John Snow LABS

STATE OF THE ART

Legal NLP

STATE OF THE ART

Introduction

Legal 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

DocuSign

A unified CV, OCR, and NLP approach for scalable document understanding at DocuSign

Roche

Accelerating Biomedical Innovation by Combining NLP and Knowledge Graphs

GE Healthcare

Extracting what, when, why, and how from Radiology Reports in Real World Data Projects

NOVARTIS

Automated Classification and Entity Extraction from Essential Clinical Trial Documents

UiPath™

Spark NLP in action: intelligent, high-accuracy fact extraction from long financial documents

BITVORE

Text Classification into a Hierarchical Market Taxonomy using Spark NLP at Bitvore

OMNY HEALTH

Adverse Drug Event Detection using Spark NLP

Providence

Lessons Learned De-Identifying 700 Million Patients Notes with Spark NLP

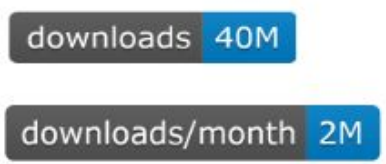
Pacific Northwest NATIONAL LABORATORY

Applying Spark NLP to Develop Multi-Modal Prediction Models from EHR Records

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

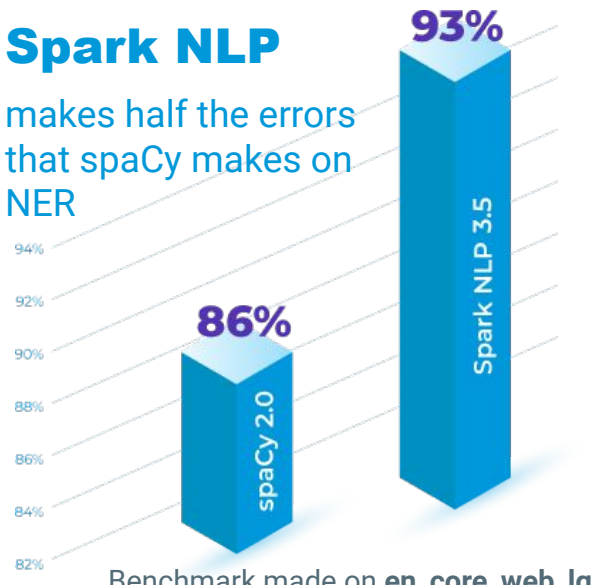
Trainable & Tunable	Scalable to a Cluster	Fast Inference	Hardware Optimized	Community
	Apache Spark ML Pipelines	LightPipeline	intel NVIDIA	NLP SUMMIT

Spark NLP

is natively scalable and production-ready

Spark NLP

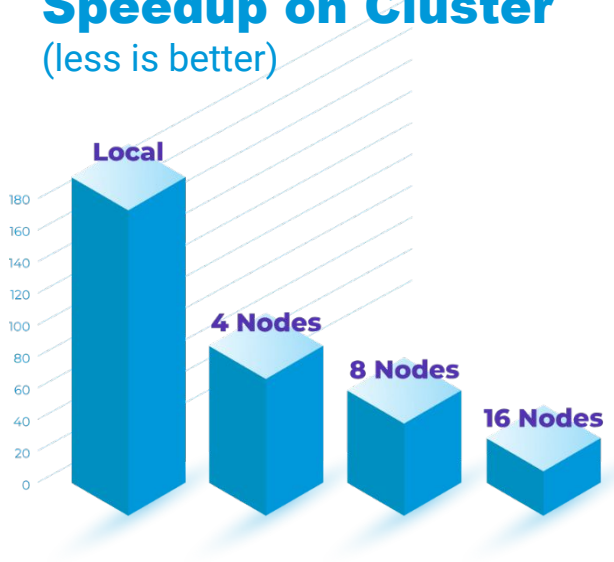
makes half the errors that spaCy makes on NER



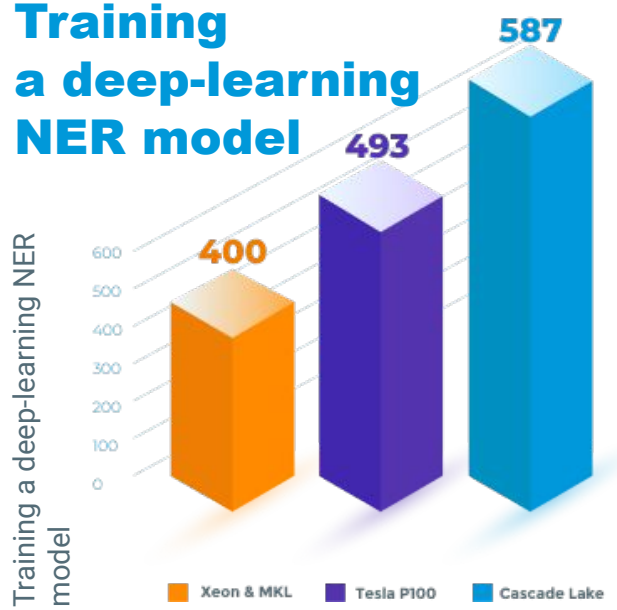
Benchmark made on **en_core_web_lg** dataset, micro-averaged F1 score

Speedup on Cluster

(less is better)



