

# Spark NLP for Data Scientists

Training & Certification

Day 1: Jan 16, 2023 – 9:00 to 13:00 PST

Day 2: Jan 17, 2023 – 9:00 to 13:00 PST

Training Location: Online

Price: \$395

4 days, 16 hrs live training  
State-Of-The-Art NLP

Certification exams in 2 weeks

# Healthcare NLP for Data Scientists

Training & Certification

Day 1: Jan 18, 2023 – 9:00 to 13:00 PST

Day 2: Jan 19, 2023 – 9:00 to 13:00 PST

Training Location: Online

Price: \$395



---

# Spark NLP for Data Scientists

Jan 16 & 17, 2023

Luca Martial  
[lucas@johnsnowlabs.com](mailto:lucas@johnsnowlabs.com)

Gursev Pirge  
[gursev@johnsnowlabs.com](mailto:gursev@johnsnowlabs.com)

David Amore Cecchini  
[cecchini@johnsnowlabs.com](mailto:cecchini@johnsnowlabs.com)

Christian Kasim Loan  
[christian@johnsnowlabs.com](mailto:christian@johnsnowlabs.com)

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

Day-1	50 min	<ul style="list-style-type: none"><li>- Intro to John Snow Labs, Spark NLP and NLP Theory</li><li>- Spark NLP Pretrained Pipelines basics and JVM/Spark concepts</li></ul>
	10 min	Break
	50 min	<ul style="list-style-type: none"><li>- Text Preprocessing and composing custom pipeline in Spark NLP</li></ul>
	10 min	Break
	50 min	<ul style="list-style-type: none"><li>- Showcase of the 10000+ pretrained Models for 300+ languages in Spark NLP</li></ul>
		Break
	50 min	<ul style="list-style-type: none"><li>- Train Named Entity Recognizers (NER) and Classifier models</li><li>- Upload to Modelshub</li></ul>

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

Day-2	50 min	<ul style="list-style-type: none"><li>- Token Classification with Transformers</li><li>- Sequence Classification with Transformers</li><li>- Image Classification with ViT</li><li>- Speech to Text with Wav2Vec2</li></ul>
	10 min	Break
	50 min	<ul style="list-style-type: none"><li>- Spell Checking</li><li>- Unsupervised Keyword Extraction with YAKE</li><li>- Rule-based Entity Recognition with EntityRuler</li><li>- Graph triplet extraction</li></ul>
	10 min	Break
	50 min	<ul style="list-style-type: none"><li>- Table Question Answering with TAPAS</li><li>- Question Answering, Summarization and other T5 applications</li><li>- Multilingual NLP - Train only on English data and predict for 100+ languages</li></ul>
	10 min	Break
	50 min	<ul style="list-style-type: none"><li>- The Python NLU library basics, use any of the 7500+ models in 1 line of code</li><li>- NLP Server</li></ul>

# Session 1 (Day 1)

- ❖ NLP Theory and Introduction to John Snow Labs
- ❖ Pretrained Pipelines
- ❖ Spark, JVM and Spark NLP basic concepts
- ❖ [Notebook 1 - Spark NLP Basics](#)

Spark NLP  
for Data Scientists



# John Snow Labs NLP Documentation



Spark NLP



Healthcare NLP  
Legal NLP  
Finance NLP



Visual NLP



NLP Lab



NLP Server



John Snow Labs NLP

# Resources

## CODE:

- [https://github.com/JohnSnowLabs/spark-nlp-works  
hop/blob/master/tutorials/Certification\\_Trainings/P  
ublic](https://github.com/JohnSnowLabs/spark-nlp-works hop/blob/master/tutorials/Certification_Trainings/Public)

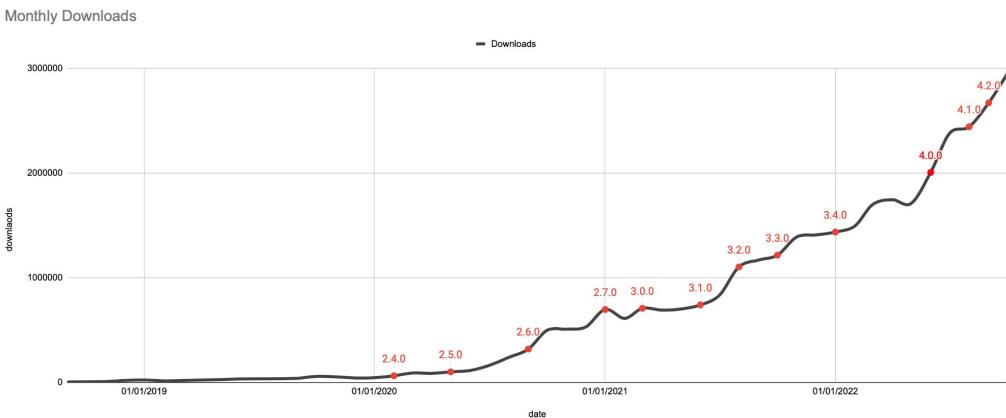
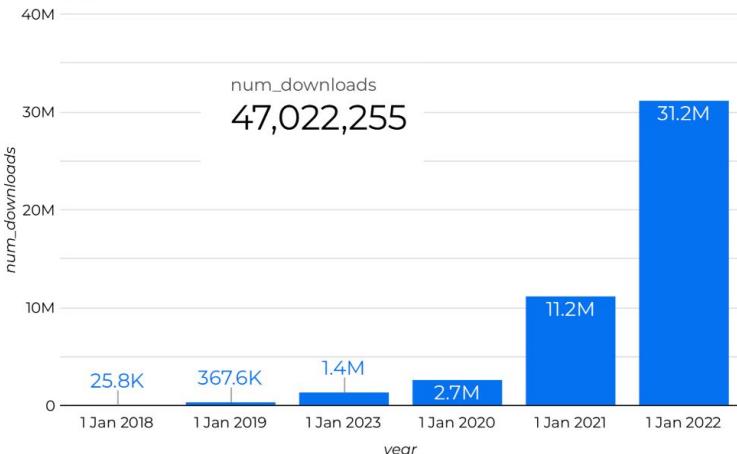
## BOOKMARK:

- <https://nlp.johnsnowlabs.com/>
- <https://nlp.johnsnowlabs.com/docs/en/quickstart>
- [https://nlp.johnsnowlabs.com/api/python/reference  
/index.html](https://nlp.johnsnowlabs.com/api/python/reference/index.html)
- <spark-nlp.slack.com>

The screenshot shows a GitHub repository interface. At the top, there's a 'master' branch dropdown, a 'spark-nlp-workshop / tutorials / Certification\_Trainings / Public /' path bar, and buttons for 'Go to file', 'Add file', and more. A prominent 'Open in Colab' button is visible. Below the header, a commit by 'gokhanturer' is shown, fixing a typo in a notebook. The commit was made on Dec 5, 2022. The repository contains several subfolders: 'data', 'slides', and 'utils', along with notebooks like '1.SparkNLP\_Basics.ipynb' through '14.Transformers\_for\_Token\_Classific...'. Each item in the list has a timestamp indicating when it was last updated.

File/Folder	Description	Last Updated
data	Add files via upload	7 months ago
slides	added slides october 2022	3 months ago
utils	updated postman collection for NLP server	6 months ago
1.SparkNLP_Basics.ipynb	notebooks updated and new notebook added	3 months ago
10.1_T5_Workshop_with_Spark_NLP...	notebooks updated and new notebook added	3 months ago
10.2_SQL_Code_Generation_and_St...	notebooks updated and new notebook added	3 months ago
10.Question_Answering_and_Summ...	notebooks updated and new notebook added	3 months ago
11.Text_Similarities_and_dimension_...	notebooks updated and new notebook added	3 months ago
12.Graph_extraction.ipynb	notebooks updated and new notebook added	3 months ago
13.NLU_crashcourse_every_Spark_....	notebooks updated and new notebook added	3 months ago
14.Transformers_for_Token_Classific...	notebooks updated and new notebook added	3 months ago

# Introducing Spark NLP



**Spark NLP** is an open-source natural language processing library, built on top of **Apache Spark** and **Spark ML**. (first release: July 2017)

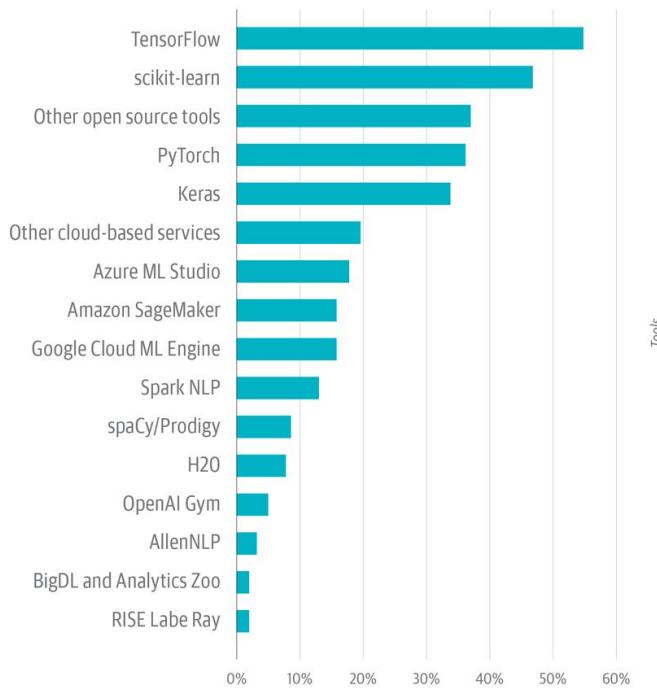
- A single unified solution for all your NLP needs
- Take advantage of transfer learning and implementing the **latest and greatest SOTA algorithms and models** in NLP research
- The most widely used NLP library in industry (5 yrs in a row)
- The most scalable, accurate and fastest library in NLP history
- 111 total releases, every two weeks for the past 5 years

<table border="1"> <thead> <tr> <th>Clinical Entity Recognition</th><th>Clinical Entity Linking</th><th>Assertion Status</th><th>Relation Extraction</th></tr> </thead> <tbody> <tr> <td>40 units <b>DOSAGE</b> of insulin glargine <b>DRUG</b> at night <b>FREQUENCY</b></td><td>Suspect diabetes SNOMED-CT: 473127005 Lisinopril 10 MG RxNorm: 316151 Hyponatremia ICD-10: E87.1</td><td>Fever and sore throat → PRESENT No stomach pain → ABSENT Father with Alzheimer → FAMILY</td><td> </td></tr> </tbody> </table>				Clinical Entity Recognition	Clinical Entity Linking	Assertion Status	Relation Extraction	40 units <b>DOSAGE</b> of insulin glargine <b>DRUG</b> at night <b>FREQUENCY</b>	Suspect diabetes SNOMED-CT: 473127005 Lisinopril 10 MG RxNorm: 316151 Hyponatremia ICD-10: E87.1	Fever and sore throat → PRESENT No stomach pain → ABSENT Father with Alzheimer → FAMILY		<table border="1"> <thead> <tr> <th>Entity Recognition</th><th>Information Extraction</th><th>Spelling &amp; Grammar</th><th>Text Classification</th><th>Image Classification</th></tr> </thead> <tbody> <tr> <td>I love Lucy PERSON</td><td>They met Last week DATE → 29-04-2020</td><td>abc She became the first... → She became the first</td><td></td><td></td></tr> <tr> <th>Translation</th><th>Summarization</th><th>Question Answering</th><th>Emotion Detection</th><th>Automatic Speech Recognition</th></tr> <tr> <td>[je t'aime → I love you]</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>					Entity Recognition	Information Extraction	Spelling & Grammar	Text Classification	Image Classification	I love Lucy PERSON	They met Last week DATE → 29-04-2020	abc She became the first... → She became the first			Translation	Summarization	Question Answering	Emotion Detection	Automatic Speech Recognition	[je t'aime → I love you]				
Clinical Entity Recognition	Clinical Entity Linking	Assertion Status	Relation Extraction																																	
40 units <b>DOSAGE</b> of insulin glargine <b>DRUG</b> at night <b>FREQUENCY</b>	Suspect diabetes SNOMED-CT: 473127005 Lisinopril 10 MG RxNorm: 316151 Hyponatremia ICD-10: E87.1	Fever and sore throat → PRESENT No stomach pain → ABSENT Father with Alzheimer → FAMILY																																		
Entity Recognition	Information Extraction	Spelling & Grammar	Text Classification	Image Classification																																
I love Lucy PERSON	They met Last week DATE → 29-04-2020	abc She became the first... → She became the first																																		
Translation	Summarization	Question Answering	Emotion Detection	Automatic Speech Recognition																																
[je t'aime → I love you]																																				
<p><b>Algorithms</b></p>				<p><b>Content</b></p>																																
<b>Extract Knowledge</b> <ul style="list-style-type: none"> <li>Entity Linker</li> <li>Entity Disambiguator</li> <li>Document Classifier</li> <li>Contextual Parser</li> </ul>	<b>De-identify text</b> <ul style="list-style-type: none"> <li>Structured Data</li> <li>Unstructured Text</li> <li>Obfuscator</li> <li>Generalizer</li> </ul>	<b>Medical Transformers</b> <table border="1"> <tr><td>JSL-BERT-Clinical</td><td>BioBERT</td></tr> <tr><td>ClinicalBERT</td><td>GloVe-Med</td></tr> <tr><td>GloVe-ICD-O</td><td>BlueBERT</td></tr> </table>	JSL-BERT-Clinical	BioBERT	ClinicalBERT	GloVe-Med	GloVe-ICD-O	BlueBERT	<b>Linked Medical Terminologies</b> <table border="1"> <tr><td>SNOMED-CT</td><td>CPT</td><td>UMLS</td></tr> <tr><td>ICD-10-CM</td><td>RxNorm</td><td>HPO</td></tr> <tr><td>ICD-10-PCS</td><td>ICD-O</td><td>LOINC</td></tr> </table>	SNOMED-CT	CPT	UMLS	ICD-10-CM	RxNorm	HPO	ICD-10-PCS	ICD-O	LOINC	<p><b>Split Text</b></p> <ul style="list-style-type: none"> <li>Sentence Detector</li> <li>Tokenizer</li> <li>Normalizer</li> <li>nGram Generator</li> <li>Word Segmentation</li> </ul> <p><b>Clean Text</b></p> <ul style="list-style-type: none"> <li>Spell Checker</li> <li>Grammar Checker</li> <li>Writing Style Checker</li> <li>Stopword Cleaner</li> <li>Summarization</li> </ul>																	
JSL-BERT-Clinical	BioBERT																																			
ClinicalBERT	GloVe-Med																																			
GloVe-ICD-O	BlueBERT																																			
SNOMED-CT	CPT	UMLS																																		
ICD-10-CM	RxNorm	HPO																																		
ICD-10-PCS	ICD-O	LOINC																																		
<p><b>Understand Grammar</b></p> <ul style="list-style-type: none"> <li>Stemmer</li> <li>Lemmatizer</li> <li>Part of Speech Tagger</li> <li>Dependency Parser</li> <li>Translation</li> </ul> <p><b>Find in Text</b></p> <ul style="list-style-type: none"> <li>Text Matcher</li> <li>Regex Matcher</li> <li>Date Matcher</li> <li>Chunker</li> <li>Question Answering</li> </ul>																																				
<p><b>Split Text</b></p> <ul style="list-style-type: none"> <li>Sentence Detector</li> <li>Deep Sentence Detector</li> <li>Tokenizer</li> <li>nGram Generator</li> </ul> <p><b>Clean Medical Text</b></p> <ul style="list-style-type: none"> <li>Spell Checking</li> <li>Spell Correction</li> <li>Normalizer</li> <li>Stopword Cleaner</li> </ul>				<p><b>800+ Pretrained Models</b></p> <table border="1"> <tr> <td><b>Clinical:</b> Signs, Symptoms, Treatments, Procedures, Tests, Labs, Sections</td> <td><b>Anatomy:</b> Organ, Subdivision, Cell, Structure, Organism, Tissue, Gene, Chemical</td> </tr> <tr> <td><b>Drugs:</b> Name, Dosage, Strength, Route, Duration, Frequency, Poisons, Adverse Effects</td> <td><b>Demographics:</b> Age, Gender, Height, Weight, Race, Ethnicity, Marital Status, Vital Signs</td> </tr> <tr> <td><b>Risk Factors:</b> Smoking, Obesity, Diabetes, Hypertension, Substance Abuse</td> <td><b>Sensitive Data:</b> Patient Name, Address, Phone, Email, Dates, Providers, Identifiers</td> </tr> </table>					<b>Clinical:</b> Signs, Symptoms, Treatments, Procedures, Tests, Labs, Sections	<b>Anatomy:</b> Organ, Subdivision, Cell, Structure, Organism, Tissue, Gene, Chemical	<b>Drugs:</b> Name, Dosage, Strength, Route, Duration, Frequency, Poisons, Adverse Effects	<b>Demographics:</b> Age, Gender, Height, Weight, Race, Ethnicity, Marital Status, Vital Signs	<b>Risk Factors:</b> Smoking, Obesity, Diabetes, Hypertension, Substance Abuse	<b>Sensitive Data:</b> Patient Name, Address, Phone, Email, Dates, Providers, Identifiers																						
<b>Clinical:</b> Signs, Symptoms, Treatments, Procedures, Tests, Labs, Sections	<b>Anatomy:</b> Organ, Subdivision, Cell, Structure, Organism, Tissue, Gene, Chemical																																			
<b>Drugs:</b> Name, Dosage, Strength, Route, Duration, Frequency, Poisons, Adverse Effects	<b>Demographics:</b> Age, Gender, Height, Weight, Race, Ethnicity, Marital Status, Vital Signs																																			
<b>Risk Factors:</b> Smoking, Obesity, Diabetes, Hypertension, Substance Abuse	<b>Sensitive Data:</b> Patient Name, Address, Phone, Email, Dates, Providers, Identifiers																																			
<p><b>Clinical Grammar</b></p> <ul style="list-style-type: none"> <li>Stemmer</li> <li>Lemmatizer</li> <li>Part of Speech Tagger</li> <li>Dependency Parser</li> </ul>		<p><b>Find in Text</b></p> <ul style="list-style-type: none"> <li>Text Matcher</li> <li>Regex Matcher</li> <li>Date Matcher</li> <li>Chunker</li> </ul>		<p><b>Trainable &amp; Tunable</b></p> <p><b>Scalable to a Cluster</b></p> <p><b>Fast Inference</b></p> <p><b>Hardware Optimized</b></p> <p><b>Community</b></p>																																

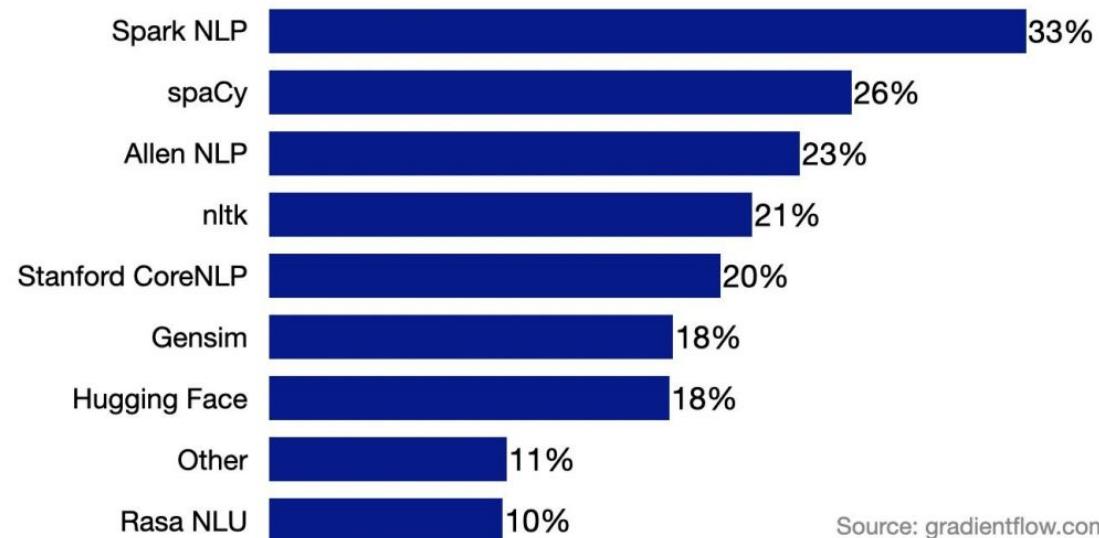
# Spark NLP Modules (Enterprise and Public)

# Spark NLP in Industry

Which of the following AI tools do you use?



Which NLP libraries does your organization use?



Source: gradientflow.com

NLP Industry Survey by Gradient Flow,  
an independent data science research & insights company, September 2021

# TRUSTED BY



Imperial College  
London



STANFORD  
UNIVERSITY

# Biomedical Named Entity Recognition at Scale

Veysel Kocaman  
John Snow Labs Inc.  
16192 Coastal Highway  
Lewes, DE , USA 19958  
veysel@johnsnowlabs.com

**Abstract**—Named entity recognition (NER) is a widely applicable natural language processing task and building block of question answering, topic modeling, information retrieval, etc. In the medical domain, NER plays a crucial role by extracting meaningful chunks from clinical notes and reports, which are then fed to downstream tasks like assertion status detection, entity resolution, relation extraction, and de-identification. Reimplementing a Bi-LSTM-CNN-Char deep learning architecture on top of Apache Spark, we present a single trainable NER model that obtains new state-of-the-art results on seven public biomedical benchmarks without using heavy contextual embeddings like BERT. This includes improving BC4CHEMD to 93.72% (4.1% gain), Species800 to 80.91% (4.6% gain), and JNLPBA to 81.29% (5.2% gain). In addition, this model is freely available within a production-grade code base as part of the open-source Spark NLP library; can scale up for training and inference in any Spark cluster; has GPU support and libraries for popular programming languages such as Python, R, Scala and Java; and can be extended to support other human languages with no code changes.

## I. INTRODUCTION

Electronic health records (EHRs) are the primary source of information for clinicians tracking the care of their patients. Information fed into these systems may be found in structured fields for which values are inputted electronically (e.g. laboratory test orders or results) [1] but most of the time information in these records is unstructured making it largely inaccessible

## Abstract

Named entity recognition (NER) is one of the most important building blocks of NLP tasks in the medical domain by extracting meaningful chunks from clinical notes and reports, which are then fed to downstream tasks like assertion status detection, entity resolution, relation extraction, and de-identification. Due to the growing volume of healthcare data in unstructured format, an increasingly important challenge is providing high accuracy implementations of state-of-the-art deep learning (DL) algorithms at scale. In this study, we introduce a production-grade clinical and biomedical NER algorithm based on a modified BiLSTM-CNN-Char DL architecture built on top of Apache Spark. This algorithm establishes new state-of-the-art accuracy on 7 of 8 well-known biomedical NER benchmarks and 3 clinical concept extraction challenges: 2010 i2b2/VA clinical concept extraction, 2014 n2c2 de-identification, and 2018 n2c2 medication extraction. Moreover, clinical NER models trained using this implemen-

Anonymous NAACL-HLT 2021 submission

# Spark NLP: Natural Language Understanding at Scale

Veysel Kocaman, David Talby

John Snow Labs Inc.  
16192 Coastal Highway  
Lewes, DE , USA 19958  
eysel, david}@johnsnowlabs.com

## Accurate Clinical and Biomedical Named Entity Recognition at Scale

## Improving Clinical Document Understanding on COVID-19 Research with Spark NLP

Veysel Kocaman, David Talby

John Snow Labs Inc.  
16192 Coastal Highway  
Lewes, DE , USA 19958  
{eysel, david}@johnsnowlabs.com

## Abstract

Following the global COVID-19 pandemic, the number of scientific papers studying the virus has grown massively, leading to increased interest in automated literature review. We present a clinical text mining system that improves on previous efforts in three ways. First, it can recognize over 100 different entity types including social determinants of health, anatomy, risk factors, and adverse events in addition to other commonly used clinical and biomedical entities. Second, the text processing pipeline includes assertion status detection, to distinguish between clinical facts that are present, absent, conditional, or about someone other than the patient. Third, the deep learning models used are more accurate than previously available, leveraging an integrated pipeline of state-of-the-art pre-trained named entity recognition models, and improving on the previous best performing benchmarks for assertion status detection. We illustrate extracting trends and insights - e.g. most frequent disorders and symptoms, and most common vital signs and EKG findings – from the COVID-19 Open Research Dataset (CORD-19). The system is built using the Spark NLP library which natively supports scaling to use distributed clusters, leveraging GPU's, configurable and reusable NLP pipelines, healthcare-specific embeddings, and the ability to train models to support new entity types or human languages with no code changes.

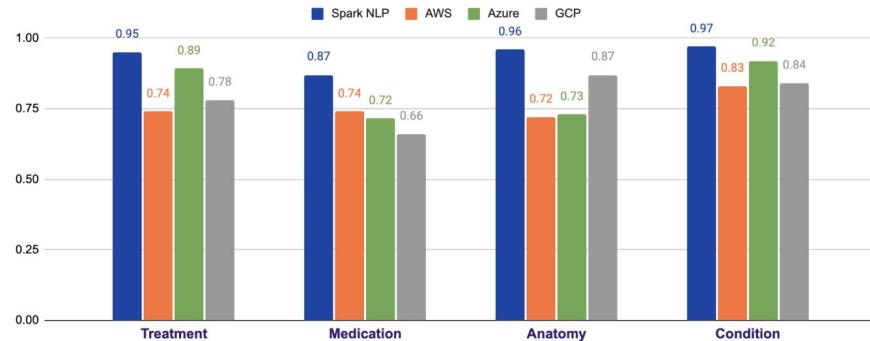
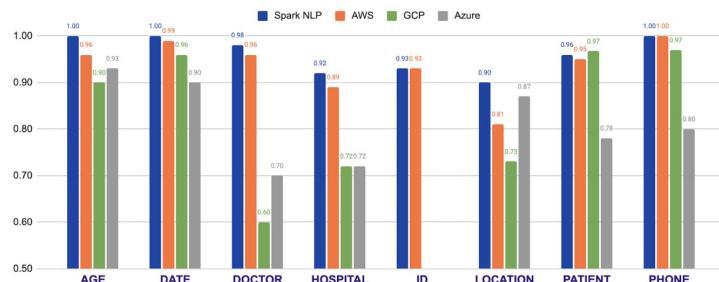
be found in structured fields for which values are inputted electronically (e.g. laboratory test orders or results) (Liede et al. 2015) but most of the time information in these records is unstructured making it largely inaccessible for statistical analysis (Murdoch and Detsky 2013). These records include information such as the reason for administering drugs, previous disorders of the patient or the outcome of past treatments, and they are the largest source of empirical data in biomedical research, allowing for major scientific findings in highly relevant disorders such as cancer and Alzheimer's disease (Perera et al. 2014).

A primary building block in such text mining systems is named entity recognition (NER) - which is regarded as a critical precursor for question answering, topic modelling, information retrieval, etc (Yadav and Bethard 2019). In the medical domain, NER recognizes the first meaningful chunks out of a clinical note, which are then fed down the processing pipeline as an input to subsequent downstream tasks such as clinical assertion status detection (Uzuner et al. 2011), clinical entity resolution (Tzitzivacos 2007) and de-identification of sensitive data (Uzuner, Luo, and Szolovits 2007) (see Figure 1). However, segmentation of clinical and drug entities is considered to be a difficult task in biomedical NER systems because of complex orthographic structures of named entities

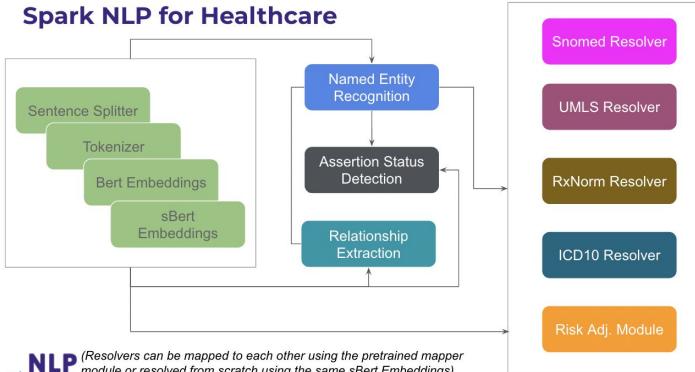
Peer-reviewed papers on  
Spark NLP NER

# Most Accurate in the Industry

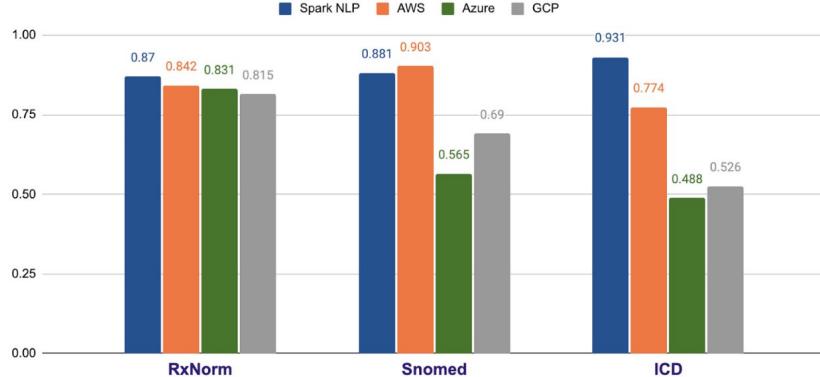
## De-Identification Benchmarks (en)



## Spark NLP for Healthcare



## Top - 5 Results



# Introducing Spark NLP

# Introducing Spark NLP



Faster inference

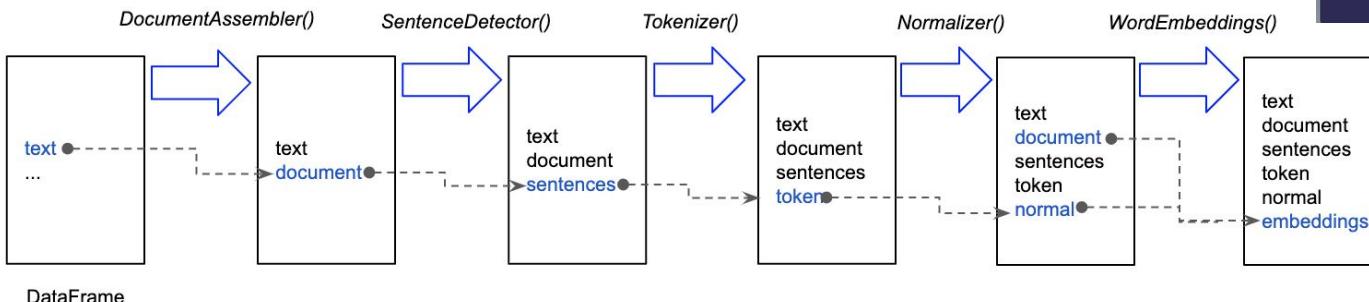
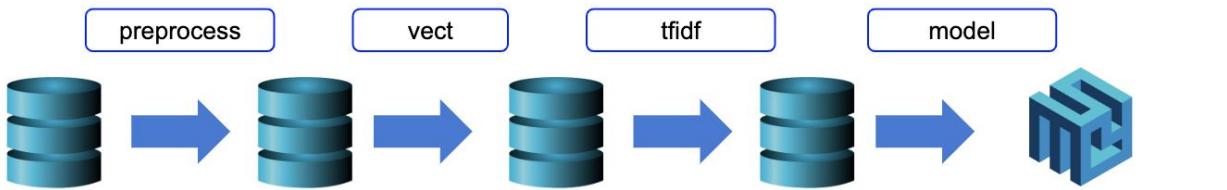
```
from sparknlp.base import LightPipeline  
LightPipeline(someTrainedPipeline).annotate(someStringOrArray)
```

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.

