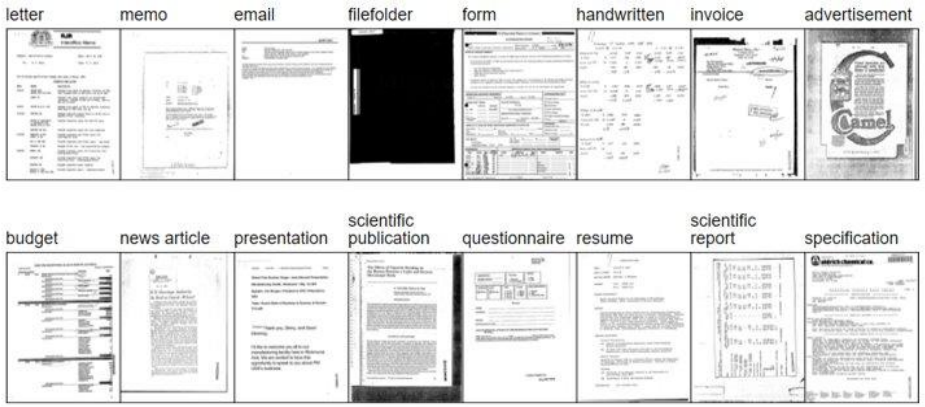
Sometimes textual information is not enough to classify a document. For example, let's suppose you have to classify 2 types of document with the same content, but only differing in the layout disposition of the information.

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

Legal 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

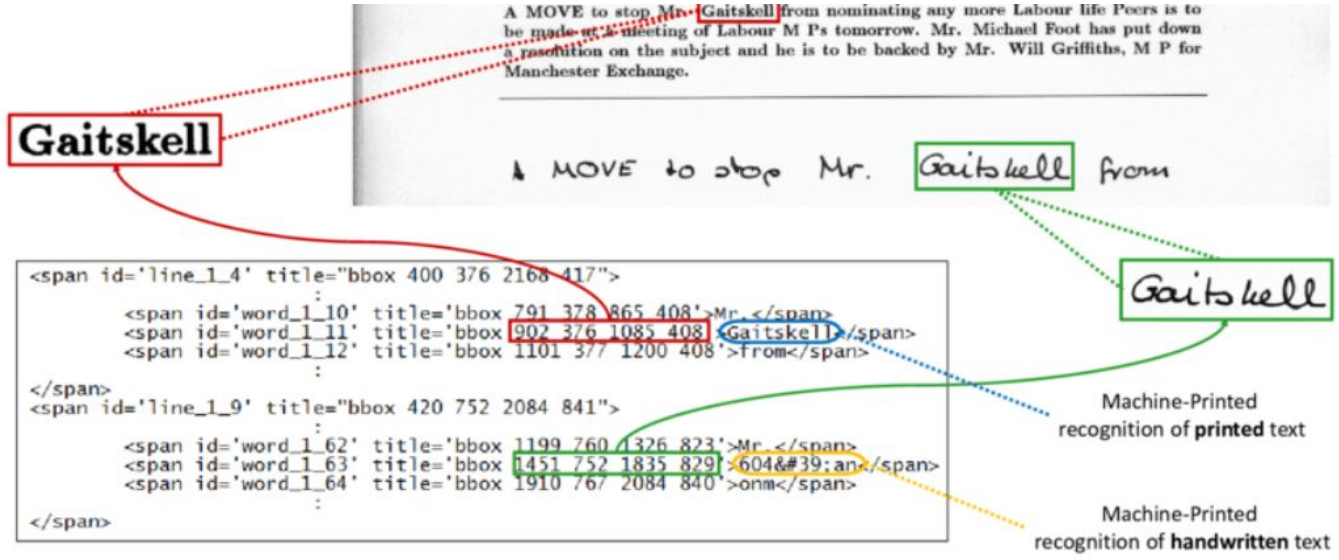
Visual classification

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, Legal NLP by itself 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.



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

Other Visual NLP capabilities

John Snow LABS

Document Classification

Classified Image

Classification

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

Classification Confidence: **99.6%**

From images, pdf, docx, ppt...

... to plain text

EXHIBIT 10.16

DRAFT (Americas) 1/12/00 (Rev 1) SUPPORT AND MAINTENANCE AGREEMENT

This Support and Maintenance Agreement ("Agreement") is entered into and is effective as of the ____ day of _____2000 (the "Effective Date") by and between XACCT Technologies, Inc., a Delaware corporation ("XACCT") with its principal place of business at 2900 Lakeside Drive, Suite 100, Santa Clara, California 95054 and _____, a _____corporation ("Licensee") with its principal place of business at _____.

Object detection

Thank you!