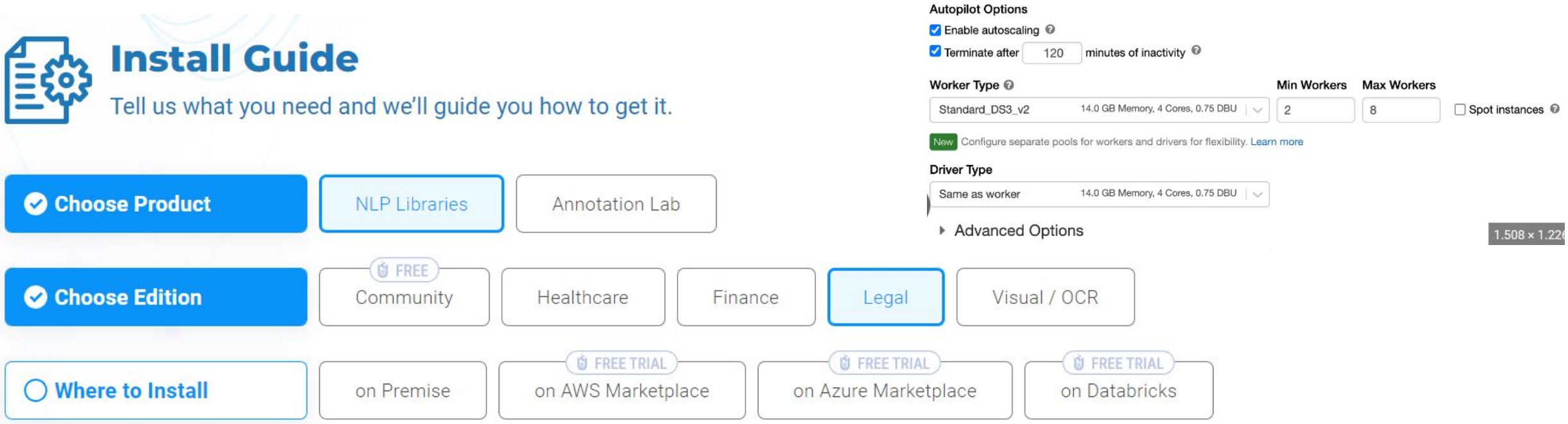
Training a deep-learning NER model



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

Legal NLP
Infrastructure

- 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



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

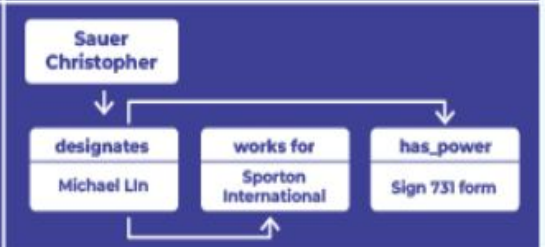
Assertion Status

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

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

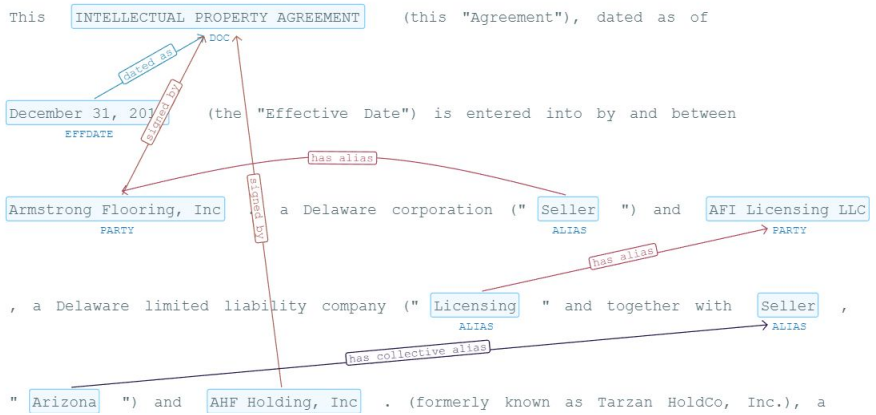
Splitting Legal texts

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



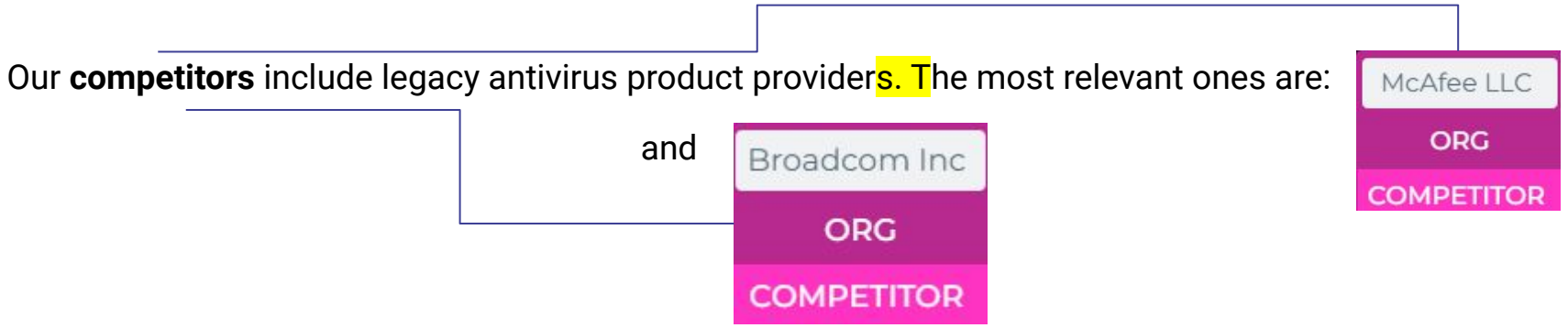
Splitting Legal texts

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
…

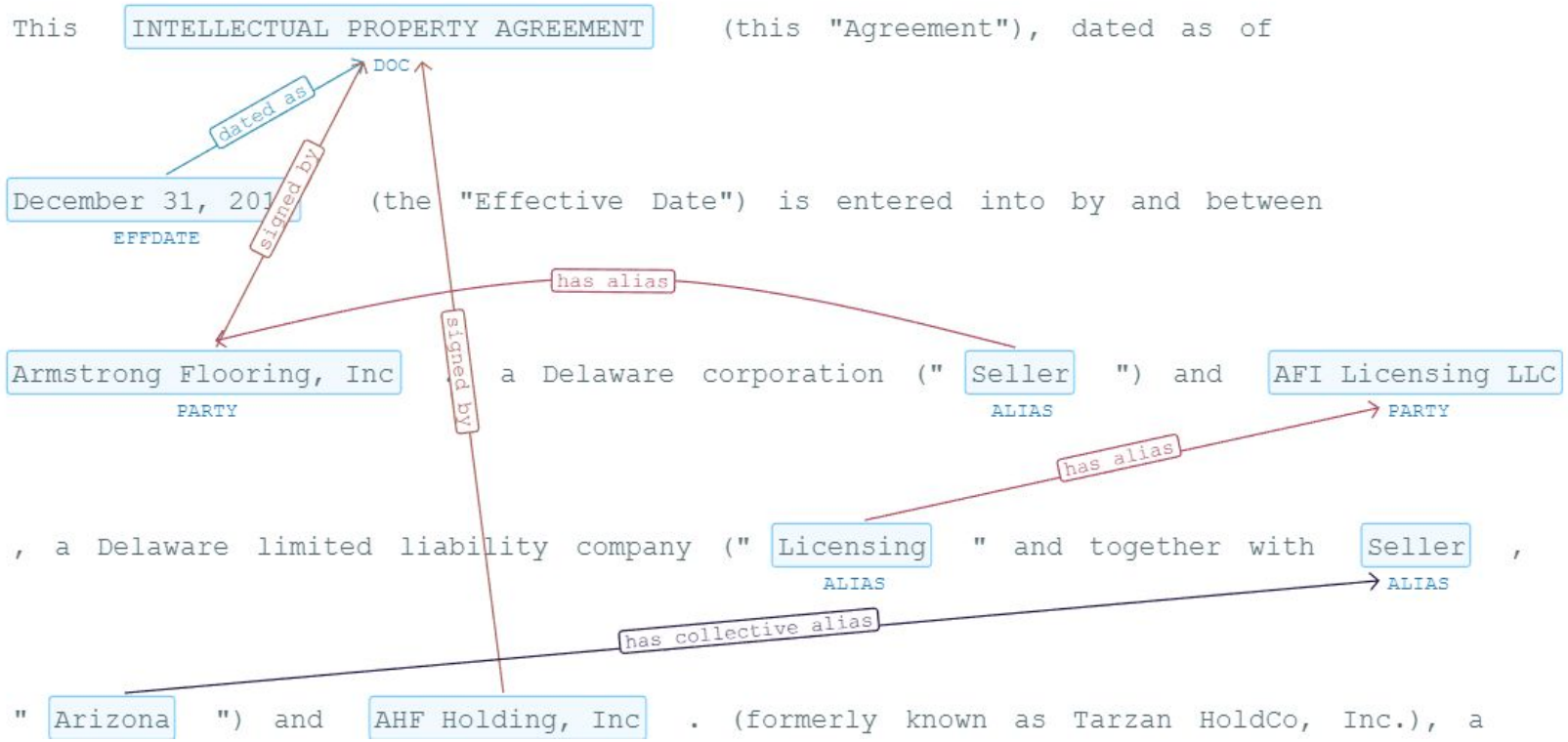
- For **Assertion**, as with Text Classification, you may want to send the model more than just a sentence.