**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).

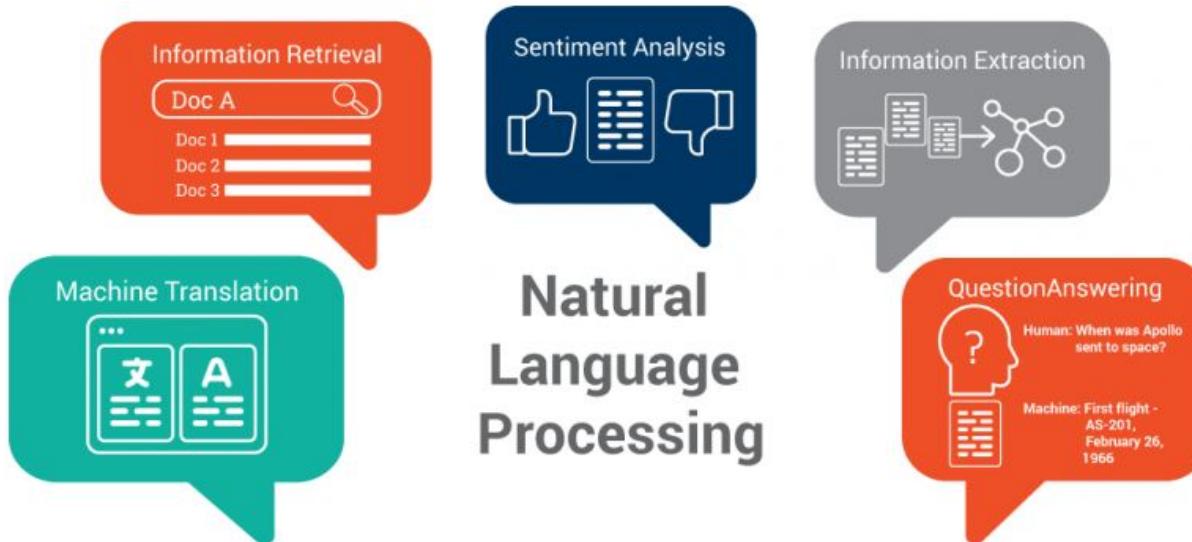
# Introducing Spark NLP

## 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)
```

# NLP Basics



# NLP Basics

## Tokenization

Tokenize on  
rules



Tokenize on  
punctuation



Tokenize on  
white spaces



- First step in NLP pipelines, raw text is broken down into smaller, more manageable pieces. Without tokenization, NLP models would have to process the text as one large unit.
- Many other tokenization methods (character, sub-word, sentence).

# NLP Basics

## LEMMATIZATION

Find the **lemma** of each word:

- How does it show in the dictionary?

Uses a lookup from a full dictionary.

am, are, is → be

liver → liver

lives → live

## STEMMING

Find the **stem** of each word.

Uses rules: e.g, remove common suffixes.

Form	Suffix	Stem
studies	-es	studi
study <sup>ing</sup>	-ing	study
niñas	-as	niñ
niñez	-ez	niñ

- The goal of both **stemming** and **lemmatization** is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form for normalization purposes.
- Lemmatization always returns real words, **stemming** doesn't.

# NLP Basics

## Stopword Removal

it was a bright cold day in april  
and the clocks **were** striking  
thirteen winston smith **his** chin  
nuzzled **into his** breast in an  
effort **to** escape **the** vile wind  
slipped quickly **through the** glass  
doors **of** victory mansions though  
**not** quickly enough **to** prevent a  
swirl **of** gritty dust **from** entering  
along **with him**



bright cold day april clocks  
striking thirteen winston smith  
chin nuzzled breast effort  
escape vile wind slipped quickly  
glass doors victory mansions  
though quickly enough prevent  
swirl gritty dust entering along

- For tasks like text classification, where the text is to be classified into different categories, **stopwords** are **removed** or excluded from the given text so that more focus can be given to those words which define the meaning of the text. Stopwords are considered insignificant in this case.

### Stopwords

a  
able  
about  
above  
according  
accordingly  
across  
actually  
after  
afterwards  
again  
against  
ain  
all  
allow  
allows  
almost  
alone  
along  
already  
also

(520 stopwords)

## Pre-Processing

# Spell Checking & Correction



```
val pipeline = PretrainedPipeline("spell_check_ml", "en")
val result = pipeline.annotate("Harry Potter is a graet muvie")

println(result("spell"))
/* will print Seq[String](..., "is", "a", "great", "movie") */
```

- 3 trainable approaches
- **Norvig Approach:**
  - Retrieves tokens and auto-corrects based on a given dictionary
- **Symmetric Delete:**
  - Uses distance metrics to find possible words
- **Context Aware:**
  - Most accurate: Judges words in context
  - Deep learning based

## Pre-Processing

# Context Spell Checker

The Spell Checker can leverage the context of words for ranking different correction sequences. Let's take a look at some examples,

```
# check for the different occurrences of the word "siter"
example1 = ["I will call my siter.", \
            "Due to bad weather, we had to move to a different siter.", \
            "We travelled to three siter in the summer."]
beautify(lp.annotate(example1))
```

```
['I will call my sister .\n',
 'Due to bad weather , we had to move to a different site .\n',
 'We travelled to three sites in the summer .\n']
```

```
# check for the different occurrences of the word "ueather"
example2 = ["During the summer we have the best ueather.", \
            "I have a black ueather jacket, so nice.", \
            "I introduce you to my sister, she is called ueather."]
beautify(lp.annotate(example2))
```

```
['During the summer we have the best weather .\n',
 'I have a black leather jacket , so nice .\n',
 'I introduce you to my sister , she is called Heather .\n']
```

Notice that in the first example, 'siter' is indeed a valid English word,

<https://www.merriam-webster.com/dictionary/siter>

# Normalization

Remove or replace undesirable characters or regular expressions:

from: @Have a\$ #2great birth) day>!  
to: Have a great birth day!

Spark NLP also comes with a Slang normalizer:

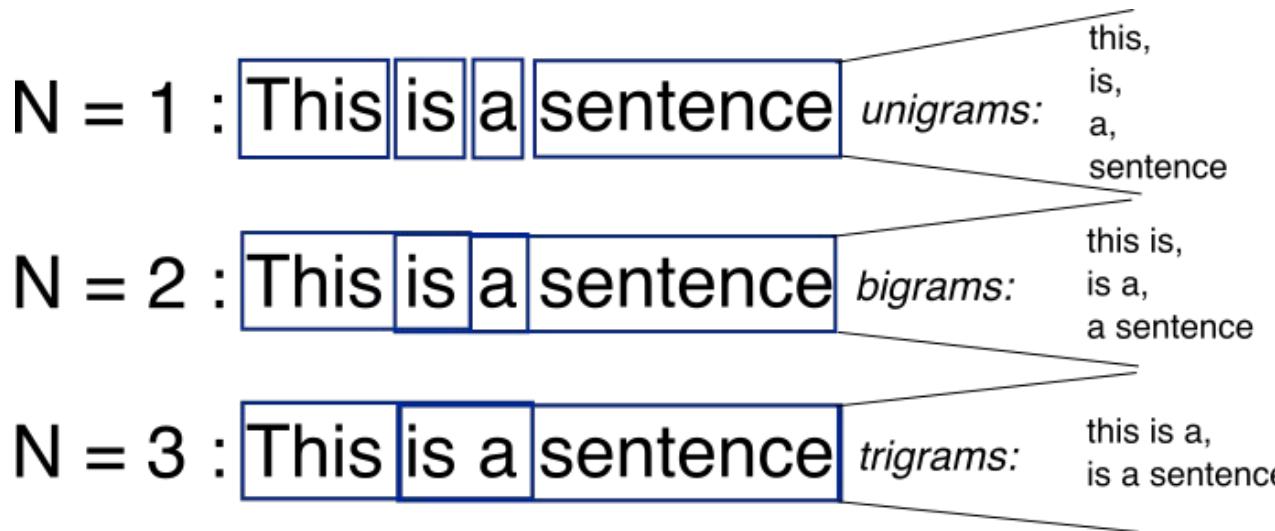
Original tweet

@USER, r u cuming 2 MidCorner dis Sunday?

Normalized tweet

@USER, are you coming to MidCorner this Sunday?

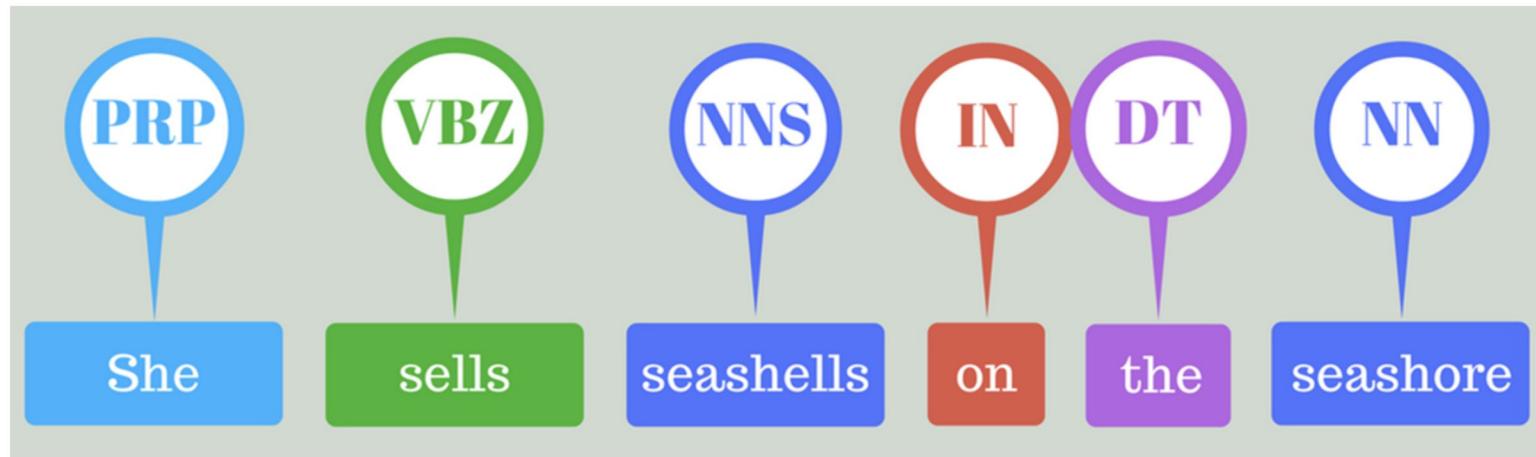
# N-gram Tokenization



- Kind of tokenizers which split words or sentences into several tokens
- Each token has certain number of words
- Number of words depends on the type of n-gram tokenizer
- Unigram, bigram, trigram, etc.

# Part Of Speech Tagging

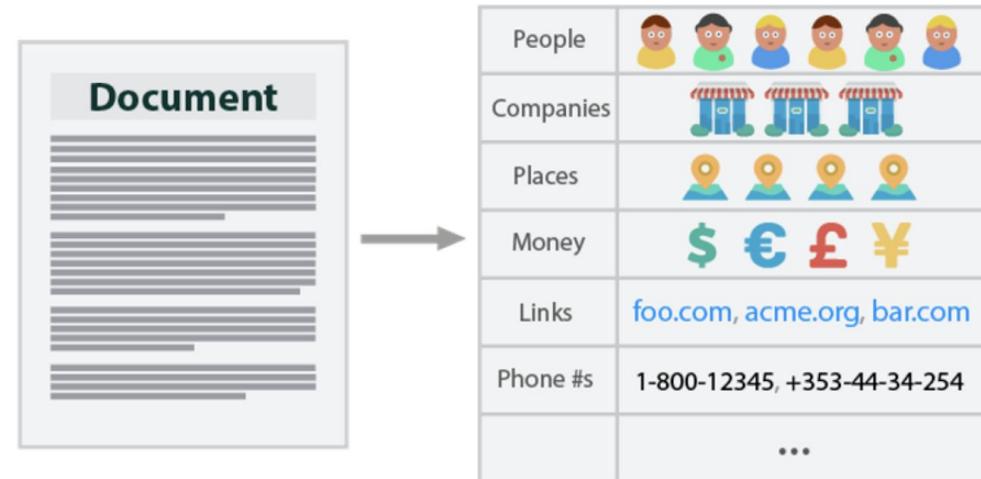
Often useful for recognizing named entities or word relationships.



A **POS tag** (or **part-of-speech tag**) is a special label assigned to each token (word) in a text corpus to indicate the **part of speech** and often also other grammatical categories such as tense, number (plural/singular), case etc.

# Named Entity Recognition (NER)

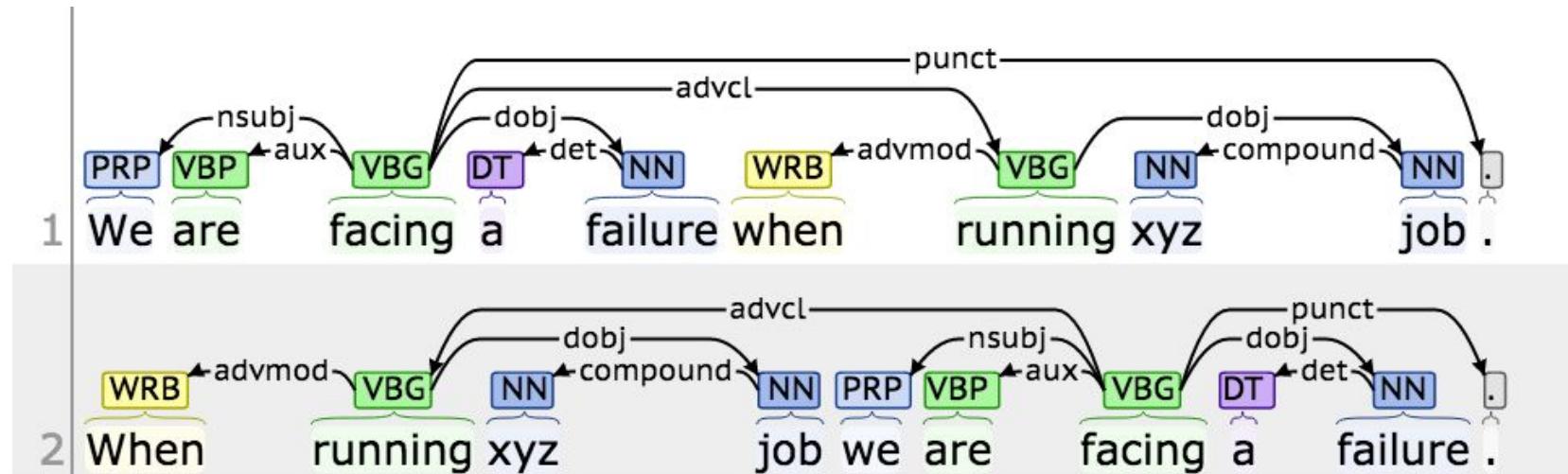
NER is a subtask of information extraction that seeks to **locate and classify named entity** mentioned in unstructured text into pre-defined categories such as **person** names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc.



But Google **ORG** is starting from behind. The company made a late push into hardware, and Apple **ORG**'s Siri **PRODUCT**, available on iPhones **PRODUCT**, and Amazon **ORG**'s Alexa **PRODUCT** software, which runs on its Echo **PRODUCT** and Dot **PRODUCT** devices, have clear leads in consumer adoption.

# Dependency Parsing

Useful for extracting relationships (i.e. building knowledge graphs):



# Session 1 (Day 1) - Coding Time

- ❖ [Notebook 1 - Spark NLP Basics](#)

Spark NLP  
for Data Scientists



# Session 2 (Day 1)

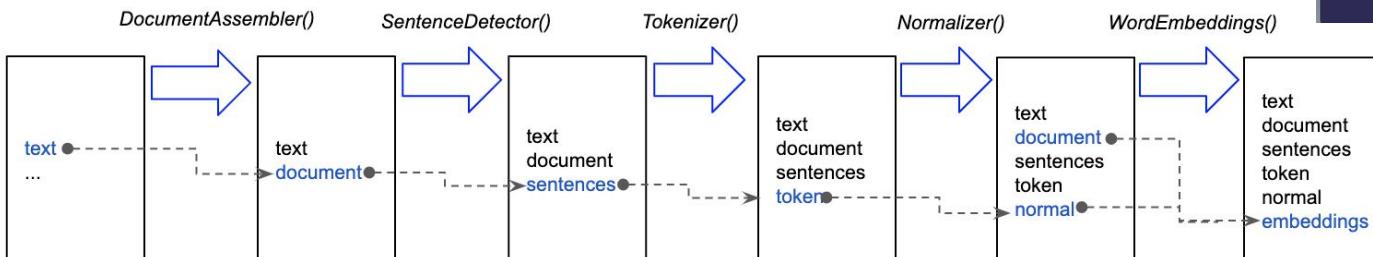
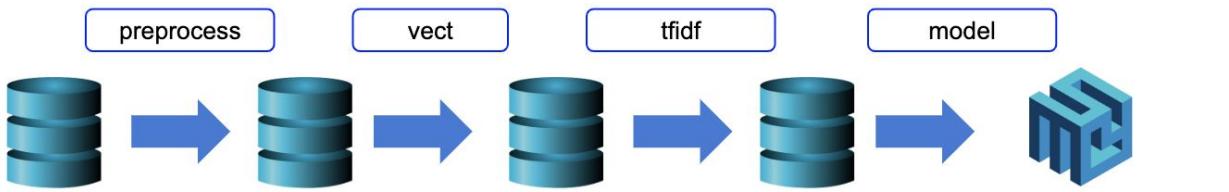
- ❖ Text Preprocessing
- ❖ Composing Pipelines
- ❖ Working with Spark Dataframes
- ❖ Notebook 2 Text Preprocessing and composing Pipelines

Spark NLP  
for Data Scientists



# Pipeline Structure

## 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)
```

# Session 2 (Day 1) - Coding Time

- ❖ Notebook 2 Text Preprocessing and composing Pipelines

# Session 3 (Day 1)

- ❖ Usage and overview of the 10000+ pretrained models for 300+ languages
- ❖ [Notebook 3 Spark NLP pretrained models](#)

Spark NLP  
for Data Scientists



<p><b>Clinical Entity Recognition</b></p> <p>40 units <b>dosage</b> of insulin glargine <b>drug</b> at night <b>frequency</b></p>				<p><b>Clinical Entity Linking</b></p> <p>Suspect diabetes SNOMED-CT: 473127005 Lisinopril 10 MG RxNorm: 316151 Hyponatremia ICD-10: E87.1</p>	<p><b>Assertion Status</b></p> <p>Fever and sore throat → PRESENT No stomach pain → ABSENT Father with Alzheimer → FAMILY</p>	<p><b>Relation Extraction</b></p>																						
<p><b>Algorithms</b></p>				<p><b>Content</b></p>																								
<p><b>Extract Knowledge</b></p> <ul style="list-style-type: none"> <li>Entity Linker</li> <li>Entity Disambiguator</li> <li>Document Classifier</li> <li>Contextual Parser</li> </ul>		<p><b>De-identify text</b></p> <ul style="list-style-type: none"> <li>Structured Data</li> <li>Unstructured Text</li> <li>Obfuscator</li> <li>Generalizer</li> </ul>	<p><b>Medical Transformers</b></p> <table border="1"> <tr><td>JSL-BERT-Clinical</td><td>BioBERT</td></tr> <tr><td>ClinicalBERT</td><td>GloVe-Med</td></tr> <tr><td>GloVe-ICD-O</td><td>BlueBERT</td></tr> </table>	JSL-BERT-Clinical	BioBERT	ClinicalBERT	GloVe-Med	GloVe-ICD-O	BlueBERT	<p><b>Linked Medical Terminologies</b></p> <table border="1"> <tr><td>SNOMED-CT</td><td>CPT</td><td>UMLS</td></tr> <tr><td>ICD-10-CM</td><td>RxNorm</td><td>HPO</td></tr> <tr><td>ICD-10-PCS</td><td>ICD-O</td><td>LOINC</td></tr> </table>	SNOMED-CT	CPT	UMLS	ICD-10-CM	RxNorm	HPO	ICD-10-PCS	ICD-O	LOINC	<p><b>800+ Pretrained Models</b></p> <table border="1"> <tr> <td><b>Clinical:</b> Signs, Symptoms, Treatments, Procedures, Tests, Labs, Sections</td> <td><b>Anatomy:</b> Organ, Subdivision, Cell, Structure, Organism, Tissue, Gene, Chemical</td> </tr> <tr> <td><b>Drugs:</b> Name, Dosage, Strength, Route, Duration, Frequency, Poisons, Adverse Effects</td> <td><b>Demographics:</b> Age, Gender, Height, Weight, Race, Ethnicity, Marital Status, Vital Signs</td> </tr> <tr> <td><b>Risk Factors:</b> Smoking, Obesity, Diabetes, Hypertension, Substance Abuse</td> <td><b>Sensitive Data:</b> Patient Name, Address, Phone, Email, Dates, Providers, Identifiers</td> </tr> </table>		<b>Clinical:</b> Signs, Symptoms, Treatments, Procedures, Tests, Labs, Sections	<b>Anatomy:</b> Organ, Subdivision, Cell, Structure, Organism, Tissue, Gene, Chemical	<b>Drugs:</b> Name, Dosage, Strength, Route, Duration, Frequency, Poisons, Adverse Effects	<b>Demographics:</b> Age, Gender, Height, Weight, Race, Ethnicity, Marital Status, Vital Signs	<b>Risk Factors:</b> Smoking, Obesity, Diabetes, Hypertension, Substance Abuse	<b>Sensitive Data:</b> Patient Name, Address, Phone, Email, Dates, Providers, Identifiers	
JSL-BERT-Clinical	BioBERT																											
ClinicalBERT	GloVe-Med																											
GloVe-ICD-O	BlueBERT																											
SNOMED-CT	CPT	UMLS																										
ICD-10-CM	RxNorm	HPO																										
ICD-10-PCS	ICD-O	LOINC																										
<b>Clinical:</b> Signs, Symptoms, Treatments, Procedures, Tests, Labs, Sections	<b>Anatomy:</b> Organ, Subdivision, Cell, Structure, Organism, Tissue, Gene, Chemical																											
<b>Drugs:</b> Name, Dosage, Strength, Route, Duration, Frequency, Poisons, Adverse Effects	<b>Demographics:</b> Age, Gender, Height, Weight, Race, Ethnicity, Marital Status, Vital Signs																											
<b>Risk Factors:</b> Smoking, Obesity, Diabetes, Hypertension, Substance Abuse	<b>Sensitive Data:</b> Patient Name, Address, Phone, Email, Dates, Providers, Identifiers																											
<b>Trainable &amp; Tunable</b>	<b>Scalable to a Cluster</b>	<b>Fast Inference</b>	<b>Hardware Optimized</b>	<b>Community</b>		<p><b>Entity Recognition</b> I love Lucy PERSON</p> <p><b>Information Extraction</b> They met <u>Last week</u> DATE → 29-04-2020</p> <p><b>Spelling &amp; Grammar</b> abc She become the first.. → She became the first</p> <p><b>Text Classification</b> </p> <p><b>Image Classification</b> </p> <p><b>Translation</b>  [je t'aime → I love you]</p> <p><b>Summarization</b> </p> <p><b>Question Answering</b> </p> <p><b>Emotion Detection</b> </p> <p><b>Automatic Speech Recognition</b> </p>																						
<p><b>Split Text</b></p> <ul style="list-style-type: none"> <li>Sentence Detector</li> <li>Tokenizer</li> <li>Normalizer</li> <li>nGram Generator</li> <li>Word Segmentation</li> </ul>				<p><b>Clean Text</b></p> <ul style="list-style-type: none"> <li>Spell Checker</li> <li>Grammar Checker</li> <li>Writing Style Checker</li> <li>Stopword Cleaner</li> <li>Summarization</li> </ul>	<p><b>12,000+</b> Pre-trained Pipelines, Models &amp; Transformers</p> <table border="1"> <tr><td>BERT</td><td>ELMO</td><td>TAPAS</td></tr> <tr><td>ALBERT</td><td>DeBERTa</td><td>USE</td></tr> <tr><td>Longformer</td><td>ELECTRA</td><td></td></tr> <tr><td>T5</td><td>NMT</td><td>ViT</td></tr> <tr><td>DistilBERT</td><td>RoBERTa</td><td></td></tr> <tr><td>XLM-RoBERTa</td><td></td><td></td></tr> <tr><td>Wav2Vec2</td><td>XLNet</td><td></td></tr> </table>			BERT	ELMO	TAPAS	ALBERT	DeBERTa	USE	Longformer	ELECTRA		T5	NMT	ViT	DistilBERT	RoBERTa		XLM-RoBERTa			Wav2Vec2	XLNet	
BERT	ELMO	TAPAS																										
ALBERT	DeBERTa	USE																										
Longformer	ELECTRA																											
T5	NMT	ViT																										
DistilBERT	RoBERTa																											
XLM-RoBERTa																												
Wav2Vec2	XLNet																											
<p><b>Understand Grammar</b></p> <ul style="list-style-type: none"> <li>Stemmer</li> <li>Lemmatizer</li> <li>Part of Speech Tagger</li> <li>Dependency Parser</li> <li>Translation</li> </ul>				<p><b>Find in Text</b></p> <ul style="list-style-type: none"> <li>Text Matcher</li> <li>Regex Matcher</li> <li>Date Matcher</li> <li>Dependency Parser</li> <li>Chunker</li> <li>Question Answering</li> </ul>																								
<p><b>Split Text</b></p> <ul style="list-style-type: none"> <li>Sentence Detector</li> <li>Deep Sentence Detector</li> <li>Tokenizer</li> <li>nGram Generator</li> </ul>		<p><b>Clean Medical Text</b></p> <ul style="list-style-type: none"> <li>Spell Checking</li> <li>Spell Correction</li> <li>Normalizer</li> <li>Stopword Cleaner</li> </ul>	<p><b>Trainable &amp; Tunable</b></p>	<p><b>Scalable to a Cluster</b></p>	<p><b>Fast Inference</b></p>	<p><b>Hardware Optimized</b></p>	<p><b>Community</b></p>																					
<p><b>Clinical Grammar</b></p> <ul style="list-style-type: none"> <li>Stemmer</li> <li>Lemmatizer</li> <li>Part of Speech Tagger</li> <li>Dependency Parser</li> </ul>																												

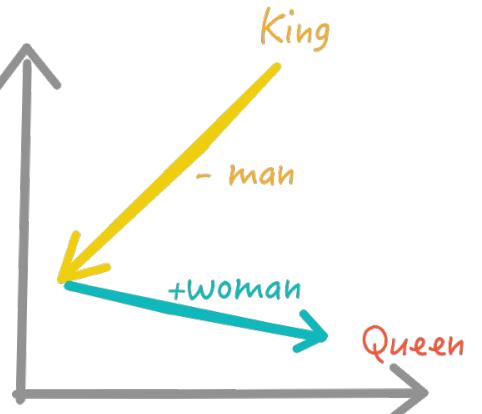
# Spark NLP Modules (Enterprise and Public)

# Word & Sentence Embeddings

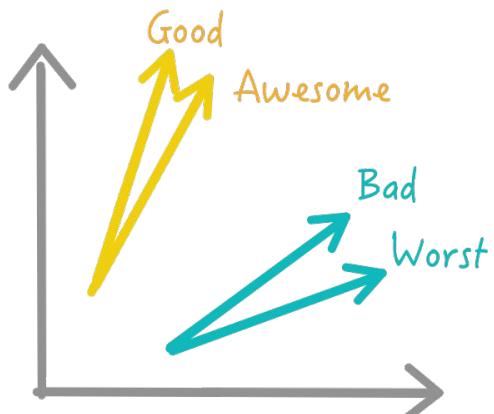
- Neural Networks and most other algorithms don't work well with strings
  - Why not convert a string into a vector with numbers!
  - Semantically similar words vectors should be **close** in the vector space learned by the
  - Dissimilar words are **far** from each other in vector space
  - Usually high dimensional, to give model more space to encode semantics
- Useful for many downstream tasks and commonly used in NLP
- Reduce dimensionality of the input space

Raw Text	Bag-of-words vector
it	2
they	0
puppy	1
and	1
cat	0
aardvark	0
cute	1
extremely	1
...	...

it is a puppy and it  
is extremely cute



a) Learns Analogy



b) Similar Words have same angles

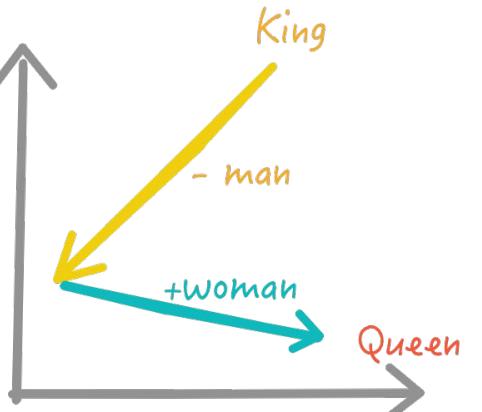
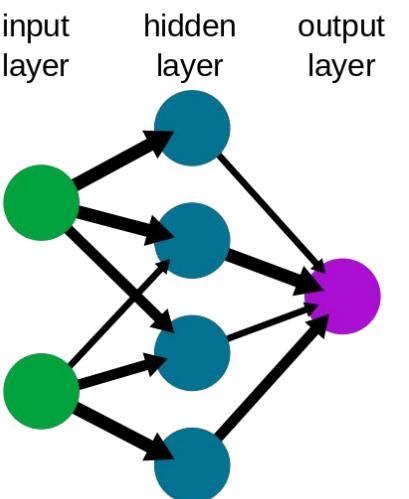
```
In [9]: doc[3].vector
Out[9]: array([-0.037103, -0.31259, -0.17857, 0.30001, 0.078154,
   -0.17958, 0.12048, -0.11879, -0.20601, 1.2849,
   -0.20409, 0.80613, 0.34344, -0.19191, -0.084511,
   0.17339, 0.042483, 2.0282, -0.16278, -0.60306,
   -0.53766, 0.35711, 0.22882, 0.1171, 0.42983,
   0.16165, 0.407, 0.036476, 0.52636, -0.13524,
   -0.016897, 0.029259, -0.079115, -0.32305, 0.052255,
   -0.36618, -0.18218, -0.098909, -0.05951, 0.16881,
   0.21018, -0.08376, -0.098909, -0.30496, -0.26935,
   0.0021152, -0.32512, 0.063977, 0.36249, 0.0021152,
   -0.59341, -0.13625, 0.016425, -0.2474, -0.07498,
   0.034708, -0.01476, -0.11648, 0.25559, -0.35002,
   -0.52707, 0.21221, 0.062456, 0.26184, 0.53149,
   0.34957, -0.22692, 0.440705, 0.4438, 0.6335,
   -0.049757, -0.08134, 0.65618, -0.4716, 0.090675,
   -0.084873, 0.31455, -0.38495, -0.19247, 0.48064,
   0.26688, 0.095743, 0.13024, 0.37023, 0.46269,
   -0.32844, 0.17375, -0.36325, 0.30672, -0.075042,
   -0.64684, -0.49822, 0.12373, -0.28547, 0.61811,
   -0.19224, 0.0040473, 0.1774, 0.033154, -0.54862,
   0.34695, -0.53506, -0.013381, 0.085712, -0.054447,
   -0.64673, 0.016749, 0.47676, 0.037803, -0.10966,
   -0.4165, -0.20252, 0.2794, 0.10852, -0.40154,
```

# Word & Sentence Embeddings

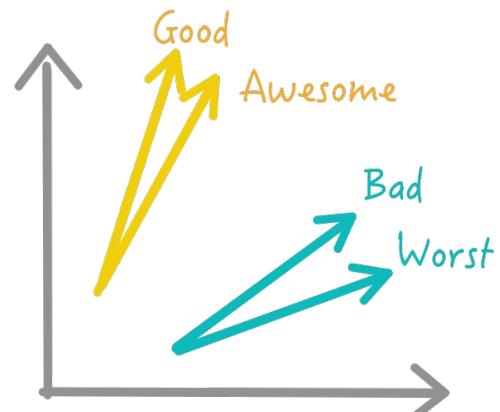
- Deep-Learning-based natural language processing systems encode **words** and **sentences** 📄 in fixed-length dense vectors ⚒ to drastically improve the processing of textual data.
- Based on **The Distributional Hypothesis**: Words that occur in the same contexts tend to have similar meanings.
- Elmo and Bert-family embeddings are context-aware.

```
In [9]: doc[3].vector  
Out[9]: array([-0.037103, -0.31259, -0.17857, 0.30001, 0.078154,  
    0.17958, 0.12048, -0.11879, -0.20601, 1.2849,  
    -0.20409, 0.80613, 0.34344, -0.19191, -0.084511,  
    0.17339, 0.042483, 2.0282, -0.16278, -0.60306,  
    -0.53766, 0.35711, 0.22882, 0.1171, 0.42983,  
    0.16165, 0.407, 0.036476, 0.52636, -0.13524,  
    -0.016897, 0.029259, -0.079115, -0.32305, 0.052255,  
    -0.36615, -0.1821, -0.09819, -0.1591, 0.16881,  
    0.21018, -0.88376, -0.098909, -0.34946, -0.26935,  
    0.00221132, -0.32512, 0.063977, 0.36249, 0.16881,  
    -0.59341, -0.13625, 0.016425, -0.2474, -0.07498,  
    0.034708, -0.01476, -0.11648, 0.25559, -0.35002,  
    -0.52707, 0.21221, 0.062456, 0.26184, 0.53149,  
    0.34957, -0.22692, 0.440707, 0.4438, 0.6335,  
    -0.049757, -0.08134, 0.65618, -0.4716, 0.090675,  
    -0.084873, 0.31455, -0.38495, -0.19247, 0.48064,  
    0.26688, 0.095743, 0.13024, 0.37023, 0.46269,  
    -0.32844, 0.17375, -0.36325, 0.30672, -0.075042,  
    -0.64684, -0.49822, 0.12372, -0.28547, 0.61811,  
    -0.19224, 0.0040473, 0.1774, 0.033154, -0.54862,  
    0.34695, -0.53506, -0.013381, 0.085712, -0.054447,  
    -0.64673, 0.016749, 0.47676, 0.037803, -0.10966,  
    -0.4165, -0.20252, 0.2794, 0.10852, -0.40154])
```

A simple neural network



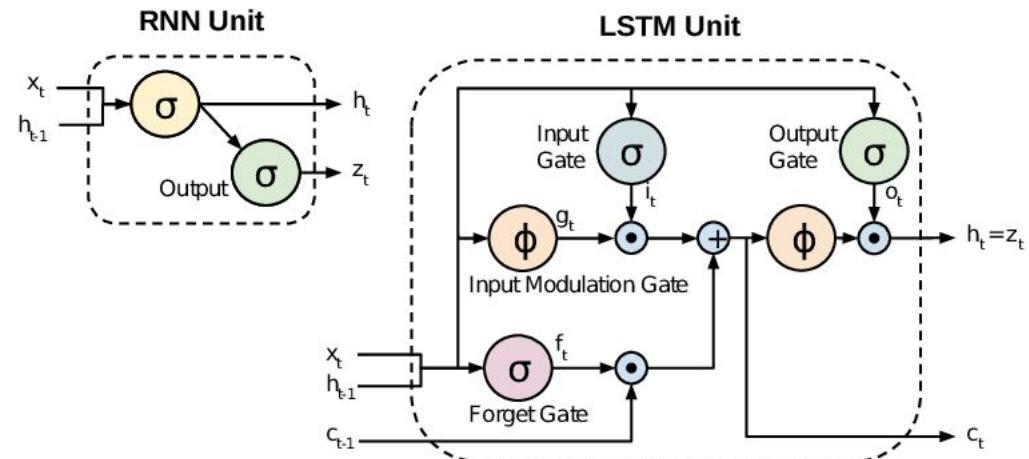
a) Learns Analogy



b) Similar Words have same angles

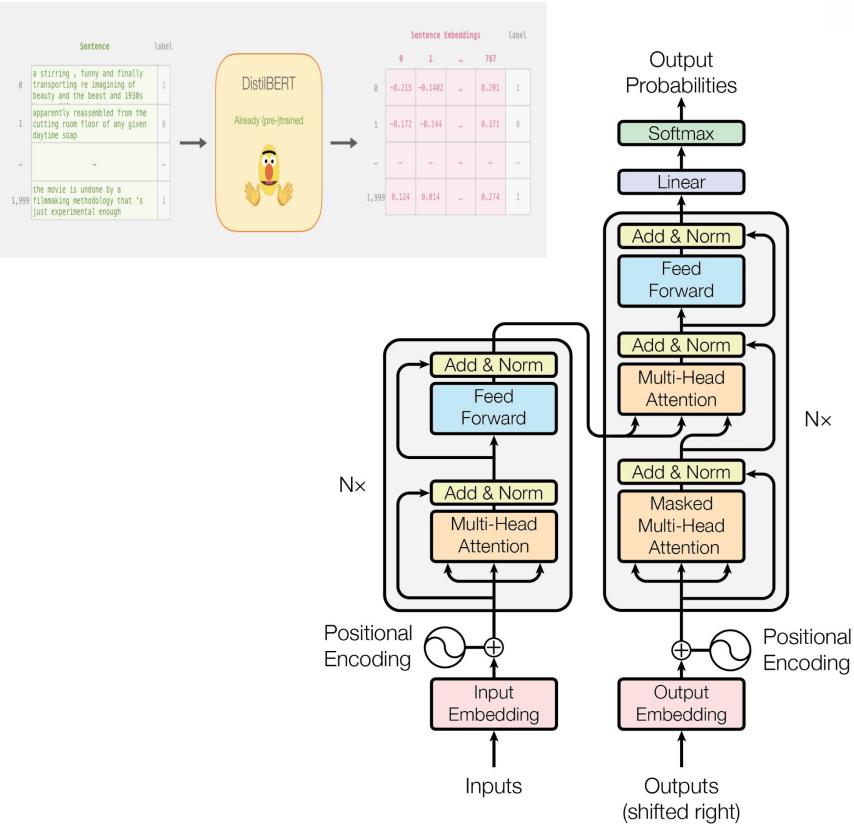
# LSTM's and RNN's for Embeddings

- RNN :
  - Slow to train
  - Not well parallelizable
  - Vanishing/Exploding Gradients
- LSTM
  - Fixes vanishing/exploding gradient
  - not scalable
  - No transfer learning



# Transformers - Attention is All you Need - Visualized

## A highly parallelizable NN Architecture for Sequential Data



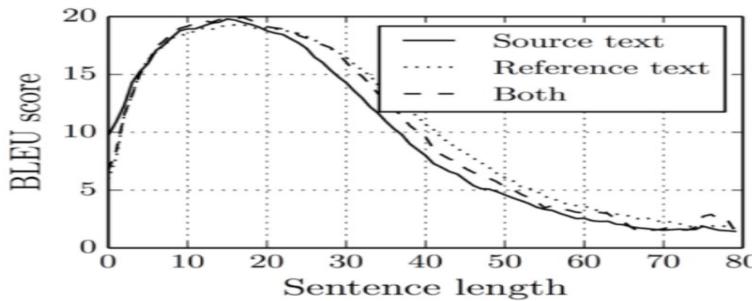
<https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html>

# Feature-Engineering

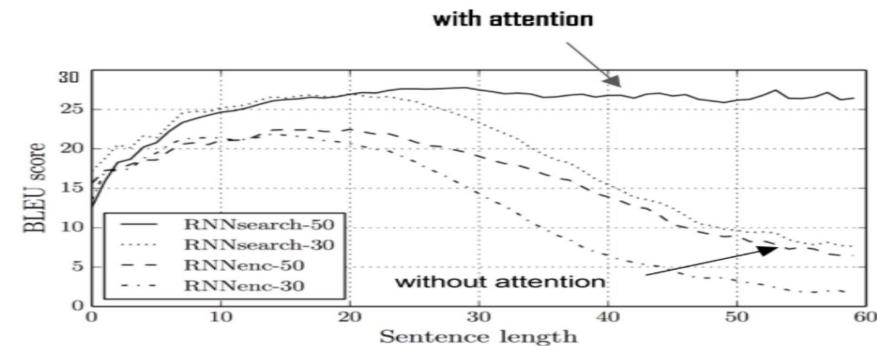
# Transformers - Attention is All you Need - Why they rock

## A highly parallelizable NN Architecture for Sequential Data

### NMT with LSTMs



### NMT with LSTMs + attention



Bahdanau et al. 2014

Slide: Text generation with attention, GTC 2017, Valentin Malykh (2017)

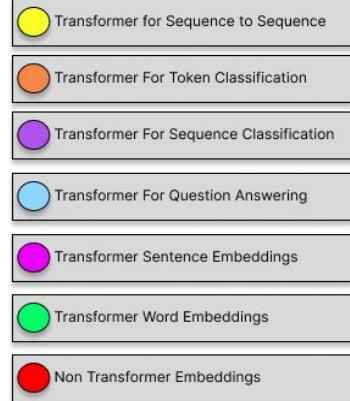
Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(\log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

# The most scalable, accurate and fastest arsenal of NLP transformers in history



## Core NLP Transformer Models

CamemBertEmbeddings	GloveEmbeddings	Word2VecEmbeddings	Doc2VecEmbeddings
DeBertaEmbeddings	LongformerEmbeddings	BertEmbeddings	XlmRoBERTaEmbeddings
DeBertaForQuestionAnswering	LongformerForTokenClassification	BertForTokenClassification	XlmRoBERTaForTokenClassification
GPT2Transformer	LongformerForSequenceClassification	BertForSequenceClassification	XlmRoBERTaForSequenceClassification
MarianTransformer	LongformerForQuestionAnswering	BertForQuestionAnswering	XlmRoBERTaForQuestionAnswering
T5Transformer	UniversalSentenceEncoder	BertSentenceEmbeddings	XlmRoBERTaSentenceEmbeddings
RoBERTaEmbeddings	XlnetEmbeddings	DistilBertEmbeddings	AlbertEmbeddings
RoBERTaForTokenClassification	XlnetForTokenClassification	DistilBertForTokenClassification	AlbertForTokenClassification
RoBERTaForSequenceClassification	XlnetForSequenceClassification	DistilBertForSequenceClassification	AlbertForSequenceClassification
RoBERTaForQuestionAnswering	ElmoEmbeddings	DistilBertForQuestionAnswering	AlbertForQuestionAnswering



## Subflavored NLP Transformer Extensions

SciBertEmbeddings	ElectraForQuestionAnswering	MurilBertForQuestionAnswering	MurilBertEmbeddings	LabseSentenceEmbeddings
IndoBertForQuestionAnswering	ElectraSentenceEmbeddings	MurilBertSentenceEmbeddings	SciBertForQuestionAnswering	SapBertForQuestionAnswering
ElectraEmbeddings	BioBertForQuestionAnswering	CovidBertForQuestionAnswering	MiniLmForQuestionAnswering	MacBertForQuestionAnswering
BioFormerForQuestionAnswering	BioBertSentenceEmbeddings	CovidDistilBertForQuestionAnswering	LinkBertForQuestionAnswering	BioBertEmbeddings

Search them all in the Modelshub  
<https://nlp.johnsnowlabs.com/models>

Number	Language	Spark NLP Model Name	Model
0	en	glove_100d	WordEmbeddingsModel
1	xx	glove_6B_300	WordEmbeddingsModel
2	xx	glove_840B_300	WordEmbeddingsModel
3	en	embeddings_clinical	WordEmbeddingsModel
4	en	elmo	ElmoEmbeddings
5	en	embeddings_healthcare	WordEmbeddingsModel
6	en	tflite_use	UniversalSentenceEncoder
7	en	tflite_use_lg	UniversalSentenceEncoder
8	en	albert_base_uncased	AlbertEmbeddings
9	en	albert_large_uncased	AlbertEmbeddings
10	en	albert_xlarge_uncased	AlbertEmbeddings
11	en	albert_xxlarge_uncased	AlbertEmbeddings
12	en	xlnet_base_cased	XlnetEmbeddings
13	en	xlnet_large_cased	XlnetEmbeddings
14	es	embeddings_scielo_150d	WordEmbeddingsModel
15	es	embeddings_scielo_300d	WordEmbeddingsModel
16	es	embeddings_scielo_50d	WordEmbeddingsModel
17	es	embeddings_scielowiki_150d	WordEmbeddingsModel
18	es	embeddings_scielowiki_300d	WordEmbeddingsModel
19	es	embeddings_scielowiki_50d	WordEmbeddingsModel
20	es	embeddings_scifiwiki_150d	WordEmbeddingsModel
21	es	embeddings_scifiwiki_300d	WordEmbeddingsModel
22	es	embeddings_scifiwiki_50d	WordEmbeddingsModel
23	en	embeddings_healthcare_100d	WordEmbeddingsModel
24	en	embeddings_biovec	WordEmbeddingsModel
25	en	bert_base_cased	BertEmbeddings

# Embeddings

in

## Spark NLP

### Part 1

Search them all in the Modelshub  
<https://nlp.johnsnowlabs.com/models>

Number	Language	Spark NLP Model Name	Model
26	en	bert_base_uncased	BertEmbeddings
27	en	bert_large_cased	BertEmbeddings
28	en	bert_large_uncased	BertEmbeddings
29	xx	bert_multi_cased	BertEmbeddings
30	en	biobert_clinical_base_cased	BertEmbeddings
31	en	biobert_discharge_base_cased	BertEmbeddings
32	en	biobert_pubmed_base_cased	BertEmbeddings
33	en	biobert_pubmed_cased	BertEmbeddings
34	en	biobert_pubmed_large_cased	BertEmbeddings
35	en	biobert_pubmed_pubmed_cased	BertEmbeddings
36	en	sent.biobert_clinical_base_cased	BertSentenceEmbeddings
37	en	sent.bert_base_cased	BertSentenceEmbeddings
38	en	sent.bert_base_uncased	BertSentenceEmbeddings
39	en	sent.bert_large_cased	BertSentenceEmbeddings
40	en	sent.bert_large_uncased	BertSentenceEmbeddings
41	xx	sent.bert_multi_cased	BertSentenceEmbeddings
42	en	sent.biobert_discharge_base_cased	BertSentenceEmbeddings
43	en	sent.biobert_pubmed_base_cased	BertSentenceEmbeddings
44	en	sent.biobert_pubmed_cased	BertSentenceEmbeddings
45	en	sent.biobert_pubmed_large_cased	BertSentenceEmbeddings
46	en	sent.biobert_pubmed_pubmed_cased	BertSentenceEmbeddings
47	en	sent.small.bert.L10.128	BertSentenceEmbeddings
48	en	sent.small.bert.L10.256	BertSentenceEmbeddings
49	en	sent.small.bert.L10.512	BertSentenceEmbeddings
50	en	sent.small.bert.L10.768	BertSentenceEmbeddings

Language	Spark NLP Model Name	Model
Number		
51	en	sent_small_bert_L12_128
52	en	BertSentenceEmbeddings
53	en	sent_small_bert_L12_256
54	en	BertSentenceEmbeddings
55	en	sent_small_bert_L12_512
56	en	BertSentenceEmbeddings
57	en	sent_small_bert_L12_768
58	en	BertSentenceEmbeddings
59	en	sent_small_bert_L2_128
60	en	BertSentenceEmbeddings
61	en	sent_small_bert_L2_256
62	en	BertSentenceEmbeddings
63	en	sent_small_bert_L2_512
64	en	BertSentenceEmbeddings
65	en	sent_small_bert_L2_768
66	en	BertSentenceEmbeddings
67	en	sent_small_bert_L4_128
68	en	BertSentenceEmbeddings
69	en	sent_small_bert_L4_256
70	en	BertSentenceEmbeddings
71	en	sent_small_bert_L4_512
72	en	BertSentenceEmbeddings
73	en	sent_small_bert_L4_768
74	en	BertSentenceEmbeddings
75	en	sent_small_bert_L6_128
76	en	BertSentenceEmbeddings
77	en	sent_small_bert_L6_256
78	en	BertSentenceEmbeddings
79	en	sent_small_bert_L6_512
80	en	BertSentenceEmbeddings
81	en	sent_small_bert_L6_768
82	en	BertSentenceEmbeddings
83	en	sent_small_bert_L8_128
84	en	BertSentenceEmbeddings
85	en	sent_small_bert_L8_256
86	en	BertSentenceEmbeddings
87	en	sent_small_bert_L8_512
88	en	BertSentenceEmbeddings
89	en	sent_small_bert_L8_768
90	en	BertSentenceEmbeddings
91	en	covidbert_large_uncased
92	en	BertEmbeddings
93	en	electra_base_uncased
94	en	BertEmbeddings
95	en	electra_large_uncased
96	en	BertEmbeddings
97	en	electra_small_uncased
98	en	BertEmbeddings
99	en	sent_covidbert_large_uncased
100	en	BertSentenceEmbeddings

# Embeddings

in

## Spark NLP

### Part 2

Search them all in the Modelhub  
<https://nlp.johnsnowlabs.com/models>

Language	Spark NLP Model Name	Model
Number		
76	en	small_bert_L12_128
77	en	BertEmbeddings
78	en	small_bert_L12_256
79	en	BertEmbeddings
80	en	small_bert_L12_512
81	en	BertEmbeddings
82	en	small_bert_L12_768
83	en	BertEmbeddings
84	en	small_bert_L4_128
85	en	BertEmbeddings
86	en	small_bert_L4_256
87	en	BertEmbeddings
88	en	small_bert_L4_512
89	en	BertEmbeddings
90	en	small_bert_L4_768
91	en	BertEmbeddings
92	en	small_bert_L6_128
93	en	BertEmbeddings
94	en	small_bert_L6_256
95	en	BertEmbeddings
96	en	small_bert_L6_512
97	en	BertEmbeddings
98	en	small_bert_L6_768
99	en	BertEmbeddings
100	en	small_bert_L8_128
101	en	BertEmbeddings
102	en	small_bert_L8_256
103	en	BertEmbeddings
104	en	small_bert_L8_512
105	en	BertEmbeddings
106	en	small_bert_L8_768
107	en	BertEmbeddings
108	en	covidbert_large_uncased
109	en	BertEmbeddings
110	en	electra_base_uncased
111	en	BertEmbeddings
112	en	electra_large_uncased
113	en	BertEmbeddings
114	en	electra_small_uncased
115	en	BertEmbeddings
116	en	sent_covidbert_large_uncased
117	en	BertSentenceEmbeddings
118	en	sent_electra_base_uncased
119	en	BertSentenceEmbeddings

Language	Spark NLP Model Name	Model	Language	Spark NLP Model Name	Model		
Number			Number				
101	en	sent_electra_large_uncased	BertSentenceEmbeddings	126	xx	tflhub_use_multi_lg	UniversalSentenceEncoder
102	en	sent_electra_small_uncased	BertSentenceEmbeddings	127	xx	tflhub_use_multi	UniversalSentenceEncoder
103	fi	bert_finnish_cased	BertEmbeddings	128	he	hebrew_cc_300d	WordEmbeddingsModel
104	fi	bert_finnish_uncased	BertEmbeddings	129	hi	hindi_cc_300d	WordEmbeddingsModel
105	de	w2v_cc_300d	WordEmbeddingsModel	130	bn	bengali_cc_300d	WordEmbeddingsModel
106	en	biobert_clinical_base_cased	BertEmbeddings	131	xx	tflhub_use_multi_lg	UniversalSentenceEncoder
107	en	biobert_discharge_base_cased	BertEmbeddings	132	xx	tflhub_use_multi	UniversalSentenceEncoder
108	en	biobert_pmc_base_cased	BertEmbeddings	133	en	sbert_jsl_medium_umls_uncased	BertSentenceEmbeddings
109	en	biobert_pubmed_base_cased	BertEmbeddings	134	en	sbert_jsl_medium_umls	BertSentenceEmbeddings
110	en	biobert_pubmed_large_cased	BertEmbeddings	135	en	sbert_jsl_mini_umls_uncased	BertSentenceEmbeddings
111	en	biobert_pubmed_pmc_base_cased	BertEmbeddings	136	en	sbert_jsl_mini_umls	BertSentenceEmbeddings
112	en	sent_biobert_clinical_base_cased	BertSentenceEmbeddings	137	en	sbert_jsl_tiny_umls_uncased	BertSentenceEmbeddings
113	en	sent_biobert_discharge_base_cased	BertSentenceEmbeddings	138	en	sbert_jsl_tiny_umls	BertSentenceEmbeddings
114	en	sent_biobert_pmc_base_cased	BertSentenceEmbeddings	139	en	sbiobert_jsl_cased	BertSentenceEmbeddings
115	en	sent_biobert_pubmed_base_cased	BertSentenceEmbeddings	140	en	sbiobert_jsl_umls_cased	BertSentenceEmbeddings
116	en	sent_biobert_pubmed_large_cased	BertSentenceEmbeddings	141	zh	bert_base_chinese	BertEmbeddings
117	en	sent_biobert_pubmed_pmc_base_cased	BertSentenceEmbeddings	142	nl	bert_base_dutch_cased	BertEmbeddings
118	en	labse	BertSentenceEmbeddings	143	de	bert_base_german_cased	BertEmbeddings
119	pt	bert_portuguese_base_cased	BertEmbeddings	144	de	bert_base_german_umls	BertEmbeddings
120	pt	bert_portuguese_large_cased	BertEmbeddings	145	it	bert_base_italian_cased	BertEmbeddings
121	en	sbiobert_base_cased_mli	BertSentenceEmbeddings	146	it	bert_base_italian_umls	BertEmbeddings
122	en	sbluebert_base_uncased_mli	BertSentenceEmbeddings	147	xx	bert_base_multilingual_cased	BertEmbeddings
123	ur	urduvec_140M_300d	WordEmbeddingsModel	148	xx	bert_base_multilingual_umls	BertEmbeddings
124	ar	arabic_w2v_cc_300d	WordEmbeddingsModel	149	tr	bert_base_turkish_cased	BertEmbeddings
125	fa	persian_w2v_cc_300d	WordEmbeddingsModel	150	tr	bert_base_turkish_umls	BertEmbeddings

# Embeddings

in

## Spark NLP

### Part 3

Search them all in the Modelshub  
<https://nlp.johnsnowlabs.com/models>

Number	Language	Spark NLP Model Name	Model
151	zh	chinese_bert_wwm	BertEmbeddings
152	en	distilbert_base_cased	DistilBertEmbeddings
153	xx	distilbert_base_multilingual_cased	DistilBertEmbeddings
154	en	distilbert_base_uncased	DistilBertEmbeddings
155	en	distilroberta_base	RoBERTaEmbeddings
156	en	roberta_base	RoBERTaEmbeddings
157	en	roberta_large	RoBERTaEmbeddings
158	xx	twitter_xlm_roberta_base	XLMRoBERTaEmbeddings
159	xx	xlm_roberta_base	XLMRoBERTaEmbeddings
160	en	albert_base_uncased	AlbertEmbeddings
161	en	albert_large_uncased	AlbertEmbeddings
162	en	albert_xlarge_uncased	AlbertEmbeddings
163	en	albert_xxlarge_uncased	AlbertEmbeddings
164	en	sbert_jsl_medium_umls_uncased	BertSentenceEmbeddings
165	en	sbert_jsl_medium_uncased	BertSentenceEmbeddings
166	en	sbert_jsl_mini_umls_uncased	BertSentenceEmbeddings
167	en	sbert_jsl_mini_uncased	BertSentenceEmbeddings
168	en	sbert_jsl_tiny_umls_uncased	BertSentenceEmbeddings
169	en	sbert_jsl_tiny_uncased	BertSentenceEmbeddings
170	en	sbiobert_jsl_cased	BertSentenceEmbeddings
171	en	sbiobert_jsl_umls_cased	BertSentenceEmbeddings
172	zh	chinese_xlnet_base	XlnetEmbeddings
173	en	xlnet_base_cased	XlnetEmbeddings
174	en	xlnet_large_cased	XlnetEmbeddings
175	xx	xlm_roberta_xtreme_base	XLMRoBERTaEmbeddings

# Embeddings

in

## Spark NLP

### Part 4

Search them all in the Modelshub  
<https://nlp.johnsnowlabs.com/models>

Number	Language	Spark NLP Model Name	Model
176	en	sent_bert_use_cmlm_en_base	BertSentenceEmbeddings
177	en	sent_bert_use_cmlm_en_large	BertSentenceEmbeddings
178	xx	sent_bert_use_cmlm_multi_base_br	BertSentenceEmbeddings
179	xx	sent_bert_use_cmlm_multi_base	BertSentenceEmbeddings
180	en	longformer_base_4096	LongformerEmbeddings
181	en	longformer_large_4096	LongformerEmbeddings
182	xx	bert_muril	BertEmbeddings
183	en	bert_pubmed	BertEmbeddings
184	en	bert_pubmed_squad2	BertEmbeddings
185	en	bert_wiki_books	BertEmbeddings
186	en	bert_wiki_books_mnli	BertEmbeddings
187	en	bert_wiki_books_qnli	BertEmbeddings
188	en	bert_wiki_books_qqp	BertEmbeddings
189	en	bert_wiki_books_squad2	BertEmbeddings
190	en	bert_wiki_books_sst2	BertEmbeddings
191	te	sentence_detector_dl	SentenceDetectorDLModel
192	en	sent_bert_pubmed	BertSentenceEmbeddings
193	en	sent_bert_pubmed_squad2	BertSentenceEmbeddings
194	en	sent_bert_wiki_books	BertSentenceEmbeddings
195	en	sent_bert_wiki_books_mnli	BertSentenceEmbeddings
196	en	sent_bert_wiki_books_qnli	BertSentenceEmbeddings
197	en	sent_bert_wiki_books_qqp	BertSentenceEmbeddings
198	en	sent_bert_wiki_books_squad2	BertSentenceEmbeddings
199	en	sent_bert_wiki_books_sst2	BertSentenceEmbeddings
200	xx	sent_bert_muril	BertSentenceEmbeddings