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



John Snow LABS

STATE OF THE ART

Language Models

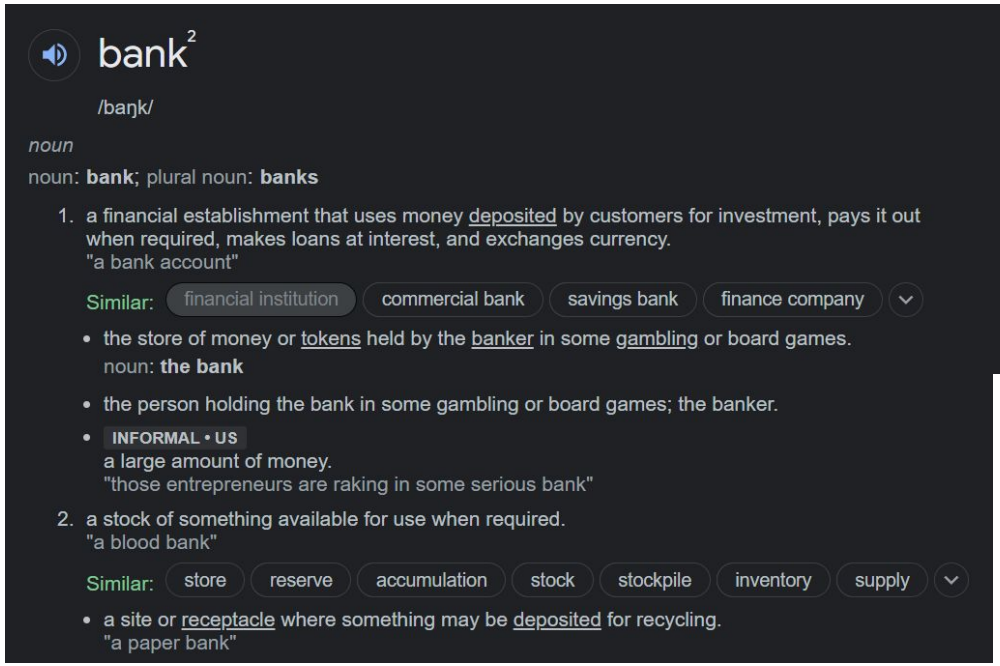
Legal NLP

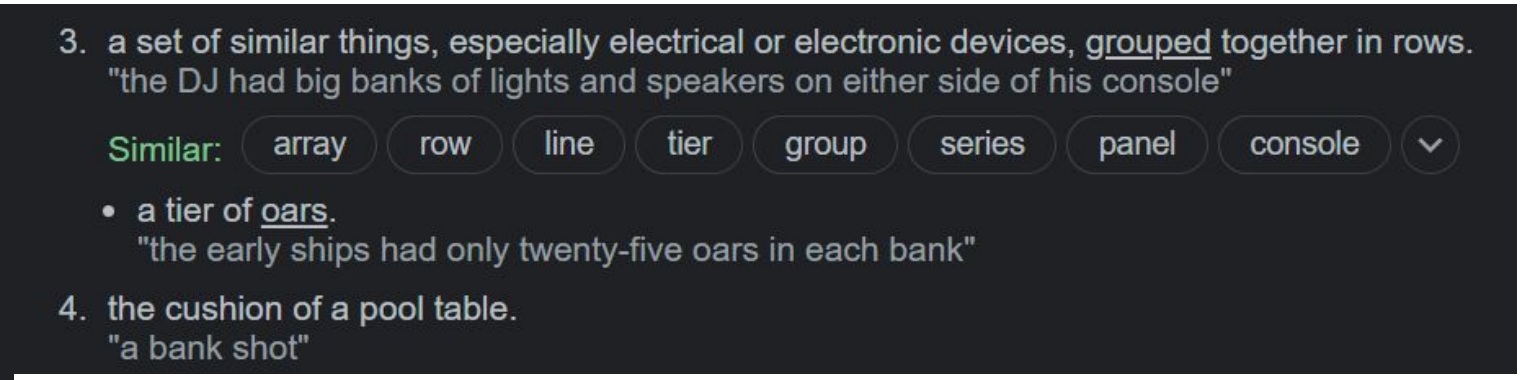
Language Models and Embeddings

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.



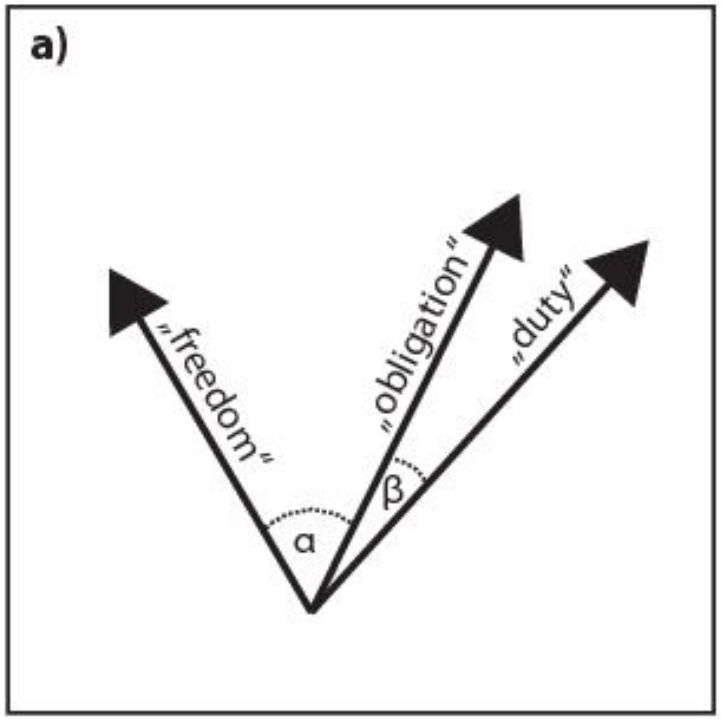


All of these will have different embeddings (numerical representations) in context!