Language	Spark NLP Model Name	Model	Language	Spark NLP Model Name	Model		
Number			Number				
201	xx	sent_xlm_roberta_base	XlmRoBERTaForTokenClassification	226	rw	sent_xlm_roberta_base_finetuned_kinyarwanda	XlmRoBERTaSentenceEmbeddings
202	es	sent_bert_base_cased	BertSentenceEmbeddings	227	lg	sent_xlm_roberta_base_finetuned_luganda	XlmRoBERTaSentenceEmbeddings
203	nl	sent_bert_base_cased	BertSentenceEmbeddings	228	pcm	sent_xlm_roberta_base_finetuned_naija	XlmRoBERTaSentenceEmbeddings
204	sv	sent_bert_base_cased	BertSentenceEmbeddings	229	sw	sent_xlm_roberta_base_finetuned_swahili	XlmRoBERTaSentenceEmbeddings
205	el	sent_bert_base_uncased	BertSentenceEmbeddings	230	wo	sent_xlm_roberta_base_finetuned_wolof	XlmRoBERTaSentenceEmbeddings
206	es	sent_bert_base_uncased	BertSentenceEmbeddings	231	yo	sent_xlm_roberta_base_finetuned_yoruba	XlmRoBERTaSentenceEmbeddings
207	en	sent_bert_base_uncased_legal	BertSentenceEmbeddings	232	lou	xlm_roberta_base_finetuned_luo	XlmRoBERTaEmbeddings
208	es	bert_base_cased	BertEmbeddings	233	pcm	xlm_roberta_base_finetuned_naija	XlmRoBERTaEmbeddings
209	nl	bert_base_cased	BertEmbeddings	234	sw	xlm_roberta_base_finetuned_swahili	XlmRoBERTaEmbeddings
210	sv	bert_base_cased	BertEmbeddings	235	wo	xlm_roberta_base_finetuned_wolof	XlmRoBERTaEmbeddings
211	el	bert_base_uncased	BertEmbeddings	236	yo	xlm_roberta_base_finetuned_yoruba	XlmRoBERTaEmbeddings
212	es	bert_base_uncased	BertEmbeddings	237	es	roberta_base_biomedical	RoBERTaEmbeddings
213	en	bert_base_uncased_legal	BertEmbeddings	238	en	doc2vec_gigaword_300	Doc2VecModel
214	ja	japanese_cc_300d	WordEmbeddingsModel	239	en	doc2vec_gigaword_wiki_300	Doc2VecModel
215	ro	bert_base_cased	BertEmbeddings	240	te	distilbert_uncased	DistilBERTEmbeddings
216	de	sent_bert_base_cased	BertSentenceEmbeddings	241	en	sbert_jsl_medium_rxnorm_uncased	BertSentenceEmbeddings
217	am	xlm_roberta_base_finetuned_amharic	XlmRoBERTaEmbeddings	242	fi	bert_base_finnish_cased	BertSentenceEmbeddings
218	ha	xlm_roberta_base_finetuned_hausa	XlmRoBERTaEmbeddings	243	fi	bert_base_finnish_uncased	BertSentenceEmbeddings
219	ig	xlm_roberta_base_finetuned_igbo	XlmRoBERTaEmbeddings	244	en	sbert_jsl_medium_rxnorm_uncased	BertSentenceEmbeddings
220	rw	xlm_roberta_base_finetuned_kinyarwanda	XlmRoBERTaEmbeddings	245	en	word2vec_gigaword_300	Word2VecModel
221	lg	xlm_roberta_base_finetuned_luganda	XlmRoBERTaEmbeddings	246	en	word2vec_gigaword_wiki_300	Word2VecModel
222	xx	xlm_roberta_large	XlmRoBERTaEmbeddings	247	en	electra_medal_acronym	BertEmbeddings
223	am	sent_xlm_roberta_base_finetuned_amharic	XlmRoBERTaSentenceEmbeddings	248	vi	distilbert_base_cased	DistilBERTEmbeddings
224	ha	sent_xlm_roberta_base_finetuned_hausa	XlmRoBERTaSentenceEmbeddings				
225	ig	sent_xlm_roberta_base_finetuned_igbo	XlmRoBERTaSentenceEmbeddings				

# Embeddings

in

## Spark NLP

### Part 5

Search them all in the Modelshub  
<https://nlp.johnsnowlabs.com/models>

# Transformer

## Sequence

## Classifiers

in

## Spark NLP

Search them all in the Modelshub  
<https://nlp.johnsnowlabs.com/models>

Number	Language	Spark NLP Model Name	Model
0	en	bert_base_sequence_classifier_dbpedia_14	BertForSequenceClassification
1	en	bert_base_sequence_classifier_imdb	BertForSequenceClassification
2	en	bert_large_sequence_classifier_imdb	BertForSequenceClassification
3	fr	bert_multilingual_sequence_classifier.allocine	BertForSequenceClassification
4	en	bert_base_sequence_classifier_ag_news	BertForSequenceClassification
5	es	bert_sequence_classifier_beto_emotion_analysis	BertForSequenceClassification
6	es	bert_sequence_classifier_beto_sentiment_analysis	BertForSequenceClassification
7	en	bert_sequence_classifier_dehatebert_mono	BertForSequenceClassification
8	en	bert_sequence_classifier_fibert	BertForSequenceClassification
9	ja	bert_sequence_classifier_japanese_sentiment	BertForSequenceClassification
10	xx	bert_sequence_classifier_multilingual_sentiment	BertForSequenceClassification
11	ru	bert_sequence_classifier_rubert_sentiment	BertForSequenceClassification
12	de	bert_sequence_classifier_sentiment	BertForSequenceClassification
13	tr	bert_sequence_classifier_turkish_sentiment	BertForSequenceClassification
14	en	bert_sequence_classifier_question_statement	BertForSequenceClassification
15	en	bert_sequence_classifier_question_statement_cl...	BertForSequenceClassification
16	en	bert_sequence_classifier_antisemitism	BertForSequenceClassification
17	en	bert_sequence_classifier_hatexplain	BertForSequenceClassification
18	en	bert_sequence_classifier_trec_coarse	BertForSequenceClassification
19	en	bert_sequence_classifier_age_news	BertForSequenceClassification
20	en	bert_sequence_classifier_banking77	BertForSequenceClassification
21	en	bert_sequence_classifier_sms_spam	BertForSequenceClassification
22	en	bert_sequence_classifier_song_lyrics	BertForSequenceClassification
23	en	distilbert_base_sequence_classifier_ag_news	DistilBertForSequenceClassification
24	en	distilbert_base_sequence_classifier_amazon_pol...	DistilBertForSequenceClassification

Number	Language	Spark NLP Model Name	Model
25	en	distilbert_base_sequence_classifier_imdb	DistilBertForSequenceClassification
26	ur	distilbert_base_sequence_classifier_imdb	DistilBertForSequenceClassification
27	fr	distilbert_multilingual_sequence_classifier_al...	DistilBertForSequenceClassification
28	en	distilbert_sequence_classifier_banking77	DistilBertForSequenceClassification
29	en	distilbert_sequence_classifier_emotion	DistilBertForSequenceClassification
30	en	distilbert_sequence_classifier_industry	DistilBertForSequenceClassification
31	en	distilbert_sequence_classifier_policy	DistilBertForSequenceClassification
32	en	distilbert_sequence_classifier_sst2	DistilBertForSequenceClassification
33	en	albert_base_sequence_classifier_ag_news	AlbertForSequenceClassification
34	en	albert_base_sequence_classifier_imdb	AlbertForSequenceClassification
35	en	longformer_base_sequence_classifier_ag_news	LongformerForSequenceClassification
36	en	longformer_base_sequence_classifier_imdb	LongformerForSequenceClassification
37	en	roberta_base_sequence_classifier_ag_news	RoBERTaForSequenceClassification
38	en	roberta_base_sequence_classifier_imdb	RoBERTaForSequenceClassification
39	en	xlm_roberta_base_sequence_classifier_ag_news	XlmRoBERTaForSequenceClassification
40	fr	xlm_roberta_base_sequence_classifier_allocine	XlmRoBERTaForSequenceClassification
41	en	xlm_roberta_base_sequence_classifier_imdb	XlmRoBERTaForSequenceClassification
42	en	xlnet_base_sequence_classifier_ag_news	XlnetForSequenceClassification
43	en	xlnet_base_sequence_classifier_imdb	XlnetForSequenceClassification

Language	Spark NLP Model Name	Model
Number		
0	en bert_base_token_classifier_conll03	BertForTokenClassification
1	en bert_base_token_classifier_ontonote	BertForTokenClassification
2	en bert_large_token_classifier_conll03	BertForTokenClassification
3	en bert_large_token_classifier_ontonote	BertForTokenClassification
4	fa bert_token_classifier_parsbert_armanner	BertForTokenClassification
5	fa bert_token_classifier_parsbert_ner	BertForTokenClassification
6	fa bert_token_classifier_parsbert_peymanner	BertForTokenClassification
7	es bert_token_classifier_spanish_ner	BertForTokenClassification
8	sv bert_token_classifier_swedish_ner	BertForTokenClassification
9	tr bert_token_classifier_turkish_ner	BertForTokenClassification
10	en distilbert_base_token_classifier_conll03	DistilBertForTokenClassification
11	en distilbert_base_token_classifier_ontonotes	DistilBertForTokenClassification
12	fa distilbert_token_classifier_persian_ner	DistilBertForTokenClassification
13	en bert_base_token_classifier_few_nerd	BertForTokenClassification
14	en distilbert_base_token_classifier_few_nerd	DistilBertForTokenClassification
15	en bert_token_classifier_ner_clinical	MedicalBertForTokenClassifier
16	en bert_token_classifier_ner_jsl	MedicalBertForTokenClassifier
17	en bert_token_classifier_ner_btc	BertForTokenClassification
18	ja bert_token_classifier_ner_ud_gsd	BertForTokenClassification
19	en bert_token_classifier_ner_deid	MedicalBertForTokenClassifier
20	en bert_token_classifier_ner_jsl	MedicalBertForTokenClassifier
21	en bert_token_classifier_ner_drugs	MedicalBertForTokenClassifier
22	en bert_token_classifier_ner_jsl_slim	MedicalBertForTokenClassifier
23	en albert_base_token_classifier_conll03	AlbertForTokenClassification
24	en albert_large_token_classifier_conll03	AlbertForTokenClassification

# Transformer Token Classifiers in Spark NLP Part 1

Search them all in the Modelshub  
<https://nlp.johnsnowlabs.com/models>

Language	Spark NLP Model Name	Model
Number		
26	en distilroberta_base_token_classifier_ontonotes	RoBertaForTokenClassification
27	en roberta_base_token_classifier_conll03	RoBertaForTokenClassification
28	en roberta_base_token_classifier_ontonotes	RoBertaForTokenClassification
29	en roberta_large_token_classifier_conll03	RoBertaForTokenClassification
30	en roberta_large_token_classifier_ontonotes	RoBertaForTokenClassification
31	fa roberta_token_classifier_zwnj_base_ner	RoBertaForTokenClassification
32	xx xlm_roberta_token_classifier_ner_40_lang	XlmRoBertaForTokenClassification
33	en xlnet_base_token_classifier_conll03	XlnetForTokenClassification
34	en xlnet_large_token_classifier_conll03	XlnetForTokenClassification
35	en bert_token_classifier_ner_ade	MedicalBertForTokenClassifier
36	en bert_token_classifier_ner_anatomy	MedicalBertForTokenClassifier
37	en bert_token_classifier_ner_bacteria	MedicalBertForTokenClassifier
38	en xlm_roberta_base_token_classifier_conll03	XlmRoBertaForTokenClassification
39	en xlm_roberta_base_token_classifier_ontonotes	XlmRoBertaForTokenClassification
40	en longformer_base_token_classifier_conll03	LongformerForTokenClassification
41	en longformer_large_token_classifier_conll03	LongformerForTokenClassification
42	en bert_token_classifier_ner_chemicals	MedicalBertForTokenClassifier
43	en bert_token_classifier_ner_chemprot	MedicalBertForTokenClassifier
44	en bert_token_classifier_ner_bionlp	MedicalBertForTokenClassifier
45	en bert_token_classifier_ner_cellular	MedicalBertForTokenClassifier
46	tr xlm_roberta_base_token_classifier_ner	XlmRoBertaForTokenClassification
47	id xlm_roberta_large_token_classification_ner	XlmRoBertaForTokenClassification
48	is roberta_token_classifier_icelandic_ner	RoBertaForTokenClassification
49	xx xlm_roberta_large_token_classifier_masakhaner	XlmRoBertaForTokenClassification
50	zh bert_token_classifier_chinese_ner	BertForTokenClassification

Language		Spark NLP Model Name	Model
Number			
0	en	google_t5_small_ssm_nq	T5Transformer
1	en	google_t5_small_ssm_nq	T5Transformer
2	en	bert_base_cased_qa_squad2	BertForQuestionAnswering
3	en	bert_qa_Bertv1_fine	BertForQuestionAnswering
4	en	bert_qa_COVID_BERTa	BertForQuestionAnswering
5	en	bert_qa_COVID_BERTb	BertForQuestionAnswering
6	en	bert_qa_COVID_BERTc	BertForQuestionAnswering
7	de	bert_qa_GBERTQnA	BertForQuestionAnswering
8	en	bert_qa_HomayounSadri_bert_base_uncased_finetu...	BertForQuestionAnswering
9	id	bert_qa_Indobert_QA	BertForQuestionAnswering
10	en	bert_qa_Klue_CommonSense_model	BertForQuestionAnswering
11	en	bert_qa_MTL_bert_base_uncased_ww_squad	BertForQuestionAnswering
12	en	bert_qa_ManuERT_for_xqua	BertForQuestionAnswering
13	en	bert_qa_MiniLM_L12_H384_uncased_finetuned_squad	BertForQuestionAnswering
14	en	bert_qa_Multi_ling_BERT	BertForQuestionAnswering
15	xx	bert_qa_Part_1_mBERT_Model_E1	BertForQuestionAnswering
16	en	bert_qa_Part_1_mBERT_Model_E2	BertForQuestionAnswering
17	en	bert_qa_Part_2_BERT_Multilingual_Dutch_Model_E1	BertForQuestionAnswering
18	en	bert_qa_Part_2_mBERT_Model_E2	BertForQuestionAnswering
19	en	bert_qa_Paul_Vinh_bert_base_multilingual_cased...	BertForQuestionAnswering
20	en	bert_qa_PruebaBert	BertForQuestionAnswering
21	en	bert_qa_SciBERT_SQuAD_QuAC	BertForQuestionAnswering
22	en	bert_qa_Seongkyu_bert_base_cased_finetuned_squad	BertForQuestionAnswering
23	en	bert_qa_Shushant_BiomedNLP_PubMedBERT_base_unc...	BertForQuestionAnswering
24	en	bert_qa_Spanbert_emotion_extraction	BertForQuestionAnswering

# Transformer Question Answering in Spark NLP Part 1

Search them all in the Modelshub  
<https://nlp.johnsnowlabs.com/models>

Language		Spark NLP Model Name	Model
Number			
25	en	bert_qa_SreyanG_NVIDIA_bert_base_cased_finetun...	BertForQuestionAnswering
26	en	bert_qa_SreyanG_NVIDIA_bert_base_uncased_finet...	BertForQuestionAnswering
27	en	bert_qa_SupriyaArun_bert_base_uncased_finetune...	BertForQuestionAnswering
28	en	bert_qa_Tianle_bert_base_uncased_finetuned_squad	BertForQuestionAnswering
29	en	bert_qa_Trial_3_Results	BertForQuestionAnswering
30	ko	bert_qa_ainize_klue_bert_base_mrc	BertForQuestionAnswering
31	en	bert_qa_andresestevez_bert_base_cased_finetune...	BertForQuestionAnswering
32	en	bert_qa_araspeedest	BertForQuestionAnswering
33	ar	bert_qa_arap_qa_bert	BertForQuestionAnswering
34	ar	bert_qa_arap_qa_bert_large_v2	BertForQuestionAnswering
35	ar	bert_qa_arap_qa_bert_v2	BertForQuestionAnswering
36	en	bert_qa_augmented_Squad_Translated	BertForQuestionAnswering
37	en	bert_qa_augmented	BertForQuestionAnswering
38	en	bert_qa_batterybert_cased_squad_v1	BertForQuestionAnswering
39	en	bert_qa_batterybert_uncased_squad_v1	BertForQuestionAnswering
40	en	bert_qa_batterydata_bert_base_uncased_squad_v1	BertForQuestionAnswering
41	en	bert_qa_batteryonlybert_cased_squad_v1	BertForQuestionAnswering
42	en	bert_qa_batteryonlybert_uncased_squad_v1	BertForQuestionAnswering
43	en	bert_qa_batteryscibert_cased_squad_v1	BertForQuestionAnswering
44	en	bert_qa_batteryscibert_uncased_squad_v1	BertForQuestionAnswering
45	en	bert_qa_bdickson_bert_base_uncased_finetuned_s...	BertForQuestionAnswering
46	en	bert_qa_bert_FT_new_newsqa	BertForQuestionAnswering
47	en	bert_qa_bert_FT_newsqa	BertForQuestionAnswering
48	en	bert_qa_bert_all	BertForQuestionAnswering
49	en	bert_qa_bert_all_squad_all_translated	BertForQuestionAnswering

Number	Language	Spark NLP Model Name	Model
51	en	bert_qa_bert_all_squad_que_translated	BertForQuestionAnswering
52	en	bert_qa_bert_all_translated	BertForQuestionAnswering
53	en	bert_qa_bert_base_1024_full_trivia_copied_embe...	BertForQuestionAnswering
54	en	bert_qa_bert_base_2048_full_trivia_copied_embe...	BertForQuestionAnswering
55	en	bert_qa_bert_base_4096_full_trivia_copied_embe...	BertForQuestionAnswering
56	en	bert_qa_bert_base_512_full_trivia	BertForQuestionAnswering
57	en	bert_qa_bert_base_cased_IUChatbot_ontologyDts	BertForQuestionAnswering
58	en	bert_qa_bert_base_cased_chaii	BertForQuestionAnswering
59	en	bert_qa_bert_base_cased_finetuned_squad_test	BertForQuestionAnswering
60	pt	bert_qa_bert_base_cased_squad_v1.1_portuguese	BertForQuestionAnswering
61	en	bert_qa_bert_base_cased_squad_v1	BertForQuestionAnswering
62	zh	bert_qa_bert_base_chinese_finetuned_squad_colab	BertForQuestionAnswering
63	fa	bert_qa_bert_base_fa_qa	BertForQuestionAnswering
64	en	bert_qa_bert_base_faquad	BertForQuestionAnswering
65	en	bert_qa_bert_base_finetuned_squad2	BertForQuestionAnswering
66	th	bert_qa_bert_base_multilingual_cased_finetune_qa	BertForQuestionAnswering
67	en	bert_qa_bert_base_multilingual_cased_finetuned...	BertForQuestionAnswering
68	nl	bert_qa_bert_base_multilingual_cased_finetuned...	BertForQuestionAnswering
69	en	bert_qa_bert_base_multilingual_cased_finetuned...	BertForQuestionAnswering
70	pl	bert_qa_bert_base_multilingual_cased_finetuned...	BertForQuestionAnswering
71	pl	bert_qa_bert_base_multilingual_cased_finetuned...	BertForQuestionAnswering
72	en	bert_qa_bert_base_multilingual_cased_finetuned...	BertForQuestionAnswering
73	en	bert_qa_bert_base_multilingual_cased_korquad	BertForQuestionAnswering
74	en	bert_qa_bert_base_multilingual_cased_korquad_v1	BertForQuestionAnswering
75	en	bert_qa_bert_base_multilingual_uncased_finetun...	BertForQuestionAnswering

# Transformer Question Answering in Spark NLP Part 2

Search them all in the Modelshub  
<https://nlp.johnsnowlabs.com/models>

Number	Language	Spark NLP Model Name	Model
76	en	bert_qa_bert_base_multilingual_xquad	BertForQuestionAnswering
77	si	bert_qa_bert_base_sinhala_qa	BertForQuestionAnswering
78	en	bert_qa_bert_base_spanish_wwm_cased_finetuned...	BertForQuestionAnswering
79	en	bert_qa_bert_base_spanish_wwm_cased_finetuned...	BertForQuestionAnswering
80	en	bert_qa_bert_base_spanish_wwm_cased_finetuned...	BertForQuestionAnswering
81	es	bert_qa_bert_base_spanish_wwm_cased_finetuned...	BertForQuestionAnswering
82	es	bert_qa_bert_base_spanish_wwm_cased_finetuned...	BertForQuestionAnswering
83	es	bert_qa_bert_base_spanish_wwm_cased_finetuned...	BertForQuestionAnswering
84	es	bert_qa_bert_base_spanish_wwm_cased_finetuned...	BertForQuestionAnswering
85	en	bert_qa_bert_base_spanish_wwm_uncased_finetune...	BertForQuestionAnswering
86	en	bert_qa_bert_base_spanish_wwm_uncased_finetune...	BertForQuestionAnswering
87	en	bert_qa_bert_base_spanish_wwm_uncased_finetune...	BertForQuestionAnswering
88	en	bert_qa_bert_base_squadv1	BertForQuestionAnswering
89	en	bert_qa_bert_base_swedish_cased_squad_experime...	BertForQuestionAnswering
90	sv	bert_qa_bert_base_swedish_squad2	BertForQuestionAnswering
91	en	bert_qa_bert_base_turkish_cased_finetuned_lr_...	BertForQuestionAnswering
92	tr	bert_qa_bert_base_turkish_squad	BertForQuestionAnswering
93	en	bert_qa_bert_base_uncased_coqa	BertForQuestionAnswering
94	en	bert_qa_bert_base_uncased_few_shot_k_1024_fine...	BertForQuestionAnswering
95	en	bert_qa_bert_base_uncased_few_shot_k_128_finet...	BertForQuestionAnswering
96	en	bert_qa_bert_base_uncased_few_shot_k_16_finetu...	BertForQuestionAnswering
97	en	bert_qa_bert_base_uncased_few_shot_k_256_finet...	BertForQuestionAnswering
98	en	bert_qa_bert_base_uncased_few_shot_k_32_finetu...	BertForQuestionAnswering
99	en	bert_qa_bert_base_uncased_few_shot_k_512_finet...	BertForQuestionAnswering

Number					
100	en	bert_qa_bert_base_uncased_few_shot_k_64_finetu...	BertForQuestionAnswering		
101	en	bert_qa_bert_base_uncased_finetuned_docvqa	BertForQuestionAnswering		
102	en	bert_qa_bert_base_uncased_finetuned_duorc_bert	BertForQuestionAnswering		
103	en	bert_qa_bert_base_uncased_finetuned_infovqa	BertForQuestionAnswering		
104	en	bert_qa_bert_base_uncased_finetuned_newsqa	BertForQuestionAnswering		
105	en	bert_qa_bert_base_uncased_finetuned_squad_froz...	BertForQuestionAnswering		
106	en	bert_qa_bert_base_uncased_finetuned_squad_v1	BertForQuestionAnswering		
107	en	bert_qa_bert_base_uncased_finetuned_squad_v2	BertForQuestionAnswering		
108	en	bert_qa_bert_base_uncased_finetuned_vi_infovqa	BertForQuestionAnswering		
109	en	bert_qa_bert_base_uncased_fiqqa_film_sq_filt	BertForQuestionAnswering		
110	en	bert_qa_bert_base_uncased_qa_squad2	BertForQuestionAnswering		
111	en	bert_qa_bert_base_uncased_squad1.1_block_spars...	BertForQuestionAnswering		
112	en	bert_qa_bert_base_uncased_squad1.1_block_spars...	BertForQuestionAnswering		
113	en	bert_qa_bert_base_uncased_squad1.1_block_spars...	BertForQuestionAnswering		
114	en	bert_qa_bert_base_uncased_squad1.1_block_spars...	BertForQuestionAnswering		
115	en	bert_qa_bert_base_uncased_squad1.1_pruned_x3.2_v2	BertForQuestionAnswering		
116	en	bert_qa_bert_base_uncased_squad2_covid_qa_deepset	BertForQuestionAnswering		
117	en	bert_qa_bert_base_uncased_squad_L3	BertForQuestionAnswering		
118	en	bert_qa_bert_base_uncased_squad_L6	BertForQuestionAnswering		
119	en	bert_qa_bert_base_uncased_squad_v1_sparse0.25	BertForQuestionAnswering		
120	en	bert_qa_bert_base_uncased_squadv1.1_sparse_80_...	BertForQuestionAnswering		
121	en	bert_qa_bert_base_uncased_squadv1_x1.16_188.1_...	BertForQuestionAnswering		
122	en	bert_qa_bert_base_uncased_squadv1_x1.84_f88.7_...	BertForQuestionAnswering		
123	en	bert_qa_bert_base_uncased_squadv1_x1.96_f88.3_...	BertForQuestionAnswering		
124	en	bert_qa_bert_base_uncased_squadv1_x2.01_f89.2_...	BertForQuestionAnswering		
125	en	bert_qa_bert_base_uncased_squadv1_x2.32_f86.6_...	BertForQuestionAnswering		

# Transformer Question Answering in Spark NLP Part 3

Search them all in the Modelshub  
<https://nlp.johnsnowlabs.com/models>

Number	Language	Spark NLP Model Name	Model
126	en	bert_qa_bert_base_uncased_squadv1_x2.44_f87.7_...	BertForQuestionAnswering
127	zh	bert_qa_bert_chinese_finetuned	BertForQuestionAnswering
128	en	bert_qa_bert	BertForQuestionAnswering
129	en	bert_qa_bert_fa_QA_v1	BertForQuestionAnswering
130	en	bert_qa_bert_large_fquad	BertForQuestionAnswering
131	en	bert_qa_bert_large_uncased_www_squadv2_x2.63_f...	BertForQuestionAnswering
132	en	bert_qa_bert_medium_finetuned_squad	BertForQuestionAnswering
133	en	bert_qa_bert_medium_finetuned_squadv2	BertForQuestionAnswering
134	en	bert_qa_bert_medium_pretrained_finetuned_squad	BertForQuestionAnswering
135	en	bert_qa_bert_medium_squad2_distilled	BertForQuestionAnswering
136	en	bert_qa_bert_medium_wslb_finetuned_squadv1	BertForQuestionAnswering
137	en	bert_qa_bert_mini_5_finetuned_squadv2	BertForQuestionAnswering
138	en	bert_qa_bert_mini_finetuned_squadv2	BertForQuestionAnswering
139	en	bert_qa_bert_mini_wslb_finetuned_squadv1	BertForQuestionAnswering
140	en	bert_qa_bert_multi_cased_finetuned_xquad_chaii	BertForQuestionAnswering
141	en	bert_qa_bert_multi_cased_finetuned_chaii	BertForQuestionAnswering
142	en	bert_qa_bert_multi_cased_finetuned_xquadv1_fin...	BertForQuestionAnswering
143	xx	bert_qa_bert_multi_cased_finetuned_xquadv1	BertForQuestionAnswering
144	de	bert_qa_bert_multi_english_german_squad2	BertForQuestionAnswering
145	en	bert_qa_bert_multi_uncased_finetuned_chaii	BertForQuestionAnswering
146	xx	bert_qa_bert_multi_uncased_finetuned_xquadv1	BertForQuestionAnswering
147	en	bert_qa_bert_gasper	BertForQuestionAnswering
148	en	bert_qa_bert_reader_squad2	BertForQuestionAnswering
149	en	bert_qa_bert_set_date_1_lr_2e_5_bs_32_ep_4	BertForQuestionAnswering

Language	Spark NLP Model Name	Model
Number		
150	en	bert_qa_bert_small_2_finetuned_squadv2
151	en	bert_qa_bert_small_cord19_squad2
152	en	bert_qa_bert_small_cord19qa
153	en	bert_qa_bert_small_finetuned_squad
154	en	bert_qa_bert_small_finetuned_squadv2
155	en	bert_qa_bert_small_pretrained_finetuned_squad
156	en	bert_qa_bert_small_wrsib_finetuned_squadv1
157	en	bert_qa_bert_tiny_2_finetuned_squadv2
158	en	bert_qa_bert_tiny_3_finetuned_squadv2
159	en	bert_qa_bert_tiny_4_finetuned_squadv2
160	en	bert_qa_bert_tiny_5_finetuned_squadv2
161	en	bert_qa_bert_tiny_finetuned_squad
162	en	bert_qa_bert_tiny_finetuned_squadv2
163	tr	bert_qa_bert_turkish_question_answering
164	en	bert_qa_bert_uncased_L_10_H_512_A_8_cord19_200...
165	en	bert_qa_bert_uncased_L_10_H_512_A_8_cord19_200...
166	en	bert_qa_bert_uncased_L_10_H_512_A_8_squad2_cov...
167	en	bert_qa_bert_uncased_L_10_H_512_A_8_squad2
168	en	bert_qa_bert_uncased_L_2_H_512_A_8_cord19_200...
169	en	bert_qa_bert_uncased_L_2_H_512_A_8_squad2_cov...
170	en	bert_qa_bert_uncased_L_2_H_512_A_8_squad2
171	en	bert_qa_bert_uncased_L_4_H_256_A_4_cord19_200...
172	en	bert_qa_bert_uncased_L_4_H_256_A_4_cord19_200...
173	en	bert_qa_bert_uncased_L_4_H_256_A_4_squad2_cov...
174	en	bert_qa_bert_uncased_L_4_H_256_A_4_squad2

# Transformer Question Answering in Spark NLP Part 4

Search them all in the Modelshub  
<https://nlp.johnsnowlabs.com/models>

Number			
175	en	bert_qa_bert_uncased_L_4_H_512_A_8_cord19_200...	BertForQuestionAnswering
176	en	bert_qa_bert_uncased_L_4_H_512_A_8_cord19_200...	BertForQuestionAnswering
177	en	bert_qa_bert_uncased_L_4_H_512_A_8_squad2_cov...	BertForQuestionAnswering
178	en	bert_qa_bert_uncased_L_4_H_512_A_8_squad2	BertForQuestionAnswering
179	en	bert_qa_bert_uncased_L_4_H_768_A_12_cord19_200...	BertForQuestionAnswering
180	en	bert_qa_bert_uncased_L_4_H_768_A_12_cord19_200...	BertForQuestionAnswering
181	en	bert_qa_bert_uncased_L_4_H_768_A_12_squad2_cov...	BertForQuestionAnswering
182	en	bert_qa_bert_uncased_L_4_H_768_A_12_squad2	BertForQuestionAnswering
183	en	bert_qa_bert_uncased_L_6_H_128_A_2_squad2_cov...	BertForQuestionAnswering
184	en	bert_qa_bert_uncased_L_6_H_128_A_2_squad2	BertForQuestionAnswering
185	en	bert_qa_bertimbau_squad1.1	BertForQuestionAnswering
186	en	bert_qa_bertserini_bert_base_squad	BertForQuestionAnswering
187	en	bert_qa_bertserini_bert_large_squad	BertForQuestionAnswering
188	ko	bert_qa_bespin_global_klue_bert_base_mrc	BertForQuestionAnswering
189	es	bert_qa_beto_base_spanish_sqac	BertForQuestionAnswering
190	pt	bert_qa_bioBERTpt_squad_v1.1_portuguese	BertForQuestionAnswering
191	en	bert_qa_biobert_base_cased_v1.1_squad	BertForQuestionAnswering
192	en	bert_qa_biobert_base_cased_v1.1_squad_finetune...	BertForQuestionAnswering
193	en	bert_qa_biobert_base_cased_v1.1_squad_finetune...	BertForQuestionAnswering
194	en	bert_qa_biobert_base_cased_v1.1_squad_finetune...	BertForQuestionAnswering
195	en	bert_qa_biobert_bioasq	BertForQuestionAnswering
196	en	bert_qa_biobert_squad2_cased	BertForQuestionAnswering
197	en	bert_qa_biobert_squad2_cased_finetuned_squad	BertForQuestionAnswering
198	en	bert_qa_biobert_v1.1_pubmed_finetuned_squad	BertForQuestionAnswering
199	en	bert_qa_biobert_v1.1_pubmed_squad_v2	BertForQuestionAnswering
200	en	bert_qa_bioformer_cased_v1.0_squad1	BertForQuestionAnswering

# Transformers & Embeddings

**BERT** is a bi-directional transformer for pre-training over a lot of unlabeled textual data to learn a language representation that can be used to fine-tune for specific machine learning tasks. While BERT outperformed the NLP state-of-the-art on several challenging tasks, its performance improvement could be attributed to the bidirectional transformer, novel pre-training tasks of Masked Language Model and Next Structure Prediction along with a lot of data and Google's compute power. It is an Auto Encoder Language Model.

**XLNet** is a large bidirectional transformer that uses improved training methodology, larger data and more computational power to achieve better than BERT prediction metrics on 20 language tasks. To improve the training, XLNet introduces permutation language modeling, where all tokens are predicted but in random order. This is in contrast to BERT's masked language model where only the masked (15%) tokens are predicted. It is an Autoregressive Language Model.

**Albert** is Google's new "ALBERT" language model and achieved state-of-the-art results on three popular benchmark tests for natural language understanding (NLU): GLUE, RACE, and SQuAD 2.0. ALBERT is a "lite" version of Google's 2018 NLU pre training method BERT. Researchers introduced two parameter-reduction techniques in ALBERT to lower memory consumption and increase training speed and the Next Sentence Prediction task is replace by Sentence Order Prediction

**USE (Universal Sentence Encoder)** is a Transformer-based model for encoding sentences into embedding vectors that specifically target transfer learning to other NLP tasks and outperforms previous word-embedding models on various NLP tasks

# Transformers & Embeddings

**DistilBert** is a compressed version of BERT, which leverages knowledge distillation during the pre-training phase and shows that it is possible to reduce the size of a BERT model by 40%, while retaining 97% of its language understanding capabilities and being 60% faster. It introduces a triple loss combining language modeling, distillation and cosine-distance losses. The smaller, faster and lighter model is cheaper to pre-train and we demonstrate its capabilities for on-device computations in a proof-of-concept experiment and a comparative on-device

**RoBerta** is a optimized version of BERT, which has improved hyperparameters and training data size. The findings show that the original BERT was significantly undertrained and with (1) Longer training, bigger batches, (2) removing next sentence prediction objective, (3) training on longer sequences and (4) dynamically changing the masking pattern applied to the training data every previous BERT can be outperformed and new state of the art can be achieved

**XlmRoBerta** is a multilingual BERT model which significantly outperforms multilingual BERT on a variety of cross-lingual benchmark and large accuracy gains on various multi-lingual benchmarks for 88 languages that appear in the Wiki-100 corpus

**Longformer** is a Transformer-based model which improves on processing long sequences by introduces a windowed attention mechanism and a linearly scaling self-attention mechanism which scales linearly with sequence length and easily processes documents with thousands or more tokens.

# NLP Transformers & Embeddings

- BERT : <https://arxiv.org/abs/1810.04805>
- XLNet : <https://arxiv.org/abs/1906.08237>
- Albert : <https://arxiv.org/abs/1909.11942>
- Elmo : <https://arxiv.org/abs/1802.05365>
- USE : <https://arxiv.org/abs/1803.11175>
- DistilBert : <https://arxiv.org/abs/1910.01108>
- RoBerta : <https://arxiv.org/abs/1907.11692>
- DeBerta : <https://arxiv.org/abs/2006.03654>
- XlmRoberta : <https://arxiv.org/abs/1911.02116>
- Longformer : <https://arxiv.org/abs/2004.05150>
- Electra : <https://arxiv.org/abs/2003.10555>
- T5 : <https://arxiv.org/abs/1910.10683>
- Marian: <https://arxiv.org/abs/1804.00344>
- GPT2: <https://openai.com/blog/better-language-models/>
- SpanBERT <https://arxiv.org/abs/1907.10529>
- CamemBERT <https://camembert-model.fr/>
- SciBert <https://www.aclweb.org/anthology/D19-1371/>
- MiniLm <https://arxiv.org/abs/2002.10957>
- CovidBERT <https://arxiv.org/abs/2005.07503>
- BioBERT <https://arxiv.org/abs/1901.08746>
- indoBERT <https://arxiv.org/abs/2011.00677>
- MuRIL <https://arxiv.org/abs/2103.10730>
- sapBERT <https://github.com/cambridgetl/sapbert>
- BioFormer <https://github.com/WGLab/Bioformer>
- LinkBERT <https://arxiv.org/abs/2203.15827>
- MacBERT <https://aclanthology.org/2020.findings-emnlp.58>
- DOC2Vec : <https://arxiv.org/abs/1301.3781>
- Word2Vec: <https://arxiv.org/abs/1301.3781>

# 100+ Languages supported by Language-agnostic BERT Sentence Embedding (LABSE) and XLM-RoBERTa

Train in 1 Language, predict in 100+ different languages



```
# Binary Class Classifier, 2 classes
nlu.load('xx.embed_sentence.labse train.sentiment').fit(train_df).predict(test_df)

# Multi Class Classifier, N classes
nlu.load('xx.embed_sentence.labse train.classifier').fit(train_df).predict(test_df)

# Multi Class Classifier with multiple labels example (i.e. Hashtags)
# N classes, where one row can be assigned up to N labels
nlu.load('xx.embed_sentence.labse train.multi_classifier').fit(train_df).predict(test_df)
```

ISO	NAME	ISO	NAME	ISO	NAME
af	AFRIKAANS	ht	HAITIAN_CREOLE	pt	PORTRUGUESE
am	AMHARIC	hu	HUNGARIAN	ro	ROMANIAN
ar	ARABIC	hy	ARMENIAN	ru	RUSSIAN
as	ASSAMESE	id	INDONESIAN	rw	KINYARWANDA
az	AZERBAIJANI	ig	IGBO	si	SINHALA
be	BELARUSIAN	is	ICELANDIC	sk	SLOVAK
bg	BULGARIAN	it	ITALIAN	sl	SLOVENIAN
bn	BENGALI	ja	JAPANESE	sm	SAMOAN
bo	TIBETAN	jav	JAVANESE	sn	SHONA
bs	BOSNIAN	ka	GEORGIAN	so	SOMALI
ca	CATALAN	kk	KAZAKH	sq	ALBANIAN
ceb	CEBUANO	km	KHMER	sr	SERBIAN
co	CORSICAN	kn	KANNADA	st	SESOTHO
cs	CZECH	ko	KOREAN	su	SUNDANESE
cy	WELSH	ku	KURDISH	sv	SWEDISH
da	DANISH	ky	KYRGYZ	sw	SWAHILI
de	GERMAN	la	LATIN	ta	TAMIL
el	GREEK	lb	LUXEMBOURGISH	te	TELUGU
en	ENGLISH	lo	LAOTHIAN	tg	TAJIK
eo	ESPERANTO	lt	LITHUANIAN	th	THAI
es	SPANISH	lv	LATVIAN	tk	TURKMEN
et	ESTONIAN	mg	MALAGASY	tl	TAGALOG
eu	BASQUE	mi	MAORI	tr	TURKISH
fa	PERSIAN	mk	MACEDONIAN	tt	TATAR
fi	FINNISH	ml	MALAYALAM	ug	UIGHUR
fr	FRENCH	mn	MONGOLIAN	uk	UKRAINIAN
fy	FRISIAN	mr	MARATHI	ur	URDU
ga	IRISH	ms	MALAY	uz	UZBEK
gd	SCOTS_GAELIC	mt	MALTESE	vi	VietNAMESE
gl	Galician	my	BURMESE	wo	WOLOF
gu	GUARATI	ne	NEPALI	xh	XHOSA
ha	HAUSA	nl	DUTCH	yi	YIDDISH
haw	HAWAIIAN	no	NORWEGIAN	yo	YORUBA
he	HEBREW	ny	NYANJA	zh	Chinese
hi	HINDI	or	ORIYA	zu	ZULU
hmn	HMONG	pa	PUNABI		
hr	CROATIAN	pl	POLISH		

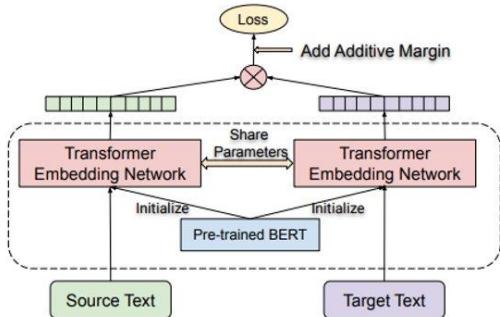


Figure 1: Dual encoder model with BERT based encoding modules.



# Translate between 200+ Languages With Marian: Fast Neural Machine Translation in C++

Afrikaans af	Arabic ar	Azeri az	Bulgarian bg	Bislama bi	Bengali bn	Breton br	Catalan ca	Czech cs	Welsh cy	Danish da	German de
Ewe ee	Greek el	English en	Esperanto eo	Spanish es	Estonian et	Basque eu	Farsi fa	Finnish fi	Fiji fj	French fr	Irish ga
Galician gl	Manx gv	Hausa ha	Hebrew he	Hindi hi	Hiri Motu ho	Haitian ht	Hungarian hu	Armenian hy	Indonesian id	Igbo ig	Icelandic is
Italian it	Japanese ja	Georgian ka	Kongo kg	Kuanyama kj	Greenlandic kl	Korean ko	Latin la	Ganda lg	Lingala ln	Luba-Katanga lu	Latvian lv
Malagasy mg	Marshallese mh	YEMRO makedoniaml	Malayalam ml	Marathi mr	Maltese mt	Ndonga ng	Dutch nl	Norwegian no	Chichewa ny	Oromo om	Punjabi pa
Polish pl	Portuguese pt	Kirundi rn	Romanian ro	Russian ru	Kinyarwanda rw	Sangro sg	Slovak sk	Slovenian sl	Samoa sm	Shona sn	Somali so
Albanian sq	Siswati ss	Sesotho st	Swedish sv	Thai th	Tigrinya ti	Tagalog tl	Tswana tn	Tongan to	Turkish tr	Tsonga ts	Twi tw
Tahitian ty	Ukrainian uk	Urdu ur	Venda ve	Vietnamese vi	Walloon wa	Xhosa xh	Yoruba yo	Chinese zh	Zulu zu		
... 94 more!											

# MARIAN NMT

## Fast Neural Machine Translation in C++



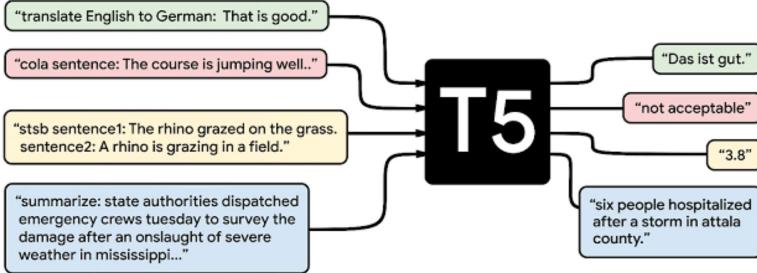
```
# Use ISO standards for the languages
nlu.load('<start_language>.translate_to.<target_language>')

#Translate Turkish to English:
nlu.load('tr.translate_to.en')

#Translate English to French:
nlu.load('en.translate_to.fr')

#Translate French to Hebrew
nlu.load('fr.translate_to.he')

#Translate English to German
nlu.load('en.translate_to.de')
```



```

# Closed book Question Answering
nlu.load('en.t5').predict('what is the capital of Germany?') # >>> Berlin
# Open Book Question answering
nlu.load('en.t5').predict('Who is president of Nigeria?') # >>> Muhammadu Buhari

# Open book Question Answering
context = 'Peters last week was terrible! He had an accident and broke his leg while skiing!'
question1 = 'Why was peters week so bad?'
question2 = 'How did peter broke his leg?'
nlu.load('answer_question').predict(question1 + context) # >>> broke his leg
nlu.load('answer_question').predict(question2 + context) # >>> skiing

# Big T5 model for Summarization, Sentiment, Text Similarity and other SQuAD/GLUE tasks
pipe = nlu.load('t5')
pipe['t5'].settask('summarize')
pipe.predict(long_text)

```

## Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

1. Text summarization
2. Question answering
3. Translation
4. Sentiment analysis
5. Natural Language inference
6. Coreference resolution
7. Sentence Completion
8. Word sense disambiguation



Every T5 Task with explanation:

Task Name	Explanation
1.CoLA	Classify if a sentence is grammatically correct
2.RTE	Classify whether a statement can be deduced from a sentence
3.MNLI	Classify for a hypothesis and premise whether they contradict or contradict each other or neither of both (3 class).
4.MRPC	Classify whether a pair of sentences is a re-phrasing of each other (semantically equivalent)
5.QNLI	Classify whether the answer to a question can be deducted from an answer candidate.
6.QQP	Classify whether a pair of questions is a re-phrasing of each other (semantically equivalent)
7.SST2	Classify the sentiment of a sentence as positive or negative
8.STSB	Classify the sentiment of a sentence on a scale from 1 to 5 (21 Sentiment classes)
9.CB	Classify for a premise and a hypothesis whether they contradict each other or not (binary).
10.COPA	Classify for a question, premise, and 2 choices which choice the correct choice is (binary).
11.MultiRc	Classify for a question, a paragraph of text, and an answer candidate, if the answer is correct (binary).
12.WIC	Classify for a pair of sentences and a disambiguous word if the word has the same meaning in both sentences.
13.WSC/DPR	Predict for an ambiguous pronoun in a sentence what it is referring to.
14.Summarization	Summarize text into a shorter representation.
15.SQuAD	Answer a question for a given context.
16.WMT1	Translate English to German
17.WMT2	Translate English to French
18.WMT3	Translate English to Romanian

# Train Transformer Models via Huggingface or TfHub and scale with Spark NLP



## TF Hub to Spark NLP

Spark NLP	TF Hub Notebooks	Colab
BertEmbeddings	<a href="#">TF Hub in Spark NLP - BERT</a>	Open in Colab
BertSentenceEmbeddings	<a href="#">TF Hub in Spark NLP - BERT Sentence</a>	Open in Colab
AlbertEmbeddings	<a href="#">TF Hub in Spark NLP - ALBERT</a>	Open in Colab

Spark NLP	HuggingFace Notebooks	Colab
BertEmbeddings	<a href="#">HuggingFace in Spark NLP - BERT</a>	Open in Colab
BertSentenceEmbeddings	<a href="#">HuggingFace in Spark NLP - BERT Sentence</a>	Open in Colab
DistilBertEmbeddings	<a href="#">HuggingFace in Spark NLP - DistilBERT</a>	Open in Colab
CamemBertEmbeddings	<a href="#">HuggingFace in Spark NLP - CamemBERT</a>	Open in Colab
RoBertaEmbeddings	<a href="#">HuggingFace in Spark NLP - RoBERTa</a>	Open in Colab
DeBertaEmbeddings	<a href="#">HuggingFace in Spark NLP - DeBERTa</a>	Open in Colab
XlmRoBertaEmbeddings	<a href="#">HuggingFace in Spark NLP - XLM-RoBERTa</a>	Open in Colab
AlbertEmbeddings	<a href="#">HuggingFace in Spark NLP - ALBERT</a>	Open in Colab
XlnetEmbeddings	<a href="#">HuggingFace in Spark NLP - XLNet</a>	Open in Colab
LongformerEmbeddings	<a href="#">HuggingFace in Spark NLP - Longformer</a>	Open in Colab
BertForTokenClassification	<a href="#">HuggingFace in Spark NLP - BertForTokenClassification</a>	Open in Colab
DistilBertForTokenClassification	<a href="#">HuggingFace in Spark NLP - DistilBertForTokenClassification</a>	Open in Colab
AlbertForTokenClassification	<a href="#">HuggingFace in Spark NLP - AlbertForTokenClassification</a>	Open in Colab
RoBertaForTokenClassification	<a href="#">HuggingFace in Spark NLP - RoBERTaForTokenClassification</a>	Open in Colab
XlmRoBertaForTokenClassification	<a href="#">HuggingFace in Spark NLP - XlmRoBertaForTokenClassification</a>	Open in Colab
BertForSequenceClassification	<a href="#">HuggingFace in Spark NLP - BertForSequenceClassification</a>	Open in Colab
DistilBertForSequenceClassification	<a href="#">HuggingFace in Spark NLP - DistilBertForSequenceClassification</a>	Open in Colab
AlbertForSequenceClassification	<a href="#">HuggingFace in Spark NLP - AlbertForSequenceClassification</a>	Open in Colab
RoBertaForSequenceClassification	<a href="#">HuggingFace in Spark NLP - RoBERTaForSequenceClassification</a>	Open in Colab
XlmRoBertaForSequenceClassification	<a href="#">HuggingFace in Spark NLP - XlmRoBertaForSequenceClassification</a>	Open in Colab
XlnetForSequenceClassification	<a href="#">HuggingFace in Spark NLP - XLNetForSequenceClassification</a>	Open in Colab
LongformerForSequenceClassification	<a href="#">HuggingFace in Spark NLP - LongformerForSequenceClassification</a>	Open in Colab
AlbertForQuestionAnswering	<a href="#">HuggingFace in Spark NLP - AlbertForQuestionAnswering</a>	Open in Colab
BertForQuestionAnswering	<a href="#">HuggingFace in Spark NLP - BertForQuestionAnswering</a>	Open in Colab
DeBertaForQuestionAnswering	<a href="#">HuggingFace in Spark NLP - DeBertaForQuestionAnswering</a>	Open in Colab
DistilBertForQuestionAnswering	<a href="#">HuggingFace in Spark NLP - DistilBertForQuestionAnswering</a>	Open in Colab
LongformerForQuestionAnswering	<a href="#">HuggingFace in Spark NLP - LongformerForQuestionAnswering</a>	Open in Colab
RoBertaForQuestionAnswering	<a href="#">HuggingFace in Spark NLP - RoBERTaForQuestionAnswering</a>	Open in Colab
XlmRobertaForQuestionAnswering	<a href="#">HuggingFace in Spark NLP - XlmRobertaForQuestionAnswering</a>	Open in Colab



# Session 3 (Day 1) - Coding Time

- ❖ [Notebook 3 Spark NLP pretrained models](#)

Spark NLP  
for Data Scientists



# Session 4 (Day 1)

- ❖ Training a Named Entity Recognizer
- ❖ Training a Deep Learning classifier
- ❖ Upload/Download your own model via John Snow Labs Models Hub

Spark NLP  
for Data Scientists



# Session 4 (Day 1)

- ❖ [Notebook 4 NERDL Training](#)
- ❖ [Notebook 5 ClassifierDL, SentimentDL, MultiClassifierDL Training](#)
- ❖ [Upload a Trained Model to JSL Models Hub](#)

## CoNLL 2003 (English)

The CoNLL 2003 NER task consists of newswire text from the Reuters RCV1 corpus tagged with four different entity types (PER, LOC, ORG, MISC). Models are evaluated based on span-based F1 on the test set. \* used both the train and development splits for training.

Model	F1	Paper / Source	Code
CNN Large + fine-tune (Baevski et al., 2019)	93.5	Cloze-driven Pretraining of Self-attention Networks	
RNN-CRF+Flair	93.47	Improved Differentiable Architecture Search for Language Modeling and Named Entity Recognition	
LSTM-CRF+ELMo+BERT+Flair	93.38	Neural Architectures for Nested NER through Linearization	Official
Flair embeddings (Akbik et al., 2018)*	93.09	Contextual String Embeddings for Sequence Labeling	Flair framework
BERT Large (Devlin et al., 2018)	92.8	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	
CVT + Multi-Task (Clark et al., 2018)	92.61	Semi-Supervised Sequence Modeling with Cross-View Training	Official
BERT Base (Devlin et al., 2018)	92.4	BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding	
BILSTM-CRF+ELMo (Peters et al., 2018)	92.22	Deep contextualized word representations	AllenNLP Project AllenNLP GitHub
Peters et al. (2017) *	91.93	Semi-supervised sequence tagging with bidirectional language models	
CRF + AutoEncoder (Wu et al., 2018)	91.87	Evaluating the Utility of Hand-crafted Features in Sequence Labelling	Official
Bi-LSTM-CRF + Lexical Features (Ghadhar and Langlais 2018)	91.73	Robust Lexical Features for Improved Neural Network Named-Entity Recognition	Official
BILSTM-CRF + IntNet (Xin et al., 2018)	91.64	Learning Better Internal Structure of Words for Sequence Labeling	
Chiu and Nichols (2016) *	91.62	Named entity recognition with bidirectional LSTM-CNNs	

# NER-DL in Spark NLP

SYSTEM	YEAR	LANGUAGE	ACCURACY
Spark NLP v2.4	2020	Python/Scala/Java/R	93.3 (test F1) - 95.9 (dev F1)
Spark NLP v2.x	2019	Python/Scala/Java/R	93
Spark NLP v1.x	2018	Python/Scala/Java/R	92
spaCy v2.x	2017	Python/Cython	92.6
spaCy v1.x	2015	Python/Cython	91.8
ClearNLP	2015	Java	91.7
CoreNLP	2015	Java	89.6
MATE	2015	Java	92.5
Turbo	2015	C++	92.4

The best NER score in production

93.3 %  
Test Set

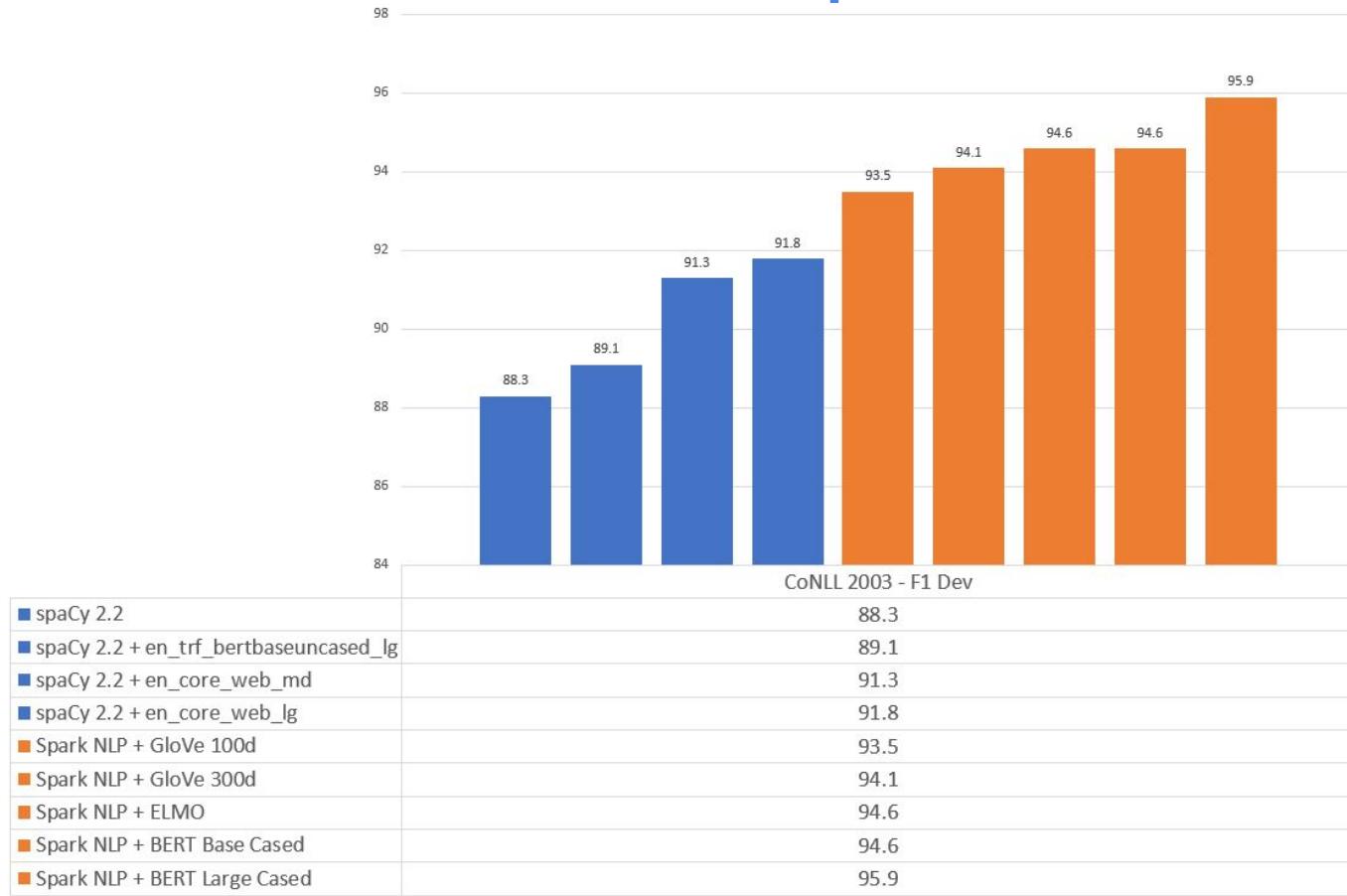


Bert



NerDLApproach

# NER-DL in Spark NLP



# NER Systems

Feature-engineered machine learning systems	Dict	SP	DU	EN	GE
Carreras et al. (2002) binary AdaBoost classifiers	Yes	81.39	77.05	-	-
Malouf (2002) - Maximum Entropy (ME) + features	Yes	73.66	68.08	-	-
Li et al. (2005) SVM with class weights	Yes	-	-	88.3	-
Passos et al. (2014) CRF	Yes	-	-	90.90	-
Ando and Zhang (2005a) Semi-supervised state of the art	No	-	-	89.31	75.27
Agerri and Rigau (2016)	Yes	<b>84.16</b>	<b>85.04</b>	<b>91.36</b>	<b>76.42</b>
Feature-inferring neural network word models					
Collobert et al. (2011) Vanilla NN +SLL / Conv-CRF	No	-	-	81.47	-
Huang et al. (2015) Bi-LSTM+CRF	No	-	-	84.26	-
Yan et al. (2016) Win-BiLSTM (English), FF (German) (Many fets)	Yes	-	-	88.91	<b>76.12</b>
Collobert et al. (2011) Conv-CRF (SENNNA+Gazetteer)	Yes	-	-	89.59	-
Huang et al. (2015) Bi-LSTM+CRF+ (SENNNA+Gazetteer)	Yes	-	-	<b>90.10</b>	-
Feature-inferring neural network character models					
Gillick et al. (2015) – BTS	No	<b>82.95</b>	<b>82.84</b>	<b>86.50</b>	<b>76.22</b>
Kuru et al. (2016) CharNER	No	82.18	79.36	84.52	70.12
Feature-inferring neural network word + character models					
Yang et al. (2017)	Yes	85.77	<b>85.19</b>	91.26	-
Luo (2015)	Yes	-	-	91.20	-
Chiu and Nichols (2015)	Yes	-	-	<b>91.62</b>	-
Ma and Hovy (2016)	No	-	-	91.21	-
Santos and Guimaraes (2015)	No	82.21	-	-	-
Lample et al. (2016)	No	85.75	81.74	90.94	<b>78.76</b>
Bharadwaj et al. (2016)	Yes	<b>85.81</b>	-	-	-
Dernoncourt et al. (2017)	No	-	-	90.5	-
Feature-inferring neural network word + character + affix models					
Re-implementation of Lample et al. (2016) (100 Epochs)	No	85.34	85.27	90.24	78.44
Yadav et al. (2018)(100 Epochs)	No	86.92	87.50	90.69	78.56
Yadav et al. (2018) (150 Epochs)	No	<b>87.26</b>	<b>87.54</b>	90.86	<b>79.01</b>

## 1. Classical Approaches (rule based)

## 2. ML Approaches

- Multi-class classification
- Conditional Random Field (CRF)

## 3. DL Approaches

- Bidirectional LSTM-CRF
- Bidirectional LSTM-CNNs
- Bidirectional LSTM-CNNS-CRF
- Pre-trained language models  
(Bert, Elmo)

## 4. Hybrid Approaches (DL + ML)

# NER-DL in Spark NLP

## Char-CNN-BiLSTM

	F1 : Tokens	F2 : Casing	F3 : POS	F4 : Char CNN	Labels
The					O
company					O
XYZ					Company
Private					Company
Limited					Company
works					O
in					O
the					O
health					Activity
sector					Activity
in					O
Europe					Location

# NER-DL in Spark NLP

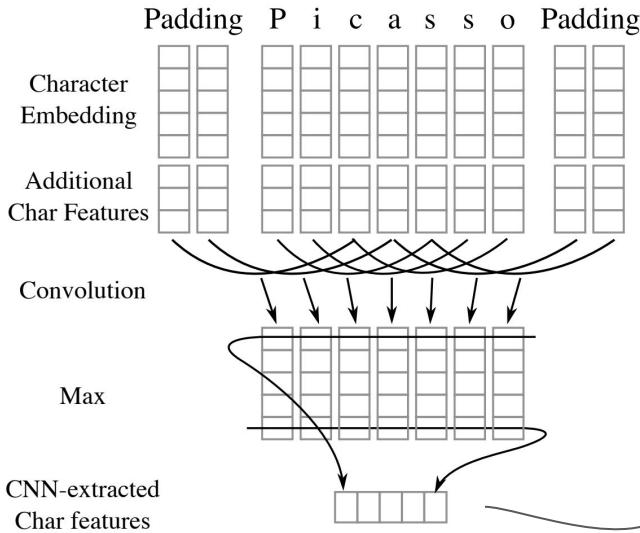


Figure 2: The convolutional neural network extracts character features from each word. The character embedding and (optionally) the character type feature vector are computed through lookup tables. Then, they are concatenated and passed into the CNN.