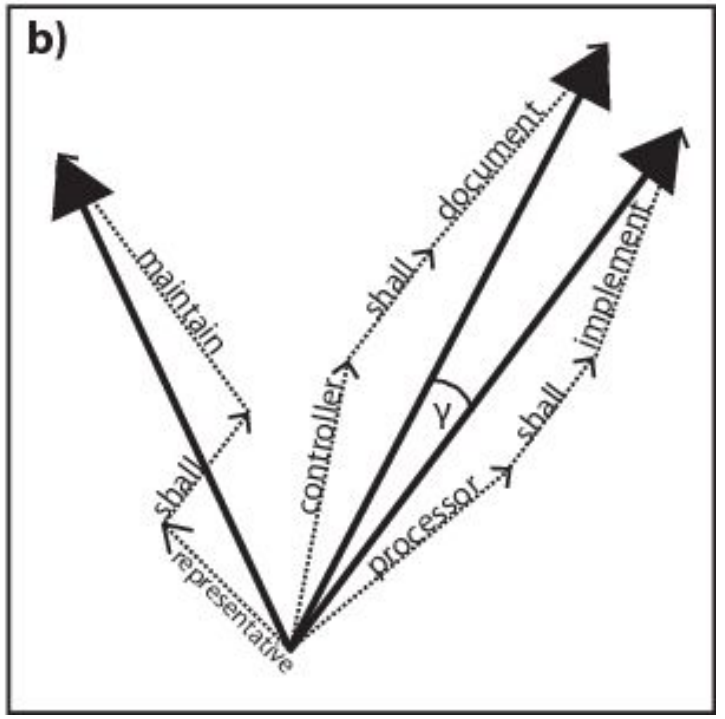
Language Models and Embeddings

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

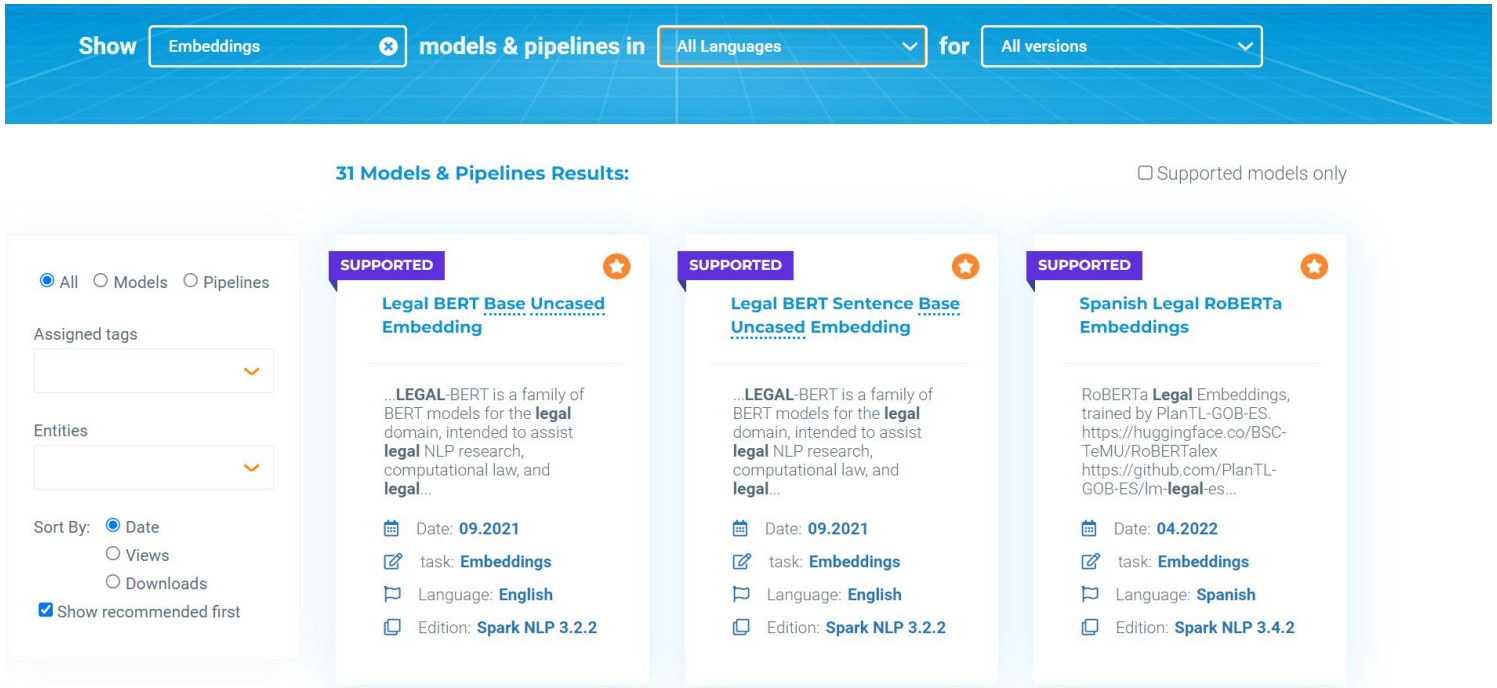
- As a consequence of their context-specificity, it's very important you use domain specific embeddings. Fortunately, we have **more than 30** Legal NLP Language Models in Models Hub, including English, Portuguese and Spanish.

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.



STATE OF THE ART

Text 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

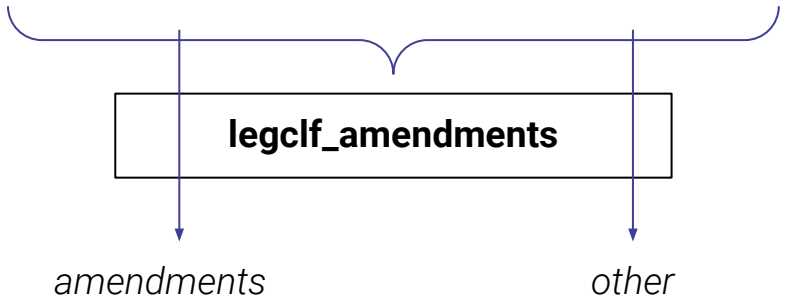
- By **output** type or **class assigned to the input text**

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

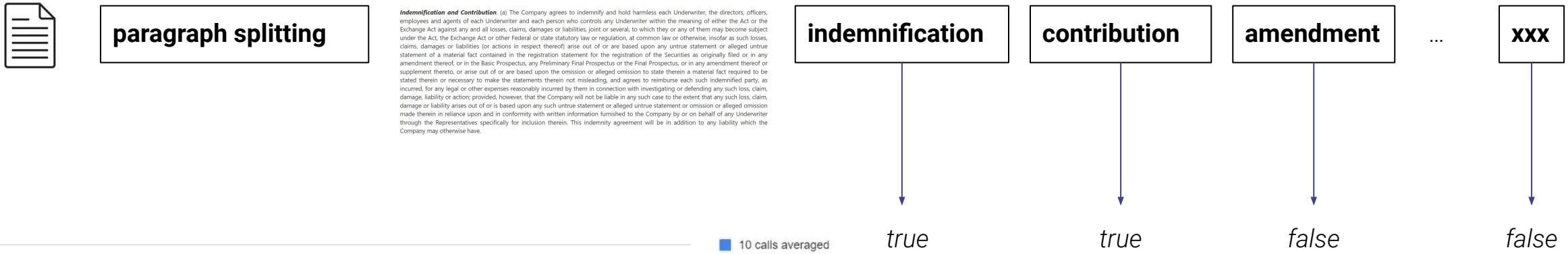
amendment - FALSE

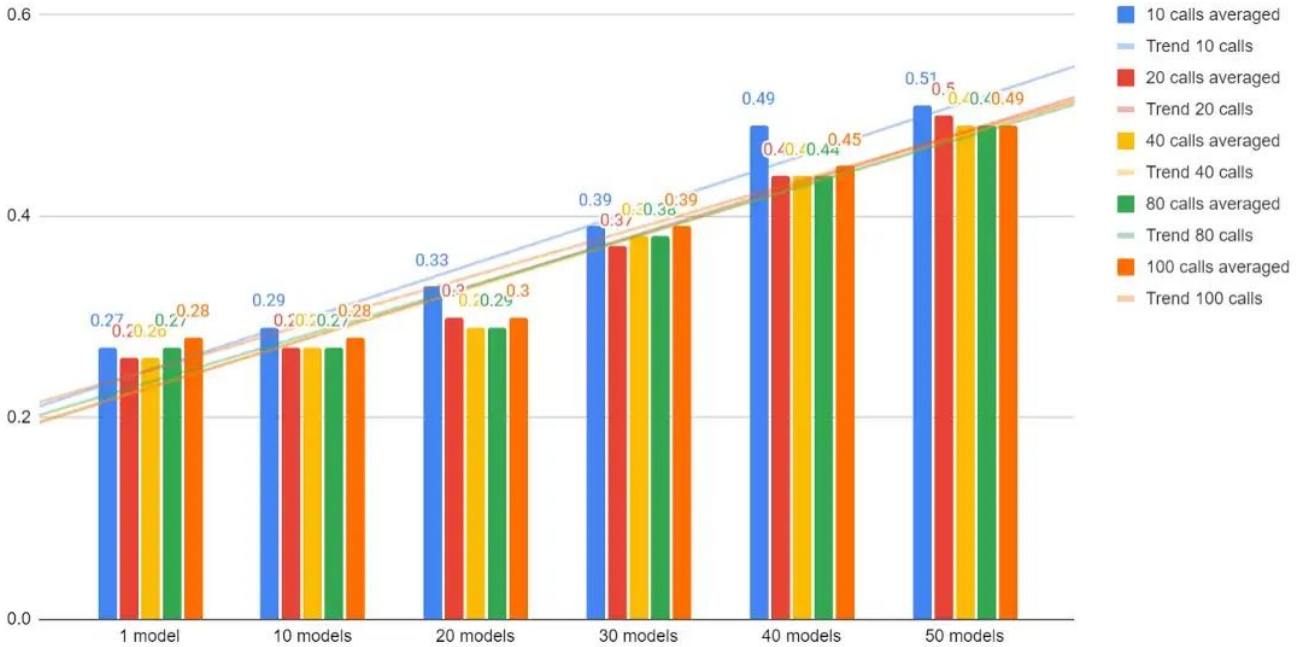
...