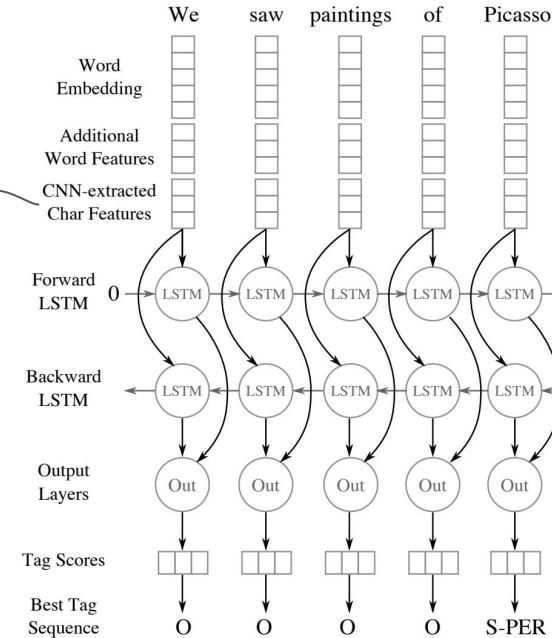


Figure 1: The (unrolled) BLSTM for tagging named entities. Multiple tables look up word-level feature vectors. The CNN (Figure 2) extracts a fixed length feature vector from character-level features. For each word, these vectors are concatenated and fed to the BLSTM network and then to the output layers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

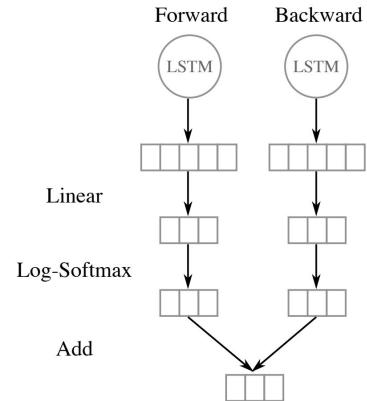


Figure 3: The output layers (“Out” in Figure 1) decode output into a score for each tag category.

## Char-CNN-BiLSTM

# NER-DL in Spark NLP

## CoNLL2003 format

All data files contain one word per line with empty lines representing sentence boundaries. At the end of each line there is a tag which states whether the current word is inside a named entity or not. The tag also encodes the type of named entity. Here is an example sentence:

U.N.	NNP	I-NP	I-ORG
official	NN	I-NP	O
Ekeus	NNP	I-NP	I-PER
heads	VBZ	I-VP	O
for	IN	I-PP	O
Baghdad	NNP	I-NP	I-LOC
.	.	O	O

\* Each line contains four fields: the word, its part-of-speech tag, its chunk tag and its named entity tag.

\* CoNLL: Conference on Computational Natural Language Learning

## BIO schema

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

John Smith ⇒ PERSON  
New York ⇒ LOCATION

# Coding ...

Open 4. NERDL Training notebook in Colab

(click on Colab icon or open in a new tab)

[https://github.com/JohnSnowLabs/spark-nlp-workshop/blob/master/tutorials/Certification\\_Trainings/Public/4.NERDL\\_Training.ipynb](https://github.com/JohnSnowLabs/spark-nlp-workshop/blob/master/tutorials/Certification_Trainings/Public/4.NERDL_Training.ipynb)

## Test set evaluation

```
In [ ]: import pyspark.sql.functions as F

predictions.select(F.explode(F.arrays_zip('token.result','label.result','ner.result')).alias("cols")) \
    .select(F.expr("cols['0']").alias("token"),
           F.expr("cols['1']").alias("ground_truth"),
           F.expr("cols['2']").alias("prediction")).show(truncate=False)

+-----+-----+-----+
|token |ground_truth|prediction|
+-----+-----+-----+
|CRICKET|o            |o          |
|-      |o            |o          |
|LEICESTERSHIRE|B-ORG|B-ORG
|TAKE   |o            |o          |
|OVER   |o            |o          |
|AT     |o            |o          |
|TOP    |o            |o          |
|AFTER  |o            |o          |
|INNINGS|o            |o          |
|VICTORY|o            |o          |
|.     |o            |o          |
|LONDON |B-LOC        |B-LOC
|1996-08-30|o            |o          |
|West   |B-MISC        |B-MISC
|Indian |I-MISC        |I-MISC
|all-roundner|o            |o          |
|Phil   |B-PER         |B-PER
|Simmons|I-PER         |I-PER
|took   |o            |o          |
|four   |o            |o          |
+-----+-----+-----+
only showing top 20 rows
```

```
In [ ]: from sklearn.metrics import classification_report

preds_df = predictions.select(F.explode(F.arrays_zip('token.result','label.result','ner.result')).alias("cols")) \
    .select(F.expr("cols['0']").alias("token"),
           F.expr("cols['1']").alias("ground_truth"),
           F.expr("cols['2']").alias("prediction")).toPandas()

print (classification_report(preds_df['ground_truth'], preds_df['prediction']))

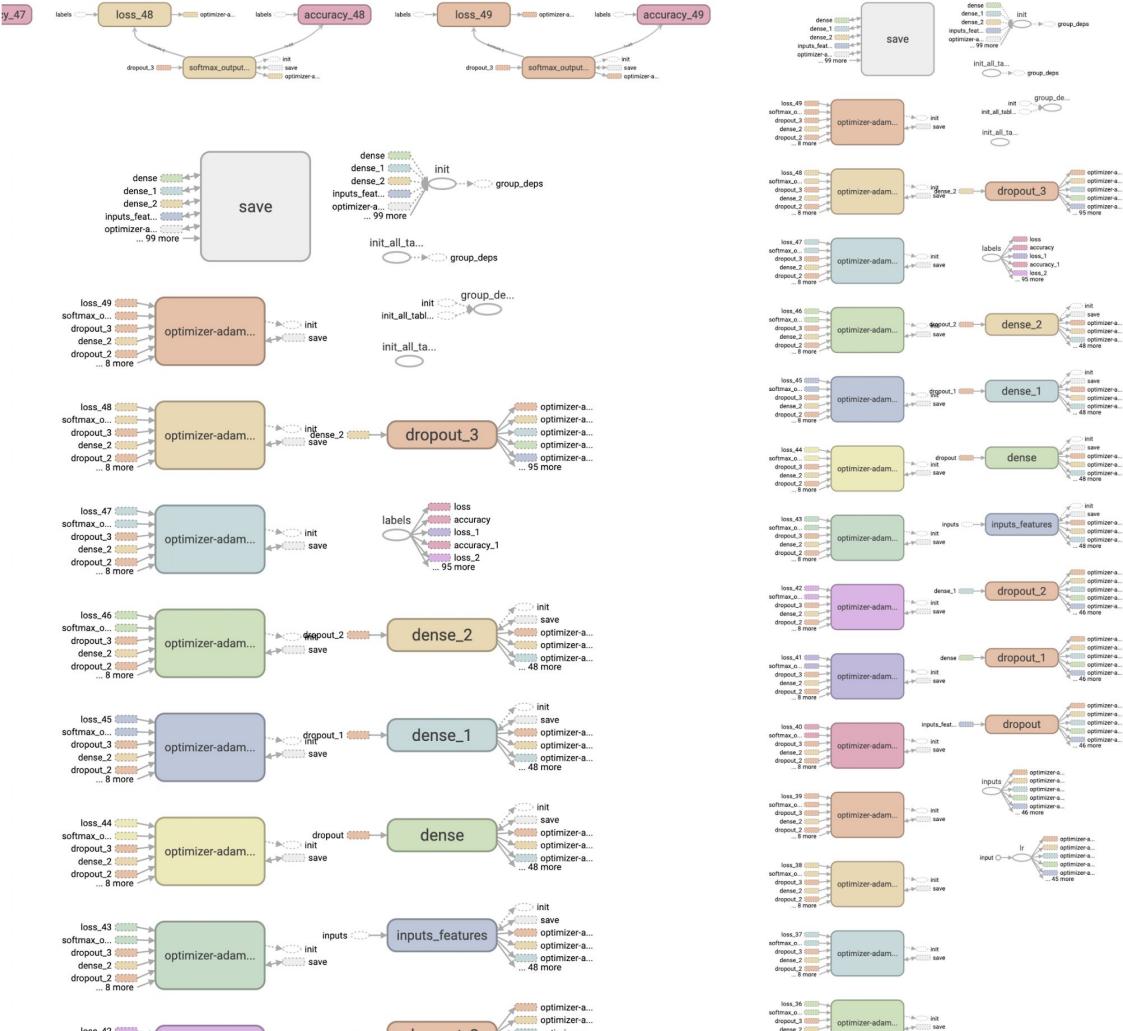
precision    recall   f1-score   support
B-LOC       0.88      0.93      0.90      1837
B-MISC      0.80      0.82      0.81       922
B-ORG       0.92      0.73      0.81      1341
B-PER       0.94      0.95      0.95      1842
```

# SentimentDL, ClassifierDL, and MultiClassifierDL

- BERT
  - Small BERT
  - BioBERT
  - CovidBERT
  - LaBSE
  - ALBERT
  - ELECTRA
  - XLNet
  - ELMO
  - Universal Sentence Encoder
  - GloVe
- 2 classes (positive/negative)
  - 3 classes (0, 1, 2)
  - 4 classes (Sports, Business, etc.)
  - 5 classes (1.0, 2.0, 3.0, 4.0, 5.0)
  - ... 100 classes!

- 100 dimensions
  - 200 dimensions
  - 128 dimensions
  - 256 dimensions
  - 300 dimensions
  - 512 dimensions
  - 768 dimensions
  - 1024 dimensions
- tfhub\_ues
  - tfhub\_use\_lg
  - glove\_6B\_100
  - glove\_6B\_300
  - glove\_840B\_300
  - bert\_base\_cased
  - bert\_base\_uncased
  - bert\_large\_cased
  - bert\_large\_uncased
  - bert\_multi\_uncased
  - electra\_small\_uncased
  - elmo
  - ... 100+ Word & Sentence models

# Classifier DL Tensorflow Architecture



# Upload trained models to Models Hub

- Trained models can be uploaded and shared via the modelshub
- Zip and Download the model
- Go to <https://modelshub.johnsnowlabs.com/> and upload zip file

The screenshot shows the 'Step 1' page of the John Snow LABS Modelshub. The top navigation bar includes links for Home, Docs, Learn, Models, Demo, GitHub, and Settings. A progress bar at the top indicates 'Step 1' is active. Below the progress bar, there's a section for uploading a model from a local computer or via a link. A 'Browse...' button shows the file path 'beto\_sentiment\_analysis.zip'. The main form area contains several input fields:

- Edition: Official
- Input Labels: [document, token]
- Output Labels: [class]
- Case sensitive: true
- Max sentence length: 128
- License: Open Source (radio button selected)
- Tags: An empty input field with a plus sign icon.
- Name: beto\_sentiment\_analysis
- Language: A dropdown menu showing '..'

# Session 4 (Day 1) - Coding Time

- ❖ [Notebook 4 NERDL Training](#)
- ❖ [Notebook 5 ClassifierDL, SentimentDL, MultiClassifierDL Training](#)
- ❖ [Upload your own model via John Snow Labs Models Hub](#)

End of Day 1 - see you tomorrow!  
Same time, same place :)

Spark NLP  
for Data Scientists



## Welcome to Day-2 - We have a lot of things ahead of us

Day-2	50 min	<ul style="list-style-type: none"><li>- Token Classification with Transformers</li><li>- Sequence Classification with Transformers</li><li>- Image Classification with ViT</li><li>- Speech to Text with Wav2Vec2</li></ul>
	10 min	Break
	50 min	<ul style="list-style-type: none"><li>- Spell Checking</li><li>- Unsupervised Keyword Extraction with YAKE</li><li>- Rule-based Entity Recognition with EntityRuler</li><li>- Graph triplet extraction</li></ul>
	10 min	Break
	50 min	<ul style="list-style-type: none"><li>- Table Question Answering with TAPAS</li><li>- Question Answering, Summarization and other T5 applications</li><li>- Multilingual NLP - Train only on English data and predict for 100+ languages</li></ul>
	10 min	Break
	50 min	<ul style="list-style-type: none"><li>- The Python NLU library basics, use any of the 7500+ models in 1 line of code</li><li>- NLP Server</li></ul>

# Session 1 (Day 2)

- ❖ Sequence & Token Classification with Transformers
- ❖ Image Classification with ViT
- ❖ Speech to Text with Wav2Vec2
- ❖ [Notebook 5.3 Transformers for Sequence Classification](#)
- ❖ [Notebook 14 Transformers for Token Classification](#)
- ❖ [Notebook 18 Vision Transformers for Image Classification](#)
- ❖ [Notebook 19 Speech Recognition with Wav2Vec](#)

Spark NLP  
for Data Scientists



# Training & Importing Transformers

JohnSnowLabs / spark-nlp Public

<> Code Issues 31 Pull requests 27 Discussions Actions ...

## Import Transformers into Spark NLP 🚀 #5669



maziyarpanahi on 7 Jun 2021

Maintainer



edited ▾

...

### Import Transformers into Spark NLP

#### Overview

We have extended support for HuggingFace 😊 and TF Hub exported models since 3.1.0 to equivalent Spark NLP 🚀 annotators. Starting this release, you can easily use the saved\_model feature in HuggingFace within a few lines of codes and import any BERT , DistilBERT , CamemBERT , RoBERTa , DeBERTa , XLM-RoBERTa , Longformer , BertForTokenClassification , DistilBertForTokenClassification , AlbertForTokenClassification , RoBertaForTokenClassification , DeBertaForTokenClassification , XlmRoBertaForTokenClassification , XlnetForTokenClassification , LongformerForTokenClassification , BertForSequenceClassification , DistilBertForSequenceClassification , AlbertForSequenceClassification , RoBertaForSequenceClassification , DeBertaForSequenceClassification , XlmRoBertaForSequenceClassification , XlnetForSequenceClassification , LongformerForSequenceClassification , AlbertForQuestionAnswering , BertForQuestionAnswering , DeBertaForQuestionAnswering , DistilBertForQuestionAnswering , LongformerForQuestionAnswering , RoBertaForQuestionAnswering , and XlmRoBertaForQuestionAnswering models to Spark NLP. We will work on the remaining annotators and extend this support to the rest with each release 😊

# Session 1 (Day 2) - Coding Time

- ❖ [Notebook 5.3 Transformers for Sequence Classification](#)
- ❖ [Notebook 14 Transformers for Token Classification](#)

# Image Classification with ViT

[Notebook 18 VIT for Image Classification](#)

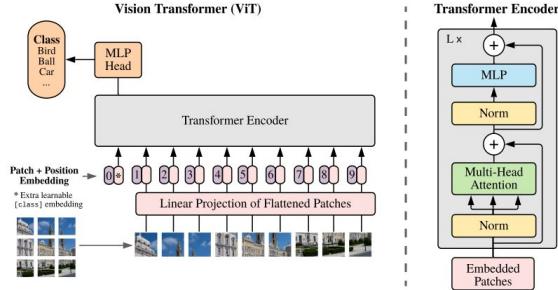
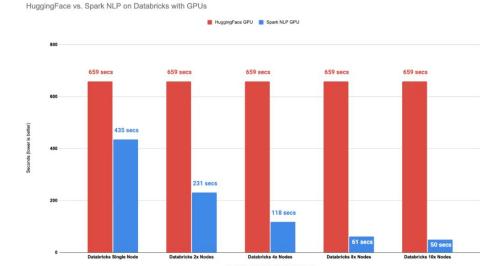


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable “classification token” to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

The screenshot shows the Hugging Face interface with a comparison between Hugging Face and Spark NLP. It highlights that Spark NLP is up to 2300x faster than Hugging Face on Databricks with GPUs. A bar chart compares the processing time for scaling from 1 node to 10 nodes:

Scaling Databricks from 1 node to 10 nodes	HuggingFace GPU	Spark NLP GPU
Databricks Single Node	659 secs	429 secs
Databricks 2x Nodes	659 secs	231 secs
Databricks 4x Nodes	659 secs	118 secs
Databricks 8x Nodes	659 secs	61 secs
Databricks 16x Nodes	659 secs	30 secs



Spark NLP is 1200% faster than Hugging Face with 10x Nodes

## Scale Vision Transformers (ViT) Beyond Hugging Face

### Spark NLP vs. HuggingFace



[VIT: An Image is Worth 16x16 Words:  
Transformers for Image Recognition at  
Scale](#)



Spark NLP 1.3x times  
faster on CPU



Spark NLP 1.5x times  
faster on GPU

Spark NLP is 15% faster than HuggingFace on a single-node Databricks by using only CPUs (with oneDNN enabled)  
Spark NLP is 49% faster than HuggingFace on a single-node Databricks by using only GPUs

[Over 200 VIT  
Models in various  
languages and  
sizes](#)

# Speech to Text Wav2vec 2.0

## Notebook 19 Speech2Text with Wav2Vec2

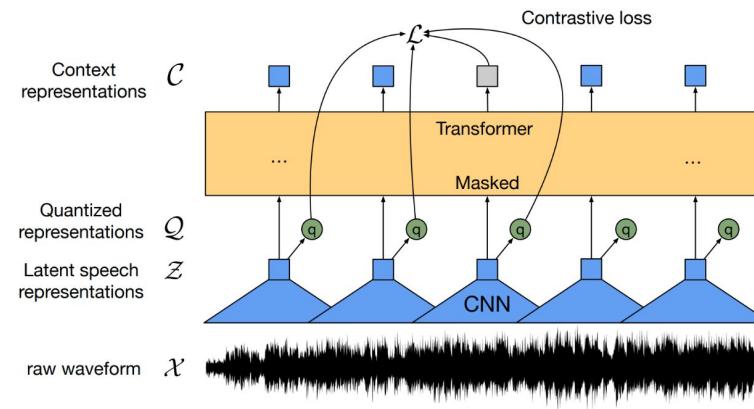
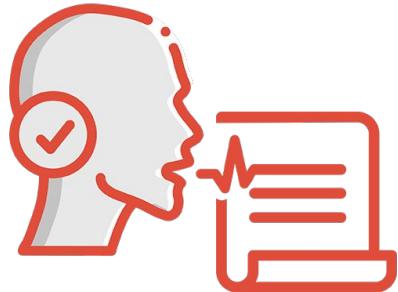


Figure 1: Illustration of our framework which jointly learns contextualized speech representations and an inventory of discretized speech units.

[wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations](#)

Meta

[Over 2000 Wav2Vec Models in various languages and sizes](#)

# Session 1 (Day 2) - Coding Time

- ❖ [Notebook 18 Vision Transformers for Image Classification](#)
- ❖ [Notebook 19 Speech Recognition with Wav2Vec](#)

Spark NLP  
for Data Scientists



# Session 2 (Day 2)

- ❖ [Notebook 7 Context Spell Checker](#)
- ❖ [Notebook 8 YAKE](#)
- ❖ [Notebook 20 EntityRuler](#)
- ❖ [Notebook 12 RDF Graph Extraction](#)

Spark NLP  
for Data Scientists



# Spell Checking & Correction



```
val pipeline = PretrainedPipeline("spell_check_ml", "en")
val result = pipeline.annotate("Harry Potter is a graet muvie")

println(result("spell"))
/* will print Seq[String](..., "is", "a", "great", "movie") */
```

- 3 trainable approaches
- **Norvig Approach:**
  - Retrieves tokens and auto-corrects based on a given dictionary
- **Symmetric Delete:**
  - Uses distance metrics to find possible words
- **Context Aware:**
  - Most accurate: Judges words in context
  - Deep learning based

# Context Spell Checker

The Spell Checker can leverage the context of words for ranking different correction sequences. Let's take a look at some examples,

```
# check for the different occurrences of the word "siter"
example1 = ["I will call my siter.", \
    "Due to bad weather, we had to move to a different siter.", \
    "We travelled to three siter in the summer."]
beautify(lp.annotate(example1))
```

```
['I will call my sister .\n',
 'Due to bad weather , we had to move to a different site .\n',
 'We travelled to three sites in the summer .\n']
```

```
# check for the different occurrences of the word "ueather"
example2 = ["During the summer we have the best ueather.", \
    "I have a black ueather jacket, so nice.", \
    "I introduce you to my sister, she is called ueather."]
beautify(lp.annotate(example2))
```

```
['During the summer we have the best weather .\n',
 'I have a black leather jacket , so nice .\n',
 'I introduce you to my sister , she is called Heather .\n']
```

Notice that in the first example, 'siter' is indeed a valid English word,

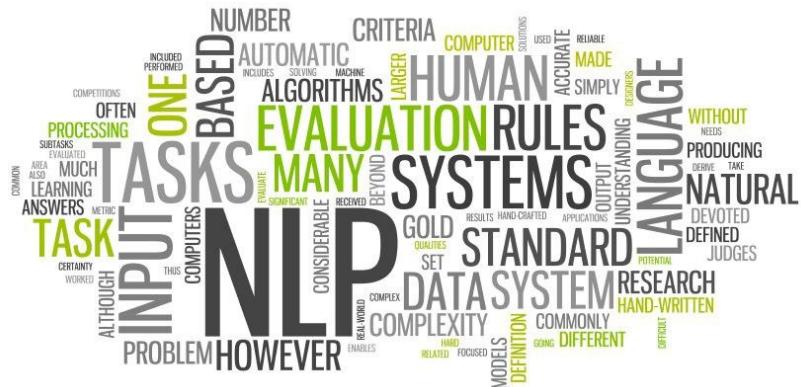
<https://www.merriam-webster.com/dictionary/siter>

[Notebook 7 Spell Checker](#)

# Unsupervised Keyword Extraction

YAKE! Is Yet Another Keyword Extraction Algorithm that can extract keywords without any weight by leveraging statistical properties of ngrams

# Notebook 8 YAKE



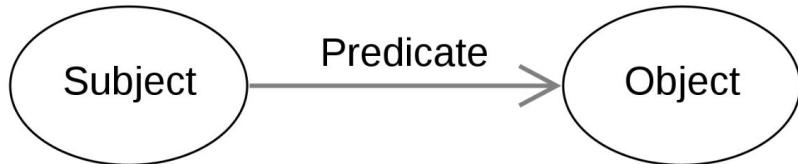
# Rule-based Entity Recognition with EntityRuler

## Notebook 20 EntityRuler

Fits an Annotator to match exact strings or regex patterns provided in a file against a Document and assigns them a named entity.

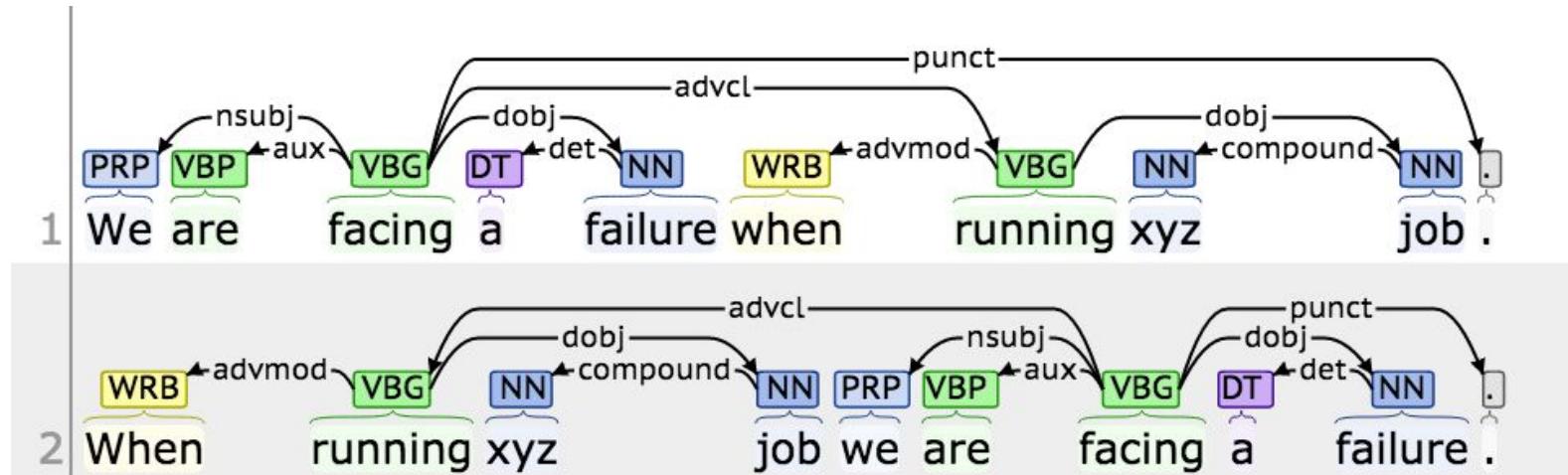
```
[  
  {  
    "id": "person-regex",  
    "label": "PERSON",  
    "patterns": ["\w+\s\w+", "\w+-\w+"]  
  },  
  {  
    "id": "locations-words",  
    "label": "LOCATION",  
    "patterns": ["Winterfell"]  
  }  
]  
  
[{"id": "names-with-j", "label": "PERSON", "patterns": ["Jon", "John", "John Snow"]}, {"id": "names-with-s", "label": "PERSON", "patterns": ["Stark", "Snow"]}, {"id": "names-with-e", "label": "PERSON", "patterns": ["Eddard", "Eddard Stark"]}]
```

# Extract RDF Semantic Triplets



Typed/Untyped Dependency Parsers define a **Predicate**  
Relationship between **Subject** and **Object**

## Notebook 12 Graph Extraction



# Session 2 (Day 2) - Coding Time

- ❖ [Notebook 7 Context Spell Checker](#)
- ❖ [Notebook 8 YAKE](#)
- ❖ [Notebook 20 EntityRuler](#)
- ❖ [Notebook 12 RDF Graph Extraction](#)

# Session 3 (Day 2)

- ❖ [Notebook 17 Table Question Answering with TAPAS](#)
- ❖ [Notebook 10 Question Answering and Summarization and more with T5](#)
- ❖ [Notebook 5.2 Training Multilingual Classifier](#)

# TAPAS for Table Classification

## Notebook 17 Table Question Answering with TAPAS

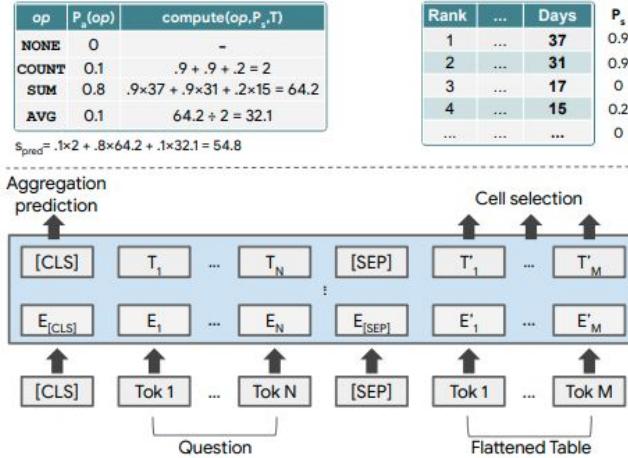


Figure 1: TAPAS model (bottom) with example model outputs for the question: “*Total number of days for the top two*”. Cell prediction (top right) is given for the selected column’s table cells in bold (zero for others) along with aggregation prediction (top left).

## TAPAS for Table Question Answering using Spark NLP 🚀

**TAPAS** is a Zero-shot Question Answering architecture, based on Bert, to carry out Table Understanding. By asking a question on natural language using these models, you can retrieve the content of the cell or cells which best answer to those questions.

Table

Rank	Name	No. of reigns	Combined days
1	Lou Thesz	3	3,749
2	Ric Flair	8	3,103
3	Harley Race	7	1,799
4	Dory Funk Jr.	1	1,563
5	Dan Severn	2	1,559
6	Gene Kiniski	1	1,131

Example questions

#	Question	Answer	Example Type
1	Which wrestler had the most number of reigns?	Ric Flair	Cell selection
2	Average time as champion for top 2 wrestlers?	AVG(3749,3103)=3426	Scalar answer
3	How many world champions are there with only one reign?	COUNT(Dory Funk Jr., Gene Kiniski)=2	Ambiguous answer
4	What is the number of reigns for Harley Race?	7	Ambiguous answer
5	Which of the following wrestlers were ranked in the bottom 3?	{Dory Funk Jr., Dan Severn, Gene Kiniski}	Cell selection
6	Out of these, who had more than one reign?	Dan Severn	Cell selection

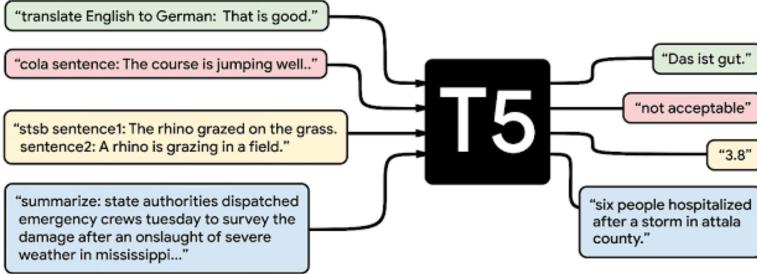
TAPAS models have been trained using a combination of three datasets:

- **SQA**, Sequential Question Answering by Microsoft (it was not trained to return aggregation operations as SUM, COUNT, etc - see below)
- **WTQ**, Wiki Table Questions by Stanford University (with aggregation operations)
- **Wiki SQL**, by Salesforce, also with aggregation operations.