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

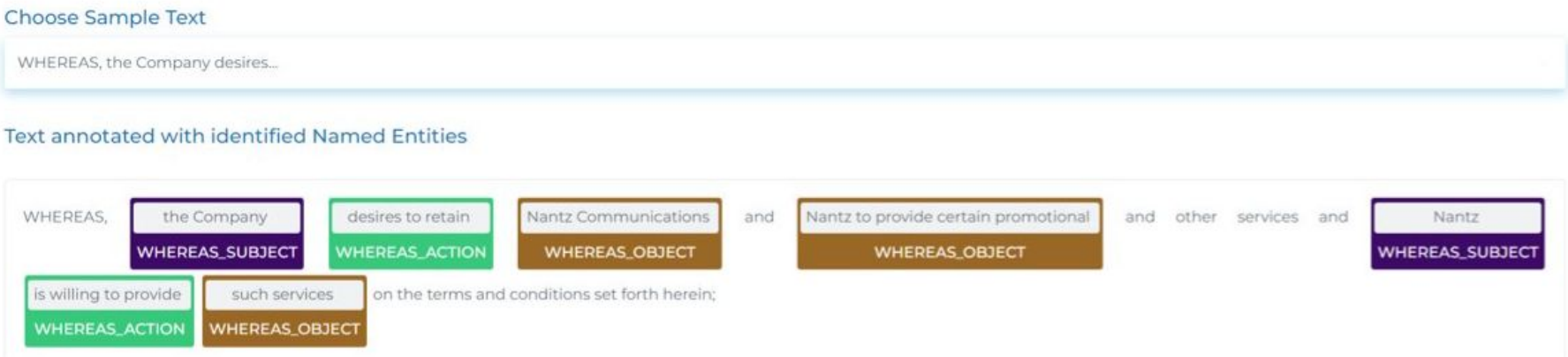
STATE OF THE ART

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

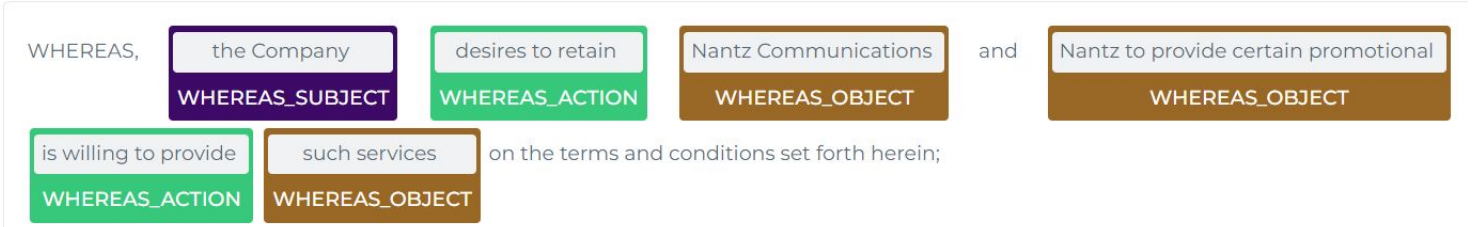
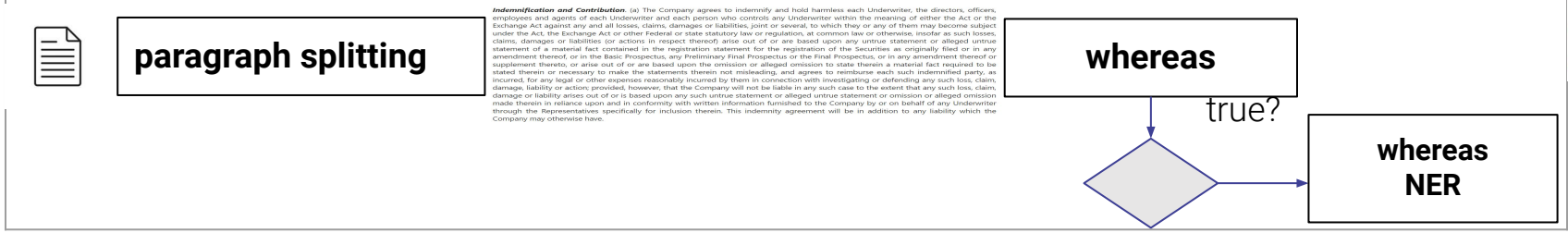
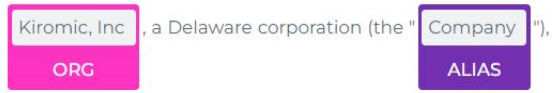
NER is the NLP task in charge of detecting relevant words / chunks in texts and categorize them.

- **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* *…*

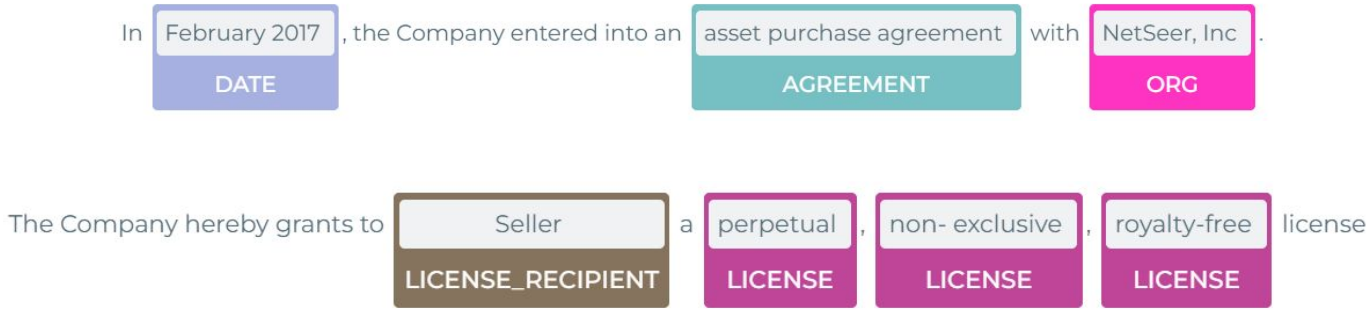
Legal NLP Named Entity Recognition

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*.