[Table Classifiers Overview](#)

[TAPAS: Weakly Supervised Table Parsing via Pre-training](#)



## [Notebook 10 Question Answering and Summarization and more with T5](#)

```

# Closed book Question Answering
nlu.load('en.t5').predict('what is the capital of Germany?') # >>> Berlin
# Open Book Question answering
nlu.load('en.t5').predict('Who is president of Nigeria?') # >>> Muhammadu Buhari

# Open book Question Answering
context = 'Peters last week was terrible! He had an accident and broke his leg while skiing!'
question1 = 'Why was peters week so bad?'
question2 = 'How did peter broke his leg?'
nlu.load('answer_question').predict(question1 + context) # >>> broke his leg
nlu.load('answer_question').predict(question2 + context) # >>> skiing

# Big T5 model for Summarization, Sentiment, Text Similarity and other SQuAD/GLUE tasks
pipe = nlu.load('t5')
pipe['t5'].settask('summarize')
pipe.predict(long_text)

```

## Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

1. Text summarization
2. Question answering
3. Translation
4. Generate SQL from natural language text
5. Text style transfer
6. Sentiment analysis
7. Natural Language inference
8. Coreference resolution
9. Sentence Completion
10. Word sense disambiguation



Every T5 Task with explanation:

Task Name	Explanation
1.CoLA	Classify if a sentence is grammatically correct
2.RTE	Classify whether a statement can be deduced from a sentence
3.MNLI	Classify for a hypothesis and premise whether they contradict or contradict each other or neither of both (3 class).
4.MRPC	Classify whether a pair of sentences is a re-phrasing of each other (semantically equivalent)
5.QNLI	Classify whether the answer to a question can be deducted from an answer candidate.
6.QQP	Classify whether a pair of questions is a re-phrasing of each other (semantically equivalent)
7.SST2	Classify the sentiment of a sentence as positive or negative
8.STSB	Classify the sentiment of a sentence on a scale from 1 to 5 (21 Sentiment classes)
9.CB	Classify for a premise and a hypothesis whether they contradict each other or not (binary).
10.COPA	Classify for a question, premise, and 2 choices which choice the correct choice is (binary).
11.MultiRc	Classify for a question, a paragraph of text, and an answer candidate, if the answer is correct (binary).
12.WIC	Classify for a pair of sentences and a disambiguous word if the word has the same meaning in both sentences.
13.WSC/DPR	Predict for an ambiguous pronoun in a sentence what it is referring to.
14.Summarization	Summarize text into a shorter representation.
15.SQuAD	Answer a question for a given context.
16.WMT1	Translate English to German
17.WMT2	Translate English to French
18.WMT3	Translate English to Romanian

# 100+ Languages supported by Language-agnostic BERT Sentence Embedding (LABSE) and XLM-RoBERTa

Train in 1 Language, predict in 100+ different languages

## Notebook 5.2 Training Multilingual Classifier

```
# Binary Class Classifier, 2 classes
nlu.load('xx.embed_sentence.labse train.sentiment').fit(train_df).predict(test_df)

# Multi Class Classifier, N classes
nlu.load('xx.embed_sentence.labse train.classifier').fit(train_df).predict(test_df)

# Multi Class Classifier with multiple labels example (i.e. Hashtags)
# N classes, where one row can be assigned up to N labels
nlu.load('xx.embed_sentence.labse train.multi_classifier').fit(train_df).predict(test_df)
```

ISO	NAME	ISO	NAME	ISO	NAME
af	AFRIKAANS	ht	HAITIAN_CREOLE	pt	PORTRUGUESE
am	AMHARIC	hu	HUNGARIAN	ro	ROMANIAN
ar	ARABIC	hy	ARMENIAN	ru	RUSSIAN
as	ASSAMESE	id	INDONESIAN	rw	KINYARWANDA
az	AZERBAIJANI	ig	IGBO	si	SINHALESE
be	BELARUSIAN	is	ICELANDIC	sk	SLOVAK
bg	BULGARIAN	it	ITALIAN	sl	SLOVENIAN
bn	BENGALI	ja	JAPANESE	sm	SAMOAN
bo	TIBETAN	jav	JAVANESE	sn	SHONA
bs	BOSNIAN	ka	GEORGIAN	so	SOMALI
ca	CATALAN	kk	KAZAKH	sq	ALBANIAN
ceb	CEBUANO	km	KHMER	sr	SERBIAN
co	CORSICAN	kn	KANNADA	st	SESOTHO
cs	CZECH	ko	KOREAN	su	SUNDANESE
cy	WELSH	ku	KURDISH	sv	SWEDISH
da	DANISH	ky	KYRGYZ	sw	SWAHILI
de	GERMAN	la	LATIN	ta	TAMIL
el	GREEK	lb	LUXEMBOURGISH	te	TELUGU
en	ENGLISH	lo	LAOTHIAN	tg	TAJIK
eo	ESPERANTO	lt	LITHUANIAN	th	THAI
es	SPANISH	lv	LATVIAN	tk	TURKMEN
et	ESTONIAN	mg	MALAGASY	tl	TAGALOG
eu	BASQUE	mi	MAORI	tr	TURKISH
fa	PERSIAN	mk	MACEDONIAN	tt	TATAR
fi	FINNISH	ml	MALAYALAM	ug	UIGHUR
fr	FRENCH	mn	MONGOLIAN	uk	UKRAINIAN
fy	FRISIAN	mr	MARATHI	ur	URDU
ga	IRISH	ms	MALAY	uz	UZBEK
gd	SCOTS_GAELIC	mt	MALTESE	vi	VietNAMESE
gl	Galician	my	BURMESE	wo	WOLOF
gu	GUARATI	ne	NEPALI	xh	XHOSA
ha	HAUSA	nl	DUTCH	yi	YIDDISH
haw	WAHAWIAN	no	NORWEGIAN	yo	YORUBA
he	HEBREW	ny	NYANJA	zh	Chinese
hi	HINDI	or	ORIYA	zu	ZULU
hmn	HMONG	pa	PUNABI	pl	Polish
hr	CROATIAN	pl	POLISH		

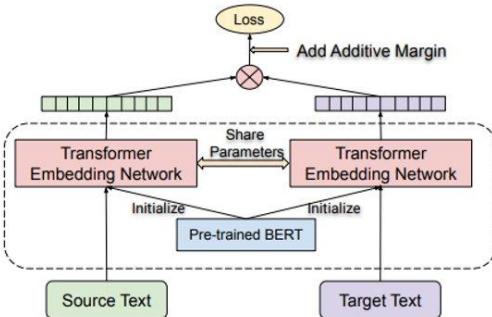


Figure 1: Dual encoder model with BERT based encoding modules.

# Spark NLP Resources

Spark NLP Official page

Spark NLP Workshop Repo

JSL Youtube channel

JSL Blogs

Introduction to Spark NLP: Foundations and Basic Components (Part-I)

Introduction to: Spark NLP: Installation and Getting Started (Part-II)

Named Entity Recognition with Bert in Spark NLP

Text Classification in Spark NLP with Bert and Universal Sentence Encoders

Spark NLP 101 : Document Assembler

Spark NLP 101: LightPipeline

<https://www.oreilly.com/radar/one-simple-chart-who-is-interested-in-spark-nlp/>

<https://blog.dominodatalab.com/comparing-the-functionality-of-open-source-natural-language-processing-libraries/>

<https://databricks.com/blog/2017/10/19/introducing-natural-language-processing-library-apache-spark.html>

<https://databricks.com/fr/session/apache-spark-nlp-extending-spark-ml-to-deliver-fast-scalable-unified-natural-language-processing>

<https://medium.com/@saif1988/spark-nlp-walkthrough-powered-by-tensorflow-9965538663fd>

<https://www.kdnuggets.com/2019/06/spark-nlp-getting-started-with-worlds-most-widely-used-nlp-library-enterprise.html>

<https://www.forbes.com/sites/forbestechcouncil/2019/09/17/why-spark-nlp-is-the-most-widely-used-nlp-library-enterprise/>

<https://medium.com/hackernoon/mueller-report-for-nerds-spark-meets-nlp-with-tensorflow-and-bert-part-1-32490a8f8f12>

<https://www.analyticsindiamag.com/5-reasons-why-spark-nlp-is-the-most-widely-used-library-enterprise/>

<https://www.oreilly.com/ideas/comparing-production-grade-nlp-libraries-training-spark-nlp-and-spacy-pipelines>

<https://www.oreilly.com/ideas/comparing-production-grade-nlp-libraries-accuracy-performance-and-scalability>

<https://www.infoworld.com/article/3031690/analytics/why-you-should-use-spark-for-machine-learning.html>

# Session 3 (Day 2) - Coding Time

- ❖ [Notebook 17 Table Question Answering with TAPAS](#)
- ❖ [Notebook 10 Question Answering and Summarization and more with T5](#)
- ❖ [Notebook 5.2 Training Multilingual Classifier](#)

# Session 4 (Day 2)

- ❖ Introduction to the Python NLU Library
- ❖ Introduction to NLP Server
- ❖ [Notebook 13 - NLU Crash course, every Spark NLP model in 1 line](#)

# What is NLU?

- All of the 10000+ Spark NLP models in 1 line of code
- Recognize text in Images, PDFs, DOCX files in 1 line of code via NLU powered by Spark OCR
- Train models in 1 line of code
- Visualize with Streamlit or in Jupyter Notebook
- Automagically generates Spark NLP pipelines based on your request (Dependency Resolution)
- Works on Pandas/Spark/Modin Dataframes and returns same type of Dataframe

## How does it work?



```
model= nlu.load(model)
```

- Returns a nlu pipeline object

```
model.predict(data)
```

- Returns a pandas DF

# How does it work?



```
model = nlu.load('emotion')
```

- Returns a nlu pipeline object

```
model.predict('I love NLU!')
```

- Returns a pandas DF

# EMOTION DETECTION

```
nlu.load('emotion').predict('I love NLU!')
```

sentence_embeddings	category_sentence	category_surprise	category_sadness	category_joy	category_fear	sentence	category	id
[0.027570432052016258, -0.052647676318883896, ...]	0	0.012899903	0.0015578865	0.9760173	0.0095249	I love NLU!	joy	1

## TOKENIZATION & SPELL CHECKING

```
nlu.load('spell').predict('I liek pentut butr and jelli')
```

token	checked	id
I	I	1
liek	like	1
peantut	peanut	1
buttr	butter	1
and	and	1
jelli	jelly	1

## NAMED ENTITY RECOGNITION

```
nlu.load('ner').predict('Angela Merkel from Germany and the American Donald Trump dont share many opinions')
```

embeddings	ner_tag	entities
[-0.563759982585907, 0.26958999037742615, 0.3...	PER	Angela Merkel
[-0.563759982585907, 0.26958999037742615, 0.3...	LOC	Germany
[-0.563759982585907, 0.26958999037742615, 0.3...	MISC	American
[-0.563759982585907, 0.26958999037742615, 0.3...	PER	Donald Trump

# CALCULATING EMBEDDINGS

#watch out for your RAM, this could kill your machine

```
nlu.load('bert elmo albert xlnet use glove').predict('Get all of them at once! Watch your RAM tough!')
```

token	glove_embeddings	albert_embeddings	xlnet_embeddings	bert_embeddings	elmo_embeddings	use_embeddings	id
Get	[0.1443299949169159, 0.4395099878311157, 0.583...]	[-0.41224443912506104, -0.4611411392688751, 0.70...]	[-0.003953204490244389, -1.5821468830108643, ...]	[-0.7420049905776978, -0.8647691011428833, 0.1...]	[0.04002974182367325, -0.43536433577537537, -0...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
all	[-0.2182299941778183, 0.691990178909302, 0.70...]	[1.1014549732208252, -0.43204769492149353, -0...]	[0.31148090958595276, -1.098618268966748, 0.3...]	[-0.8933112025260925, 0.44822725653648376, -0...]	[0.17885173857212067, 0.045830272138118744, -0...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
of	[-0.15289999544620514, -0.24278999865055084, 0...]	[1.1535910367965698, 0.28440719842910767, 0.60...]	[-1.403516411781311, 0.3108177185058594, -0.32...]	[-0.5550722479820251, 0.2702311873435974, 0.04...]	[0.24783466756343842, -0.248960942029953, 0.02...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
them	[-0.10130999982357025, 0.10941000282764435, 0...]	[0.5475010871887207, 0.8660883903503418, 2.817...]	[-0.7559828758239746, -0.4712887704372406, -1...]	[-0.2922026813030243, -0.1301671266555786, -0...]	[-0.24157099425792694, -0.8055092692375183, -0...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
at	[0.17659999430179596, 0.0938510000705719, 0.24...]	[-0.5005946159362793, -0.4600788354873657, 0.5...]	[0.04092511534690857, -1.0951932668685913, -1...]	[-0.5613634586334229, -0.00903533399105072, ...]	[-0.11999595910310745, 0.012994140386581421, ...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
once	[-0.2383799999523163, 0.22167001745224, 0.35...]	[-0.39100387692451477, -0.8297092914581299, 2...]	[-0.46001458168029785, -1.2062749862670898, 0...]	[0.29886400609961548, 0.3360409140586853, -0.37...]	[0.6701997518539429, 1.1368376016616821, 0.244...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
!	[0.38471999764442444, 0.49351000785827637, 0.4...]	[0.007945209741592407, -0.27733859419822693, 0...]	[-1.5816600322723389, -0.992130696773529, -0.1...]	[0.7550013065338135, -0.525778167724609, -0.4...]	[-1.335283073425293, 0.6296550035476685, -1.4...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
Watch	[-0.38264000415802, -0.08968199785924078, 0.02...]	[-0.10218311846256256, -0.433427620526886, 0...]	[-1.3921688795089722, 0.6997514963150024, -0.8...]	[-0.24852752685546875, 1.222611427307129, -0.1...]	[0.04002974182367325, -0.43536433577537537, -0...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
your	[-0.5718399882316589, 0.046348001807928085, 0...]	[-0.4086211323738098, 1.0755341053009033, 1.78...]	[-0.8588163256645203, -2.3702170848846436, 0.0...]	[-0.035358428955078125, 0.7711482048034668, 0...]	[0.17885173857212067, 0.045830272138118744, -0...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
RAM	[-1.87559980534309, -0.40814998745918724, 0...]	[-0.09772858023643494, 0.3632940351963043, -0...]	[1.1277621984481812, -1.689896583557129, -0.19...]	[0.4528151750564575, -0.36768051981925964, -0...]	[0.24783466756343842, -0.248960942029953, 0.02...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
tough	[-0.5099300146102905, -0.142800032901764, 0.5...]	[-0.22261293232440948, 0.21325691044330597, 0...]	[-1.3547197580337524, 0.43423181772232056, -1...]	[0.4607307978194275, 0.05694812536239624, 0.5...]	[-0.24157099425792694, -0.8055092692375183, -0...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
!	[0.38471999764442444, 0.49351000785827637, 0.4...]	[0.21658605337142944, -0.04937351495027542, 0...]	[-1.5816600322723389, -0.992130696773529, -0.1...]	[0.6830563545227051, -0.5751053094863892, -0.6...]	[-0.11999595910310745, 0.012994140386581421, ...]	[[-0.0019260947592556477, 0.009215019643306732...]	1

# NLU WORKS DIRECTLY ON TYPICAL PYTHON DATASETS

## Strings

```
import nlu  
nlu.load('sentiment').predict('This is just one string')
```

## Lists

```
import nlu  
nlu.load('sentiment').predict(['This is an array', ' Of strings!'])
```

## Pandas data frame

```
import nlu  
import pandas as pd  
data = {"text": ['This day sucks', 'I love this day', 'I dont like Sami']}  
text_df = pd.DataFrame(data)  
nlu.load('sentiment').predict(text_df)
```

## Pandas series

```
import nlu  
import pandas as pd  
data = {"text": ['This day sucks', 'I love this day', 'I dont like Sami']}
```

text\_df = pd.DataFrame(data)  
nlu.load('sentiment').predict(text\_df['text'])

**Spark**  
Data Frame

**Ray**  
Data Frame

**Dask**  
Data Frame

# NLU : Apache License 2.0

```
# Multiple binary sentiment classifiers trained on various datasets
nlu.load('classify.sentiment').predict('I love NLU and Python WebDev Conf 2021!')
nlu.load('classify.sentiment.imdb').predict('The Matrix was a pretty good movie')
nlu.load('classify.sentiment.twitter').predict('@elonmusk Tesla stock price is too high imo')

# Translate between 200 languages
nlu.load('en.translate_to.zh').predict('NLU can translate between 200 languages!')

# Spellchecking
nlu.load('spell').predict('I liek to live dangertus!')

# Extract Named Entities
nlu.load('ner').predict('Donald Trump and John Biden dont share many oppinions')

# Unsupervised Keyword Extraction
nlu.load('yake').predict('Weights extract keywords without requiring weights!')

# Over 50+ classifiers on various problems
nlu.load('classify.emotion').predict('He was suprised by the diversity of NLU')
nlu.load('classify.spam').predict('Hello you are the heir to a 100 Million fortune!')
nlu.load('classify.fakenews').predict('Unicorns landed on mars!')
nlu.load('classify.sarcasm').predict('love the teachers who give exams the day after halloween')
nlu.load('en.classify.question').predict('How expensive is the Watch?')
nlu.load('en.classify.toxic').predict('You are to stupid')
nlu.load('classify.cyberbullying').predict('Women belong in the kitchen!') #sorry

# Get BERTology and Transformer Embeddings for Sentences and Words
nlu.load('bert').predict('BERTology Word embeddings!')
nlu.load('bert elmo albert glove').predict('Multiple BERTology Word embeddings!')
nlu.load('embed_sentence.bert').predict('BERTology Sentence embeddings!')

# Text cleaning and Pre-Processing
nlu.load('lemmatize').predict('Get me the lemmatized version of a string')
nlu.load('normalize').predict('Get me the lemmatized version of a string')
nlu.load('clean').predict('Get me the lemmatized version of a string')

# Grammatical Parts of Speech
nlu.load('pos').predict('Extract Parts of Speech')
```

- Tokenization
- Sentence Detector
- Stop Words Removal
- Normalizer
- Stemmer
- Lemmatizer
- NGrams
- Regex Matching
- Text Matching
- Chunking
- Date Matcher
- Part-of-speech tagging
- Dependency parsing
- Sentiment Detection (ML models)
- Spell Checker (ML and DL models)
- Word Embeddings

- BERT Embeddings
- ELMO Embeddings
- ALBERT Embeddings
- XLNet Embeddings
- Universal Sentence Encoder
- BERT Sentence Embeddings
- Sentence Embeddings
- Chunk Embeddings
- Unsupervised keywords extraction
- Language Detection & Identification
- Multi-class Text Classification
- Multi-label Text Classification
- Multi-class Sentiment Analysis
- Named entity recognition
- Easy TensorFlow integration
- Full integration with Spark ML functions
- +250 pre-trained models in 46 languages!
- +90 pre-trained pipelines in 13 languages!

# Visualize NER results

```
| nlu.load('ner').viz("Donald Trump from America and Angela Merkel from Germany don't share many oppinions.")
```

```
onto_recognize_entities_sm download started this may take some time.
```

```
Approx size to download 160.1 MB
```

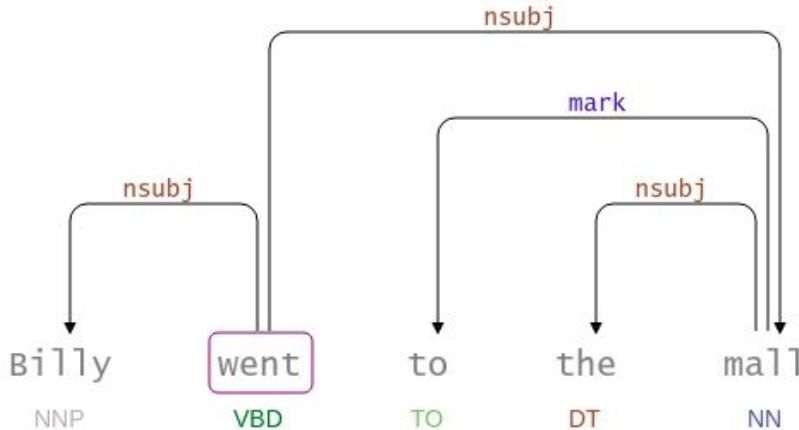
```
[OK!]
```

```
Donald Trump from America and Angela Merkel from Germany don't share many oppinions.  
PERSON GPE PERSON GPE
```

# Visualize Dependency Trees

```
nlu.load('dep.typed').viz("Billy went to the mall")
```

```
dependency_typed_conllu download started this may take some time.  
Approximate size to download 2.3 MB  
[OK!]  
dependency_conllu download started this may take some time.  
Approximate size to download 16.7 MB  
[OK!]  
pos_anc download started this may take some time.  
Approximate size to download 3.9 MB  
[OK!]  
sentence_detector_dl download started this may take some time.  
Approximate size to download 354.6 KB  
[OK!]
```



# Visualize Assertion results

```
nlu.load('med_ner.clinical assert').viz("The MRI scan showed no signs of cancer in the left lung")
```

```
ner_clinical download started this may take some time.
```

```
Approximate size to download 13.9 MB
```

```
[OK!]
```

```
assertion_dl download started this may take some time.
```

```
Approximate size to download 1.3 MB
```

```
[OK!]
```

```
embeddings_clinical download started this may take some time.
```

```
Approximate size to download 1.6 GB
```

```
[OK!]
```

```
sentence_detector_dl download started this may take some time.
```

```
Approximate size to download 354.6 KB
```

```
[OK!]
```

The MRI scan showed no signs of cancer in the left lung

TEST

PRESENT

PROBLEM

ABSENT

# Visualize Resolution

```
nlu.load('med_ner.jsl.wip.clinical_resolve_chunk.rxnorm.in').viz("He took 2 pills of Aspirin daily")
```

jsl\_ner\_wip\_clinical download started this may take some time.

Approximate size to download 14.5 MB

[OK!]

chunkresolve\_rxnorm\_in\_clinical download started this may take some time.

Approximate size to download 26.9 MB

[OK!]

embeddings\_clinical download started this may take some time.

Approximate size to download 1.6 GB

[OK!]

sentence\_detector\_dl download started this may take some time.

Approximate size to download 354.6 KB

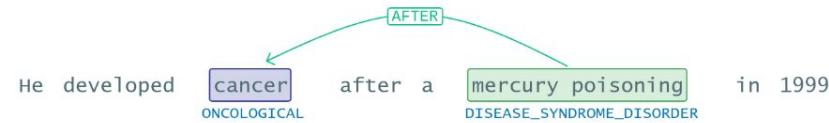
[OK!]

He	took	2	pills	of	Aspirin	daily
GENDER	DOSAGE	FORM	DRUG_INGREDIENT		FREQUENCY	
	2122105		1191			
	RNAPC2		ASPIRIN			

# Visualize Entity Relationships

```
nlu.load('med_ner.jsl.wip.clinical relation.temporal_events').viz('He developed cancer after a mercury poisoning in 1999 ')
```

```
jsl_ner_wip_clinical download started this may take some time.  
Approximate size to download 14.5 MB  
[OK!]  
redl_temporal_events_biobert download started this may take some time.  
Approximate size to download 383.3 MB  
[OK!]  
embeddings_clinical download started this may take some time.  
Approximate size to download 1.6 GB  
[OK!]  
sentence_detector_dl download started this may take some time.  
Approximate size to download 354.6 KB  
[OK!]
```



### nlu\_streamlit\_cheatsheet.py

```
# Input text data for visualizations
data = 'Billy from Berlin wants some pretty visualizations please!'

# Full UI with self generating Python code snippets for every feature
nlu.load('ner').viz_streamlit(data)

# Classification
nlu.load('sentiment').viz_streamlit_classes(data)

# Named Entity Recognition (NER)
nlu.load('ner').viz_streamlit_ner(data)

# Dependency Tree and Part of Speech (POS) Tags
nlu.load('dep.typed').viz_streamlit_dep_tree(data)

# Token features
nlu.load('stemm pos spell').viz_streamlit_token(data)

# Calculate similarity between two texts based on Word Embeddings
nlu.load('bert').viz_streamlit_word_similarity(['I love NLU!', 'I also Streamlit! <3'])

# Raw visualizations in Streamlit
nlu.load('<Model>').viz(data, write_to_streamlit=True)

# Predict on any datatype and returns Pandas df
nlu.load('<Model>').predict(data)

# Enable caching
nlu.enable_streamlit_caching()
```

### nlu\_viz\_cheatsheet.py

```
data = 'I want some pretty visualizations please!'

# Viz detected entities
nlu.load('ner').viz(data)

# Viz labeled dependency tree and POS tags
nlu.load('dep.typed').viz(data)

# Viz asserted statuses
nlu.load('med_ner.clinical_assert').viz(data)

# Viz resolved sentences
nlu.load('med_ner.jsl.wip.clinical_resolve.icd10cm').viz(data)

# Viz resolved entities
nlu.load('med_ner.jsl.wip.clinical_resolve_chunk.rxnorm.in').viz(data)

# Viz extracted relationships between entities
nlu.load('med_ner.jsl.wip.clinical_relation.temporal_events').viz(data)
```

# NLU Cheat Sheets

```
# Multiple binary sentiment classifiers trained on various datasets
nlu.load('classify.sentiment').predict('I love NLU and Python WebDev Conf 2021!')
nlu.load('classify.sentiment.imdb').predict('The Matrix was a pretty good movie')
nlu.load('classify.sentiment.twitter').predict('@elonmusk Tesla stock price is too high imo')

# Translate between 200 languages
nlu.load('en.translate_to.zh').predict('NLU can translate between 200 languages!')

# Spellchecking
nlu.load('spell').predict('I liek to live dangertus!')

# Extract Named Entities
nlu.load('ner').predict('Donald Trump and John Biden dont share many oppinions')

# Unsupervised Keyword Extraction
nlu.load('yake').predict('Weights extract keywords without requiring weights!')

# Over 50+ classifiers on various problems
nlu.load('classify.emotion').predict('He was suprised by the diversity of NLU')
nlu.load('classify.spam').predict('Hello you are the heir to a 100 Million fortune!')
nlu.load('classify.fakenews').predict('Unicorns landed on mars!')
nlu.load('classify.sarcasm').predict('love the teachers who give exams the day after halloween')
nlu.load('en.classify.question').predict('How expensive is the Watch?')
nlu.load('en.classify.toxic').predict('You are to stupid')
nlu.load('classify.cyberbullying').predict('Women belong in the kitchen!') #sorry

# Get BERTology and Transformer Embeddings for Sentences and Words
nlu.load('bert').predict('BERTology Word embeddings!')
nlu.load('bert elmo albert glove').predict('Multiple BERTology Word embeddings!')
nlu.load('embed_sentence.bert').predict('BERTology Sentence embeddings!')

# Text cleaning and Pre-Processing
nlu.load('lemmatize').predict('Get me the lemmatized version of a string')
nlu.load('normalize').predict('Get me the lemmatized version of a string')
nlu.load('clean').predict('Get me the lemmatized version of a string')

# Grammatical Parts of Speech
nlu.load('pos').predict('Extract Parts of Speech')
```

# Text Generating Sequence to Sequence Transformers in NLU & Spark NLP



Task Name	Explanation
1.CoLA	Classify if a sentence is grammatically correct
2.RTE	Classify whether a statement can be deduced from a sentence
3.MNLI	Classify for a hypothesis and premise whether they contradict or contradict each other or neither of both (3 classes).
4.MRPC	Classify whether a pair of sentences is a re-phrasing of each other (semantically equivalent)
5.QNLI	Classify whether the answer to a question can be deduced from an answer candidate.
6.QQP	Classify whether a pair of questions is a re-phrasing of each other (semantically equivalent)
7.SST2	Classify the sentiment of a sentence as positive or negative
8.STSB	Classify the sentiment of a sentence on a scale from 1 to 5 (21 Sentiment classes)
9.CB	Classify for a premise and a hypothesis whether they contradict each other or not (binary).
10.COPA	Classify for a question, premise, and 2 choices which choice the correct choice is (binary).
11.MultiRc	Classify for a question, a paragraph of text, and an answer candidate, if the answer is correct (binary).
12.WIC	Classify for a pair of sentences and a disambiguous word if the word has the same meaning in both sentences.
13.WSCIDPR	Predict for an ambiguous pronoun in a sentence what it is referring to.
14.Summarization	Summarize text into a shorter representation.
15.SQuAD	Answer a question for a given context.
16.WMT1	Translate English to German
17.WMT2	Translate English to French
18.WMT3	Translate English to Romanian

Summarize, answer questions, generate SQL and more with [Google's T5](#)  
Showcased in [this tutorial Webinar](#)



Generate long sequences of text, Augment Text and more with [Open AI's GPT2](#)  
Showcased today

Translate between over 200 languages with [Microsoft's MarianNMT](#)  
showcased on chinese in [this tutorial webinar](#)