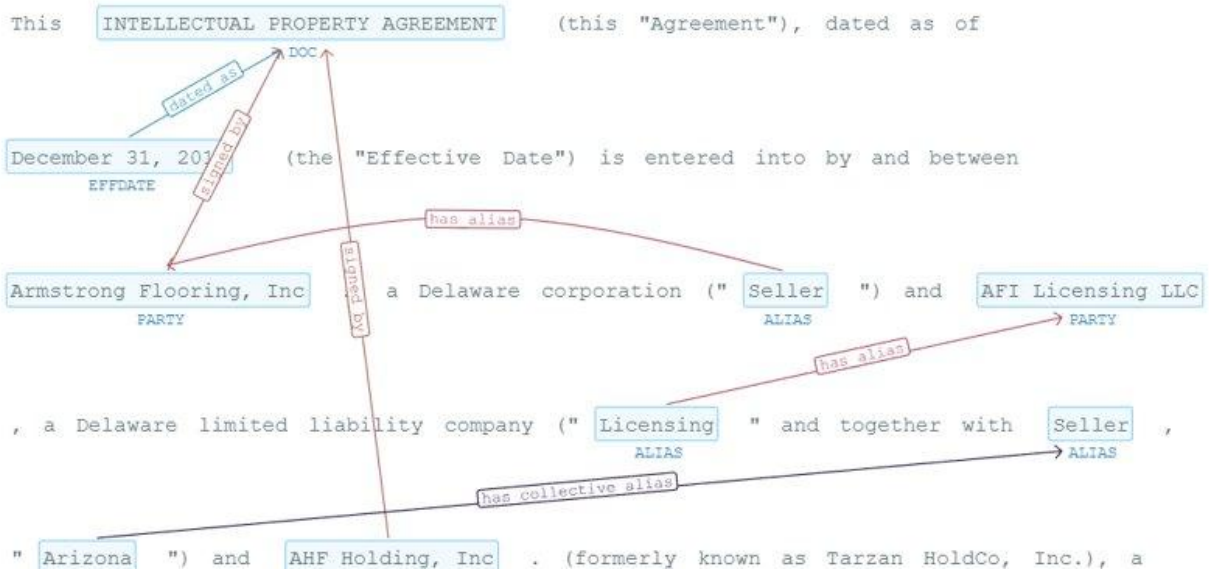
Using our **ZeroShotNER** annotator, those *questions (prompts)* can be asked to our NLI-based language model, and *retrieve the answers* in form of *predictions*, without a training step. And most importantly, **without any training data required.**

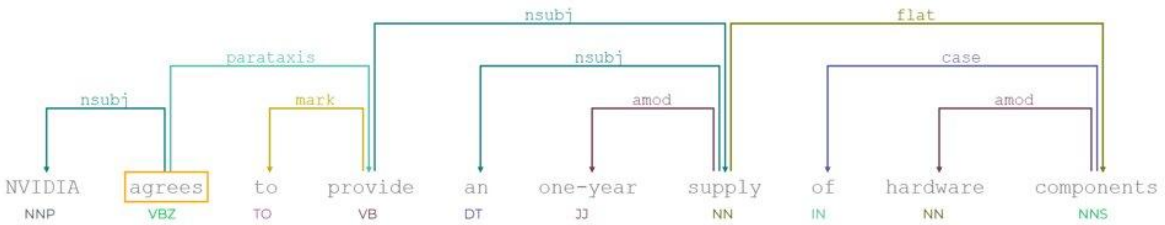
STATE OF THE ART

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.

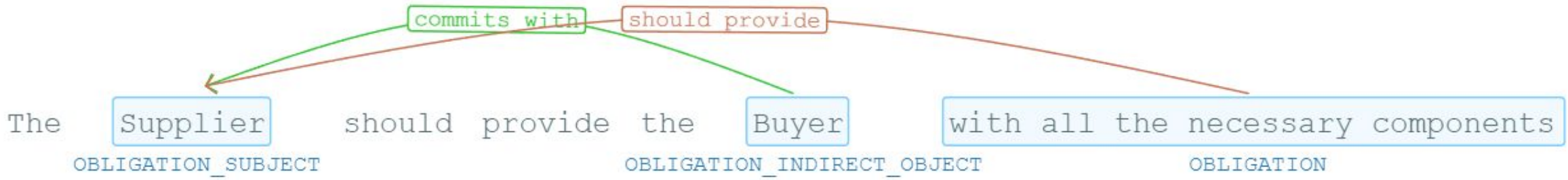
Zero-shot Relation Extraction

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.



John Snow LABS

STATE OF THE ART

Assertion Status

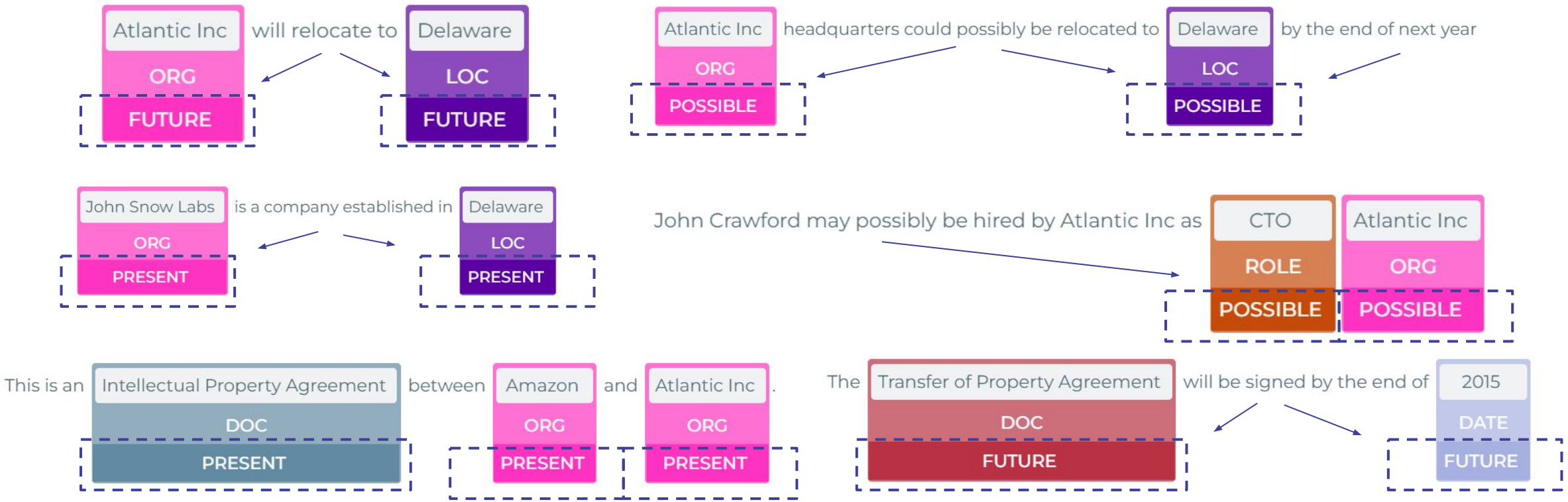
Legal NLP

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).**



John Snow LABS

STATE OF THE ART

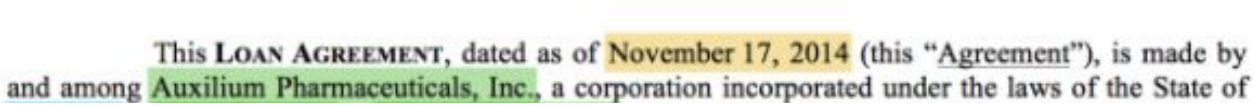
Entity Resolution

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**

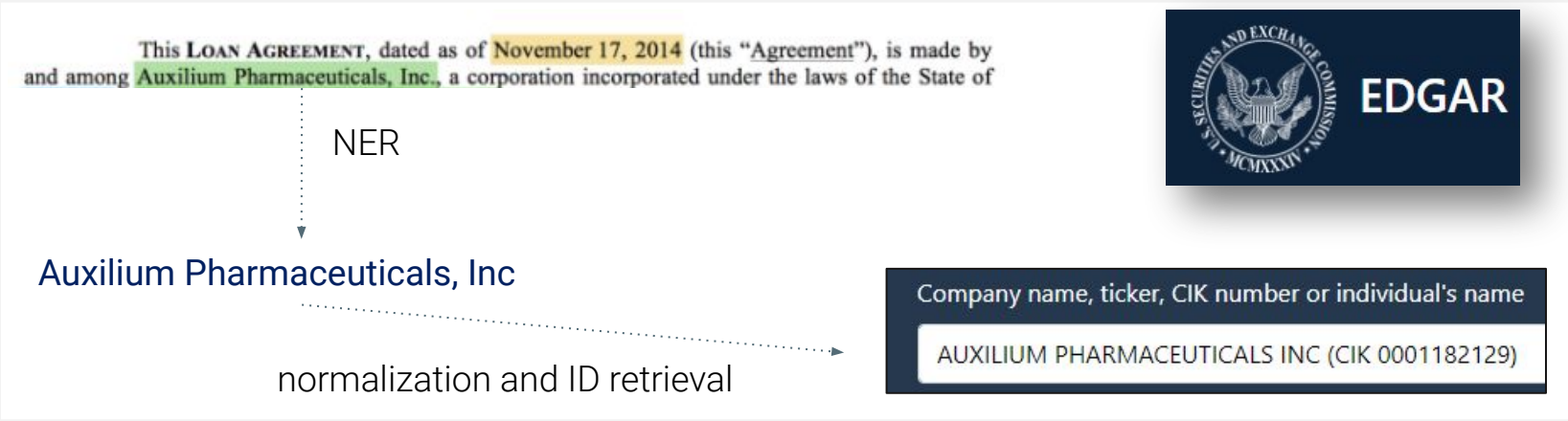
Entity Resolution

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).**



STATE OF THE ART

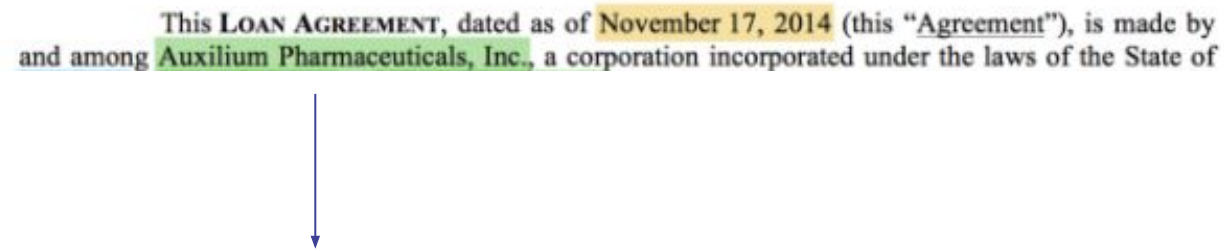
Data Augmenation with Chunk Mappers

Legal NLP

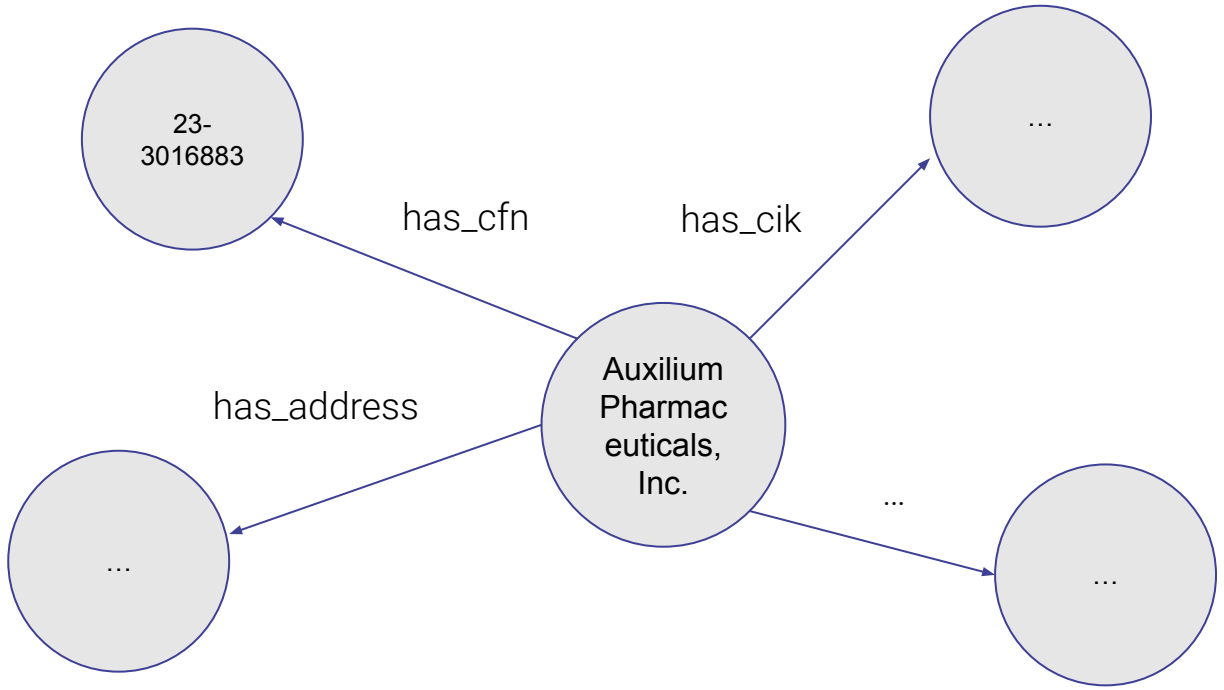
Data augmentation with Chunk Mappers

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.

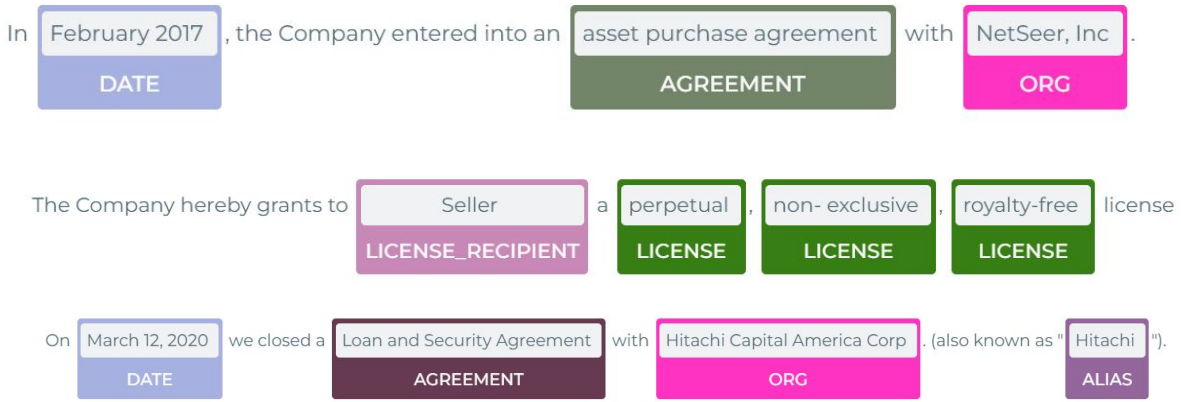
STATE OF THE ART

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:**

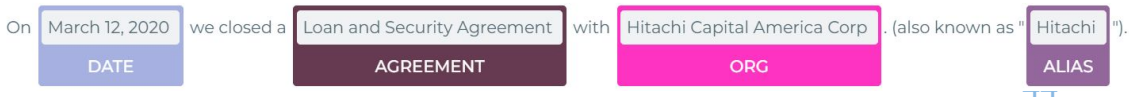
- **Open book**: We provide also with a context where to look.
- **Closed book**: The knowledge is stored in the Language Model and you don't give any example.

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?']



STATE OF THE ART

Automatic Question Generation

Legal NLP

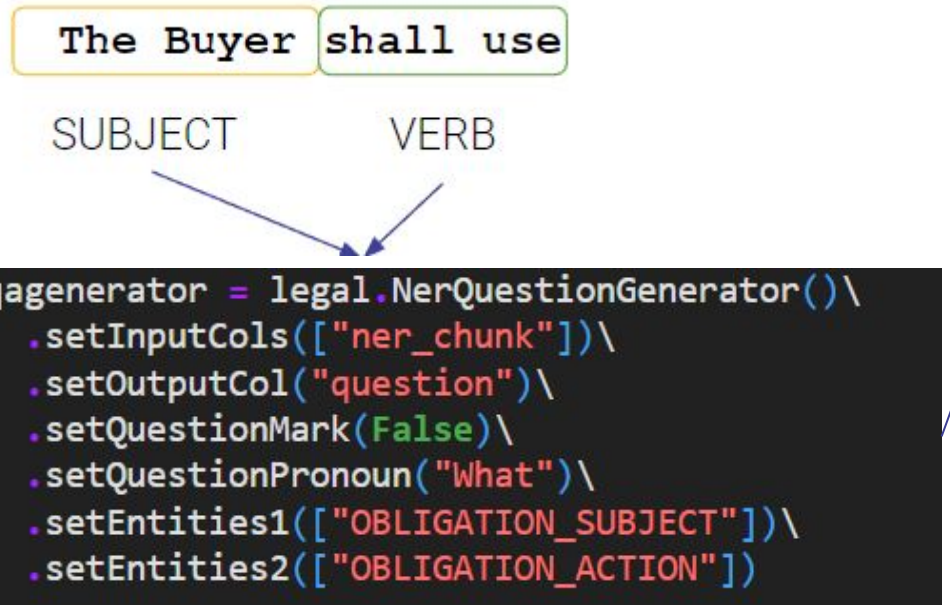
Legal NLP NER QA-based: Automatic Question Generation

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**)




```
+----------------------+
|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`

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

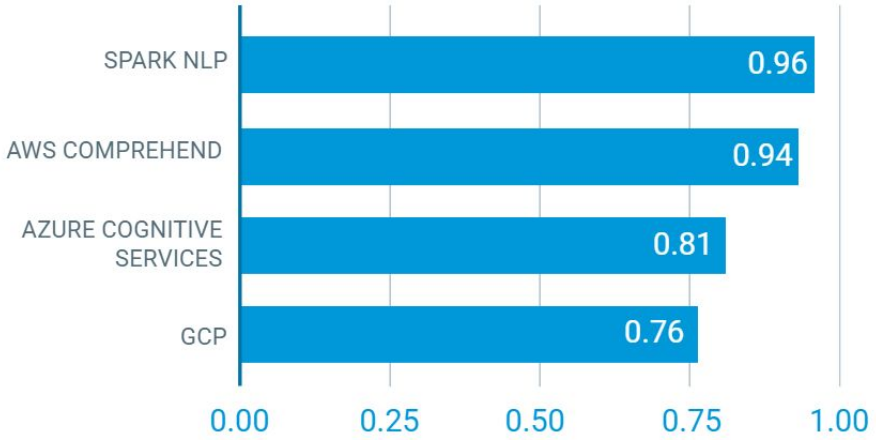
AGREEMENT NUMBER: 1234567

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\...

John Snow LABS

STATE OF THE ART

Text and Layout information Legal NLP and Visual NLP

Visual classification

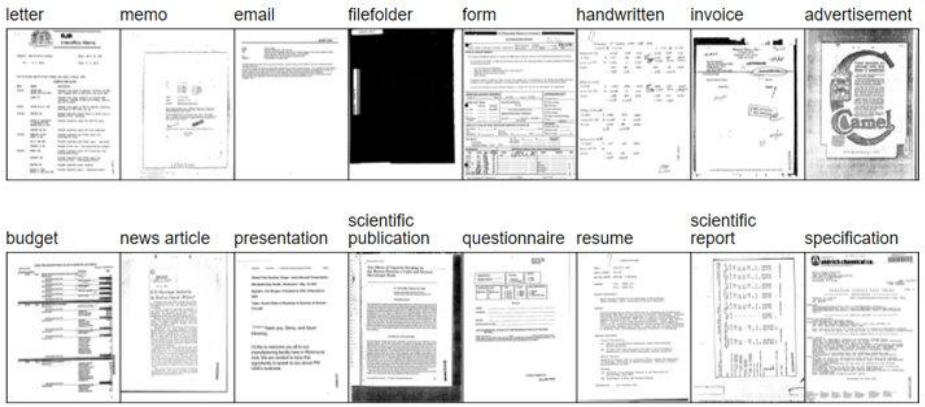
Sometimes textual information is not enough to classify a document. For example, let's suppose you have to classify 2 types of document with the same content, but only differing in the layout disposition of the information.

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

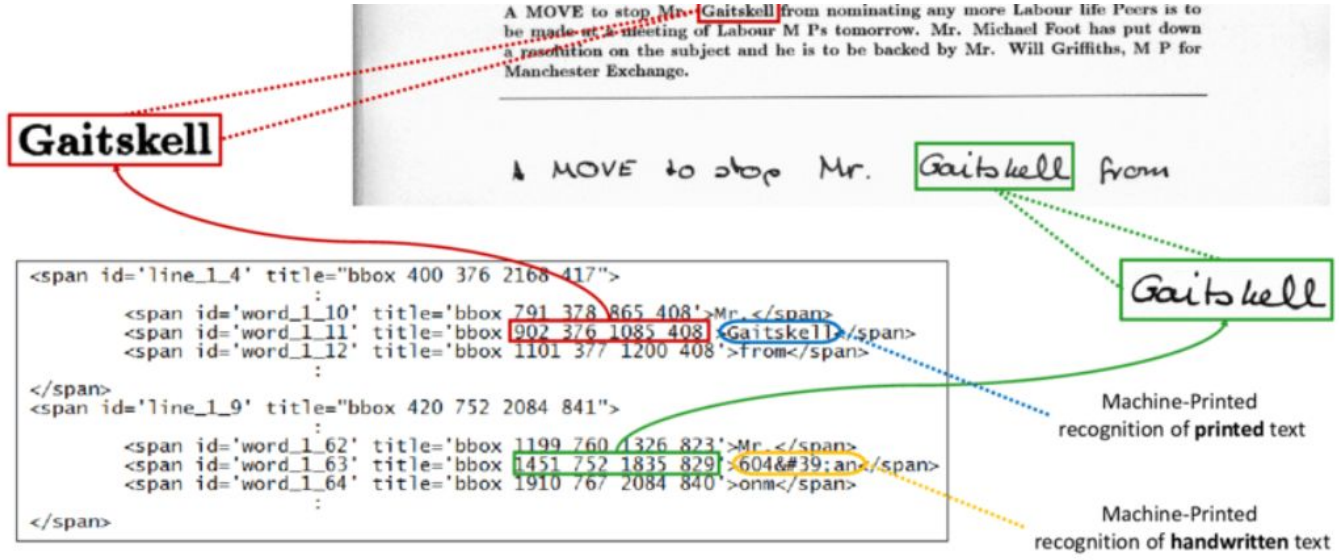
Visual classification

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!