# Sentence Similarity With BERTology Embeds or T5

Published in sparkNLP · Nov 20, 2020

## 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 using Bert, Electra, and Universal Sentence Encoder...

Data Science · 8 min read

Published in sparkNLP · Oct 24, 2020

## 1 Python Line for ELMO Word Embeddings and t-SNE plots with John Snow Labs' NLU

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

NLP · Train read

Published in sparkNLP · Oct 24, 2020

## Indepth and Easy Sentence Similarity Tutorial, with StackOverflow Questions using BERTology embeddings

Published in sparkNLP · Oct 24, 2020

## Explore 100+ Embeddings via 6 Similarity Metrics with 0 lines of code with NLU & Streamlit

Published in sparkNLP · Oct 24, 2020

## Explore 100+ Embeddings via 10+ Manifold and Matrix Decomposition with 0 lines of code with NLU & Streamlit

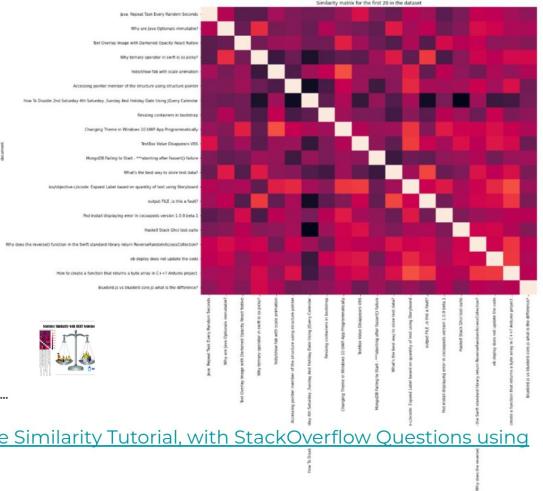
Published in sparkNLP · Oct 24, 2020

## Compare Multiple dimension reduction Techniques and Embeddings at the same time

Published in sparkNLP · Oct 24, 2020

## Supported similarities:

- Cosine
- Cityblock
- Euclidean
- L2
- L1
- Manhattan



# t-SNE Visualizations with NLU

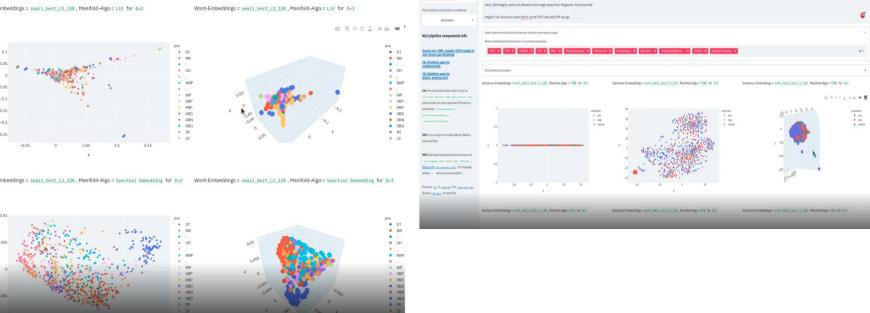


1 line of Python code for BERT, ALBERT, ELMO, ELECTRA, XLNET, GLOVE, Part of Speech with NLU and t-SNE

Explore 100+ Embeddings via 10+ Manifold and Matrix Decomposition with 0 lines of code with NLU & Streamlit



Compare multiple dimension reduction Techniques and Embeddings at the same time



## OCR Tables from PDFs, DOC, DOCX and PPT files in 1 Line of code with NLU 4.0



```
[1] nlu.load('pdf2table').predict('/content/tables.pdf')
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear
1	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4
2	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4
3	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4
4	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3
5	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3
6	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3
7	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3
8	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4
9	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4
10	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4
11	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4
12	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3
13	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3
14	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3
15	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3
16	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3
17	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3
18	32.4	4	78.7	66	4.08	2.200	18.47	1	1	4
19	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4

## Overview of all OCR features in NLU

### Overview of OCR Text Extractors

These models grab the text directly from your input file and returns it as a Pandas DataFrame

NLU Spell	Transformer Class
nlu.load('img2text')	<a href="#">ImageToText</a>
nlu.load('pdf2text')	<a href="#">PdfToText</a>
nlu.load('doc2text')	<a href="#">DocToText</a>

### Overview of OCR Table Extractors

These models grab all Table data from the files detected and return a [list of Pandas DataFrames](#), containing Pandas DataFrame for every table detected

NLU Spell	Transformer Class
nlu.load('pdf2table')	<a href="#">PdfToTextTable</a>
nlu.load('ppt2table')	<a href="#">PptToTextTable</a>
nlu.load('dec2table')	<a href="#">DocToTextTable</a>

More info on

[https://nlu.johnsnowlabs.com/docs/en/nlu\\_for\\_ocr](https://nlu.johnsnowlabs.com/docs/en/nlu_for_ocr)

### Image to Text

"The Old Pond" by Matsuo Bashō

An old silent pond

A frog jumps into the pond—

Splash! Silence again.

Sample image:

```
nlu.load('img2text').predict('path/to/haiku.png')
```

Output of IMG OCR:

text
"The Old Pond" by Matsuo Bashō
An old silent pond
A frog jumps into the pond—
Splash! Silence again.

### PDF to Text

haiku.pdf

"Lighting One Candle" by Yosa Buson

The light of a candle

Is transferred to another candle—

Spring twilight

Output of PDF OCR:

text
"Lighting One Candle" by Yosa Buson
The light of a candle
Is transferred to another candle—
Spring twilight

### DOCX to text

haiku.docx - LibreOffice Writer

Sample DOCX:

"In a Station of the Metro" by Ezra Pound

The apparition of these faces in the crowd;

Petals on a wet, black bough.

Output of DOCX OCR:

text
"In a Station of the Metro" by Ezra Pound
The apparition of these faces in the crowd;
Petals on a wet, black bough.

## NER Domains and Models Overview

# NER Domains and Models Overview

<a href="#">Radiology</a>	Find Radiology related entities	<a href="#">ned_ner.radiology.wip_clinical</a>	Bilateral, breast, ultrasound, ovoid mass, 0.5 x 0.5 x 0.4, cm; anteromedial aspect, left, shoulder, mass, isoechogenic echotexture, muscle, internal color flow, benign fibrous tissue, lipoma	ImagingTest, Imaging Technique, ImagingFindings, OtherFindings, BodyPart, Direction, Test, Symptom, Disease_Syndrome_Disorder, Medical_Device, Procedure, Measurements, Units	<a href="#">Dataset by John Snow Labs, MIMIC-CXR and MT Radiology texts</a>
<a href="#">Radiology Clinical JSL-V1</a>	Find radiology related entities in clinical setting	<a href="#">ned_ner.radiology.wip_greedy_biobert</a>	Bilateral, breast, ultrasound, ovoid mass, 0.5 x 0.5 x 0.4, cm, anteromedial aspect, left, shoulder, mass, isoechogenic echotexture, muscle, internal color flow, benign fibrous tissue, lipoma	Test Result, OtherFindings, BodyPart, ImagingFindings, Disease_Syndrome_Disorder, ImagingTest, Measurements, Procedure, Score, Test, Medical_Device, Direction, Symptom, Imaging_Technique, ManualFix, Units	<a href="#">Dataset by John Snow Labs</a>
<a href="#">Genes and Phenotypes</a>	Find Genes and Phenotypes (the observable physical properties of an organism) related entities	<a href="#">ned_ner.human_phenotype.gene_biobert</a>	AP0C4 . polyhydramnios	GENE, PHENOTYPE	<a href="#">PGR_1, PGR_2</a>
<a href="#">Normalized Genes and Phenotypes</a>	Find Normalized Genes and Phenotypes (the observable physical properties of an organism) related entities	<a href="#">ned_ner.human_phenotype.go_biobert</a>	protein complex oligomerization , defective platelet aggregation	G0, HP	<a href="#">PGR_1, PGR_2</a>
				Kidney Disease, HDL, Diet, Test, Imaging Technique, Triglycerides, Obesity, Duration, Weight, Sex, Height, Header, ImagingTest, Labour_Delivery, Disease_Syndrome_Disorder, Communicable Disease, Overweight, Units, Smoking, Score, Substance_Quantity, Form, Frequency, Modifier, Hyperlipidemia, ImagingFindings, Psychological_Condition, OtherFindings, Cerebrovascular Disease, Date, Test_Result, VS_Finding, Employment, Death_Entity, Gender, Oncological, Headache, Disease, Medical_Device, Total_Cholesterol, ManualFix, Time, Route, Pulse, Admission_Discharge, RelativeDate, O_Saturation, Frequency, Red_Actions, Hypertension, Alcohol, Allergen, Fever, Newborn, Birth_Entity, Age, Respiration, Medical_History_Header, Oxygen_Therapy, Section_Header, LDL, Treatment, Vital_Signs_Header, Direction, BMI, Primary, Secondary, Sexuality_Active or Sexual_Orientation, Symptom, Clinical_Dept, Measurements, Height, Family_History_Header, Substance, Strength, Injury or Poisoning, Relationship_Status, Blood_Pressure, Drug, Temperature, EKG_Findings, Diabetes, BodyPart, Vaccine, Procedure, Dosage	
<a href="#">Radiology Clinical JSL-V2</a>	Find radiology related entities in clinical setting	<a href="#">ned_ner.jsl.wip.clinical.rd</a>		Qualitative_Concept, Organization, Manufactured_Object, Amino_Acid, Peptide or Protein, Pharmacologic_Substance, Professional or Occupational_Group, Cell_Component, Neoplastic_Process, Substance, Laboratory_Procedure, Nucleic_Acid_Nucleoside or Nucleotide, Research_Group, Genetic_Element, Indicator_Reagent_or_Diagnostic_Aid, Biologic_Function, Chemical_Manufacture, Molecular_Function, Quantitative_Concept, Prokaryote, Mental or Behavioral_Dysfunction, Injury or Poisoning, Body_Location or Region, Spatial_Relation, Molecular_Sequence, Tissue, Pathologic_Function, Body_Substance, Fungus, Mental_Process, Medical_Device, Plant, Health_Care_Activity, Clinical_Attribute, Genetic_Function, Food, Therapeutic or Preventive_Procedure, Bio, Biological, Organism, Organ, Organ_Component, Geographic_Area, Virus, Biomedical or Dental_Material, Diagnostic_Procedure, Eukaryote, Anatomical_Structure, Organism_Attribute, Molecular_Biology_Research_Technique, Organic_Chemical_Cell, Daily or Recreational_Activity, Person, Group, Disease or Syndrome, Group_Sign or Symptom, Body_System	<a href="#">Dataset by John Snow Labs</a>
<a href="#">General Medical Terms</a>	Find general medical terms and medical entities.	<a href="#">ned_ner.medmentions</a>			<a href="#">MedMentions</a>

# Assertion Domains and Models Overview

Domain	Description	Spell	Predicted Entities	Examples	Reference Dataset
Radiology	Predict status of Radiology related entities	assert.radiology	Confirmed, Negative, Suspected	<ul style="list-style-type: none"> <li>- Confirmed : X-Ray scan shows cancer in lung.</li> <li>- Negative : X-Ray scan shows no sign of cancer in lung.</li> <li>- Suspected : X-Ray raises suspicion of cancer in lung but does not confirm it.</li> </ul>	<a href="#">Internal Dataset by Annotated by John Snow Labs</a>
Healthcare/Clinical extended and Family JSL powerd	Predict status of Healthcare/Clinical/Family related entities. Additional training with JSL Dataset	assert.jsl	Present, Absent, Possible, Planned, Someoneelse, Past, Family, Hypothetical	<ul style="list-style-type: none"> <li>- Present : Patient diagnosed with cancer in 1999</li> <li>- Absent : No sign of cancer was shown by the scans</li> <li>- Possible : Tests indicate patient might have cancer</li> <li>- Planned : CT-Scan is scheduled for 23.03.1999</li> <li>- Someoneelse : The patient gave Aspirin to daughter.</li> <li>- Past : The patient has no more headaches since the operation</li> <li>- Family : The patients father has cancer</li> <li>- Hypothetical : Death could be possible;</li> </ul>	<a href="#">2010 I2b2 + Data provided by JSL</a>
Healthcare/Clinical JSL powerd	Predict status of Healthcare/Clinical related entities. Additional training with JSL Dataset	assert.jsl_large	present, absent, possible, planned, someoneelse, past	<ul style="list-style-type: none"> <li>- present : Patient diagnosed with cancer in 1999</li> <li>- absent : No sign of cancer was shown by the scans</li> <li>- possible : Tests indicate patient might have cancer</li> <li>- planned : CT-Scan is scheduled for 23.03.1999</li> <li>- someoneelse : The patient gave Aspirin to daughter</li> <li>- past : The patient has no more headaches since the operation</li> </ul>	<a href="#">2010 I2b2 + Data provided by JSL</a>
Healthcare/Clinical classic	Predict status of Healthcare/Clinical related entities	assert.biobert	present , absent, possible, conditional, associated_with_someone_else ,hypothetical	<ul style="list-style-type: none"> <li>- present : Patient diagnosed with cancer in 1999</li> <li>- absent : No sign of cancer was shown by the scans</li> <li>- possible : Tests indicate patient might have cancer</li> <li>- conditional : If the test is positive, patient has AIDS</li> <li>- associated with someone else : The patients father has cancer</li> <li>- hypothetical : Death could be possible.</li> </ul>	<a href="#">2010 I2b2</a>

## Resolution Domains Models Overview

Domain/Terminology	Description	Sample NLU Spells	Sample Entities	Sample Predicted Codes	Reference Links
<a href="#">ICD-10 / ICD-10-CM (International Classification of Diseases - Clinical Modification)</a>	Get ICD-10-CM codes of Medical and Clinical Entities. The ICD-10 Clinical Modification (ICD-10-CM) is a modification of the ICD-10, authorized by the World Health Organization, used as a source for diagnosis codes in the U.S.. Be aware, ICD10-CM is often referred to as ICD10	resolve.icd10cm.augmented	hypertension , gastritis	I10, K2970	<a href="#">ICD-10-CM</a> <a href="#">WHO ICD-10-CM</a>
<a href="#">ICD-10-PCS (International Classification of Diseases - Procedure Coding System)</a>	Get ICD-10-PCS codes of Medical and Clinical Entities. The International Classification of Diseases, Procedure Coding System (ICD-10-PCS), is a U.S. cataloging system for procedural code. It is maintained by Centers for Medicare & Medicaid Services	resolve.icd10pcs	hypertension , gastritis	DWY18ZZ, 0472326	<a href="#">ICD10-PCS</a> <a href="#">CMS ICD-10-PCS</a>
<a href="#">ICD-O (International Classification of Diseases, Oncology) Topography &amp; Morphology codes</a>	Get ICD-8 codes of Medical and Clinical Entities. The International Classification of Diseases for Oncology (ICD-O), is a domain-specific extension of the International Statistical Classification of Diseases and Related Health Problems for tumor diseases.	resolve.icdo.base	metastatic lung cancer	9858/3 + C38.3, 8801/3 + C39.8	<a href="#">ICD-O Histology Behaviour dataset</a>
<a href="#">HCC (Hierarchical Conditional Categories)</a>	Get HCC codes of Medical and Clinical Entities. Hierarchical condition category (HCC) relies on ICD-10 coding to assign risk scores to patients. Along with demographic factors (such as age and gender), insurance companies use HCC coding to assign patients a risk adjustment factor (RAF) score.	resolve.hcc	hypertension , gastritis	139, 188	<a href="#">HCC</a>
<a href="#">ICD-10-CM + HCC Billable</a>	Get ICD-10-CM and HCC codes of Medical and Clinical Entities.	resolve.icd10cm.augmented_billable	metastatic lung cancer	C7880 + ['1', '1', '8']	<a href="#">ICD10-CM HCC</a>
<a href="#">CPT (Current Procedural Terminology)</a>	Get CPT codes of Medical and Clinical Entities. The Current Procedural Terminology(CPT) is developed by the American Medical Association (AMA) and used to assign codes to medical procedures/services/diagnoses. The codes are used to derive the amount of payment a healthcare provider may receives from insurance companies for the provided service/receives	resolve.cpt.procedures_measurements	calcium score, heart surgery	82310, 33257	<a href="#">CPT</a>
<a href="#">LOINC (Logical Observation Identifiers Names and Codes)</a>	Get LOINC codes of Medical and Clinical Entities. Logical Observation Identifiers Names and Codes (LOINC) developed by the U.S. organization Regenstrief Institute	resolve.loinc	acute hepatitis , obesity	28083-4, 50227-8	<a href="#">LOINC</a>
<a href="#">HPO (Human Phenotype Ontology)</a>	Get HPO codes of Medical and Clinical Entities.	resolve.hpo	cancer, bipolar disorder	0002664, 0007302, 0180753	<a href="#">HPO</a>
<a href="#">UMLS (Unified Medical Language System) CUI</a>	Get UMLS codes of Medical and Clinical Entities.	resolve.umls.findings	vomiting, polydipsia, hepatitis	C1963281, C3278316, C1963279	<a href="#">UMLS</a>
<a href="#">SNOMED International (Systematized Nomenclature of Medicine)</a>	Get SNOMED (INT) codes of Medical and Clinical Entities.	resolve.snomed.findings_int	hypertension	148439082	<a href="#">SNOMED</a>
<a href="#">SNOMED CT (Clinical Terms)</a>	Get SNOMED (CT) codes of Medical and Clinical Entities.	resolve.snomed.findings	hypertension	73578008	<a href="#">SNOMED</a>
<a href="#">SNOMED Conditions</a>	Get SNOMED Conditions codes of Medical and Clinical Entities.	resolve.snomed_conditions	schizophrenia	50214004	<a href="#">SNOMED</a>
<a href="#">RxNorm and RxCUI (Concept Unique Identifier)</a>	Get Normalized RxNorm and RxCUI codes of Medical, Clinical and Drug Entities.	resolve.rxnorm	50 mg of eltrombopag oral	825427	<a href="#">RxNorm Overview</a> <a href="#">November 2020 RxNorm Clinical Drugs ontology graph</a>

## Relation Extraction Domains and Models examples

Domain	Description	Sample NLU Spells	Predictable Relationships and Explanation
<a href="#">Dates and Clinical Entities</a>	Predict binary temporal relationship between Date Entities and Clinical Entities	relation.date	- 1 for Date Entity and Clinical Entity are related - 0 for Date Entity and Clinical Entity are not related
<a href="#">Body Parts and Directions</a>	Predict binary direction relationship between Bodypart Entities and Direction Entities	relation.bodypart.direction	- 1 for Body Part and Direction are related - 0 for Body Part and Direction are not related
<a href="#">Body Parts and Problems</a>	Predict binary location relationship between Bodypart Entities and Problem Entities	relation.bodypart.problem	- 1 for Body Part and Problem are related - 0 for Body Part and Problem are not related
<a href="#">Body Parts and Procedures</a>	Predict binary application relationship between Bodypart Entities and Procedure Entities	relation.bodypart.procedure	- 1 for Body Part and Test/Procedure are related - 0 for Body Part and Test/Procedure are not related
<a href="#">Adverse Effects between drugs (ADE)</a>	Predict binary effect relationship between Drugs Entities and Adverse Effects/Problem Entities	relation.ade	- 1 for Adverse Event Entity and Drug are related - 0 for Adverse Event Entity and Drug are not related
<a href="#">Phenotype abnormalities,Genes and Diseases</a>	Predict binary caused by relationship between Phenotype Abnormality Entities, Gene Entities and Disease Entities	relation.human_phenotype_gene	- 1 for Gene Entity and Phenotype Entity are related - 0 for Gene Entity and Phenotype Entity are not related
<a href="#">Temporal events</a>	Predict multi-class temporal relationship between Time Entities and Event Entities	relation.temporal_events	- AFTER if Any Entity occurred after Another Entity - BEFORE if Any Entity occurred before Another Entity - OVERLAP if Any Entity during Another Entity
<a href="#">Dates and Tests/Results</a>	Predict multi-class temporal cause/reasoning and conclusion relationship between Date Entities, Test Entities and Result Entities	relation.test_result_date	- relation.test_result_date - is finding of for Medical Entity is found because of Test Entity - is result of for Medical Entity reason for doing Test Entity - is date of for Date Entity relates to time of Test/Result - 0 : No relationship
<a href="#">Clinical Problem, Treatment and Tests</a>	Predict multi-class cause/reasoning and effect relationship between Treatment Entities, Problem Entities and Test Entities	relation.clinical	- TrIP: A certain treatment has improved/cured a medical problem - TrHP: A patient's medical problem has deteriorated or worsened because of treatment - TrCP: A treatment caused a medical problem - TrAP: A treatment administered for a medical problem - TrAP: The administration of a treatment was avoided because of a medical problem - TeRP: A test has revealed some medical problem - TeCP: A test was performed to investigate a medical problem - PIP: Two problems are related to each other
<a href="#">DDI Effects of using Multiple Drugs (Drug Drug Interaction)</a>	Predict multi-class effects, mechanisms and reasoning for DDI effects(Drug Drug Interaction) relationships between Drug Entities	relation.drug_drug_interaction	- DDI-advise when an advice/recommendation regarding a Drug Entity and Drug Entity is given - DDI-effect when Drug Entity and Drug Entity have an effect on the human body (pharmacodynamic mechanism); including a clinical finding, signs or symptoms, an increased toxicity or therapeutic failure. - DDI-int when effect between Drug Entity and Drug Entity is already known and thus provides no additional information. - DDI-mechanism when Drug Entity and Drug Entity are affected by an organism (pharmacokinetic). Such as the changes in levels or concentration in a drug. Used for DDIs that are described by their PK mechanism - DDI-false when a Drug Entity and Drug Entity have no interaction mentioned in the text.
<a href="#">Posology (Drugs, Dosage, Duration, Frequency, Strength)</a>	Predict multi-class posology relationships between Drug Entities, Dosage Entities, Strength Entities,Route Entities, Form Entities, Duration Entities and Frequency Entities	relation.posology	- DRUG-ADE if Problem Entity Adverse effect of Drug Entity - DRUG-DOSAGE if Dosage Entity refers to a Drug Entity - DRUG-DURATION if Duration Entity refers to a Drug Entity - DRUG-FORM if Mode/Form Entity refers to intake form of Drug Entity - DRUG-FREQUENCY if Frequency Entity refers to usage of Drug Entity - DRUG-REASON if Problem Entity is reason for taking Drug Entity - DRUG-ROUTE if Route Entity refer to administration method of Drug Entity - DRUG-STRENGTH if Strength Entity refers to Drug Entity
<a href="#">Chemicals and Proteins</a>	Predict Regulator, Upregulator, Downregulator, Agonist, Antagonist, Modulator, Cofactor, Substrate relationships between Chemical Entities and Protein Entities	relation.chemprot	- CPR:1 if One ChemProt Entity is Part of of Another ChemProt Entity - CPR:2 if One ChemProt Entity is Regulator (Direct or Indirect) of Another ChemProt Entity - CPR:3 if One ChemProt Entity is Upregulator/Activator/Indirect Upregulator of Another ChemProt Entity - CPR:4 if One ChemProt Entity is Downregulator/Inhibitor/Indirect Downregulator of Another ChemProt Entity - CPR:5 if One ChemProt Entity is Agonist of Another ChemProt Entity - CPR:6 if One ChemProt Entity is Antagonist of Another ChemProt Entity - CPR:7 if One ChemProt Entity is Modulator (Activator/Inhibitor) of Another ChemProt Entity - CPR:8 if One ChemProt Entity is Cofactor of Another ChemProt Entity - CPR:9 if One ChemProt Entity is Substrate and product of of Another ChemProt Entity - CPR:10 if One ChemProt Entity is Not Related to Another ChemProt Entity

## Relation Domain Examples

Domain	Sentence With Relationships	Predicted Relationships for Sample Sentence	Reference Links
Dates and Clinical Entities	This 73 y/o patient had CT on 1/12/95, with cognitive decline since 8/11/94.	- 1 for CT and 1/12/95 - 0 for cognitive decline and 1/12/95 - 1 for cognitive decline and 8/11/94	<a href="#">Internal Dataset by Annotated by John Snow Labs</a>
Body Parts and Directions	MRI demonstrated infarction in the upper - brain stem , left cerebellum and right basil ganglia	- 1 for upper and brain stem - 0 for upper and cerebellum - 1 for left and cerebellum	<a href="#">Internal Dataset by Annotated by John Snow Labs</a>
Body Parts and Problems	Patient reported numbness in his left hand and bleeding from ear.	- 1 for numbness and hand - 0 for numbness and ear - 1 for bleeding and ear	<a href="#">Internal Dataset by Annotated by John Snow Labs</a>
Body Parts and Procedures	The chest was scanned with portable ultrasound and amputation was performed on foot	- 1 for chest and portable ultrasound - 0 for chest and amputation - 1 for foot and amputation	<a href="#">Internal Dataset by Annotated by John Snow Labs</a>
Adverse Effects between drugs (ADE)	Taking Lipitor for 15 years, experienced much sever fatigue! Doctor moved me to voltaren 2 months ago , so far only experienced cramps	- 1 for sever fatigue and Lipitor - 0 for sever fatigue and voltaren - 0 for cramps and Lipitor - 1 for cramps and voltaren	<a href="#">Internal Dataset by Annotated by John Snow Labs</a>
Phenotype abnormalities,Genes and Diseases	She has a retinal degeneration, hearing loss and renal failure, short stature. Mutations in the SH3PXD2B gene coding for the Tks4 protein are responsible for the autosomal recessive.	- 1 for hearing loss and SH3PXD2B - 0 for retinal degeneration and hearing loss - 1 for retinal degeneration and autosomal recessive	<a href="#">PGR aciAntology</a>
Temporal events	She is diagnosed with cancer in 1991. Then she was admitted to Mayo Clinic in May 2000 and discharged in October 2001	- OVERLAP for cancer and 1991 - AFTER for admitted and Mayo Clinic - BEFORE for admitted and discharged	<a href="#">Temporal JSL Dataset and n2c2</a>
Dates and Tests/Results	On 23 March 1995 a X-Ray applied to patient because of headache, found tumor in brain	- is finding of for tumor and X-Ray - is result of for headache and X-Ray - is_date_of for 23 March 1995 and X-Ray	<a href="#">Internal Dataset by Annotated by John Snow Labs</a>
Clinical Problem, Treatment and Tests	- TrIP : infection resolved with antibiotic course - TrWP : the tumor was growing despite the drain - TrCP : penicillin causes a rash - TrAP : Dexamphetamine for narcolepsy - TrNAP : Ralafen was not given because of ulcers - TeRP : an echocardiogram revealed a pericardial effusion - TeCP : chest x-ray for pneumonia - PIP : Azotenia presumed secondary to sepsis	- TrIP for infection and antibiotic course - TrWP for tumor and drain - TrCP for penicillin and rash - TrAP for Dexamphetamine and narcolepsy - TrNAP for Ralafen and ulcers - TeRP for echocardiogram and pericardial effusion - TeCP for chest x-ray and pneumonia - PIP for Azotenia and sepsis	<a href="#">2010 I2b2 relation challenge</a>
DDI Effects of using Multiple Drugs (Drug Drug Interaction)	- DDI-advice: UROXATRAL should not be used in combination with other alpha-blockers - DDI-effect: Chlorthalidone may potentiate the action of other antihypertensive drugs - DDI-int: The interaction of omeprazole and ketoconazole has been established - DDI-mechanism: Grepafloxacin may inhibit the metabolism of the theobromine - DDI-false: Aspirin does not interact with Chlorthalidone	- DDI-advice for UROXATRAL and alpha-blockers - DDI-effect for Chlorthalidone and antihypertensive drugs - DDI-int for omeprazole and ketoconazole - DDI-mechanism for Grepafloxacin and theobromine - DDI-false for Aspirin and Chlorthalidone	<a href="#">DDI Extraction corpus</a>
Posology (Drugs, Dosage, Duration, Frequency, Strength)	- DRUG-ADE: had a headache after taking Paracetamol - DRUG-DOSAGE: took 0.5ML of Celstone - DRUG-DURATION: took Aspirin daily for two weeks - DRUG-FORM: took Aspirin as tablets - DRUG-FREQUENCY: Aspirin usage is weekly - DRUG-REASON : Took Aspirin because of headache - DRUG-ROUTE: Aspirin taken orally - DRUG-STRENGTH: 2mg of Aspirin	- DRUG-ADE for headache and Paracetamol - DRUG-DOSAGE for 0.5ML and Celstone - DRUG-DURATION for Aspirin and for two weeks - DRUG-FORM for Aspirin and tablets - DRUG-FREQUENCY for Aspirin and weekly - DRUG-REASON for Aspirin and headache - DRUG-ROUTE for Aspirin and orally - DRUG-STRENGTH for 2mg and Aspirin	<a href="#">Magie, Scotch, Gonzalez-Hernandez (2018)</a>
Chemicals and Proteins	- CPR:1 (Part of) : The amino acid sequence of the rabbit alpha(2A)-adrenoceptor has many interesting properties. - CPR:2 (Regulator) : Triacsin inhibited ACS activity - CPR:3 (Upregulator) : Ibandronate increases the expression of the FAS gene - CPR:4 (Downregulator) : Vitamin C treatment resulted in reduced C-Rel nuclear translocation - CPR:5 (Agonist) : Reports show tricyclic antidepressants act as agonists at distinct opioid receptors - CPR:6 (Antagonist) : GDC-0152 is a drug triggers tumor cell apoptosis by selectively antagonizing LAPs - CPR:7 (Modulator) : Hydrogen sulfide is a allosteric modulator of ATP-sensitive potassium channels - CPR:8 (Cofactor) : polyinosinic:polycytidylic acid and the IFNa/β demonstrate capability of endogenous IFN. - CPR:9 (Substrate) : ZIP9 plays an important role in the transport and toxicity of Cd(2+) cells - CPR:10 (Not Related) : Studies indicate that GSK-3β inhibition by palbutorin cannot be competed out by ATP	- CPR:1 (Part of) for amino acid and rabbit alpha(2A)-adrenoceptor - CPR:2 (Regulator) for Triacsin and ACS - CPR:3 (Upregulator) for Ibandoante and FAS gene - CPR:4 (Downregulator) for Vitamin C and C-Rel - CPR:5 (Agonist) for tricyclic antidepressants and opioid receptors - CPR:6 (Antagonist) (Antagonist) for GDC-0152 and LAPs - CPR:7 (Modulator) for Hydrogen sulfide and ATP-sensitive potassium channels - CPR:8 (Cofactor) for polyinosinic:polycytidylic acid and IFNa/β - CPR:9 (Substrate) for ZIP9 and Cd(2+) cells - CPR:10 (Not Related) for GSK-3β and ATP	<a href="#">ChemProt Paper</a>

# More NLU Ressources

- [Join our Slack](#)
- [NLU Website](#)
- [NLU Github](#)
- [Many more NLU example tutorials](#)
- [Overview of every powerful nlu 1-liner](#)
- [Checkout the Modelshub for an overview of all models](#)
- [Checkout the NLU Namespace where you can find every model as a tabel](#)
- [Intro to NLU article](#)
- [In Depth and Easy Sentence Similarity Tutorial, with StackOverflow Questions using BERTology embeddings](#)
- [1 line of Python code for BERT, ALBERT, ELMO, ELECTRA, XLNET, GLOVE, Part of Speech with NLU and t-SNE](#)

## Webinars and Videos

- [NLU & Streamlit Tutorial](#)
- [Crash course of the 50 + Medical Domains and the 200+ Healthcare models in NLU](#)
- [Multi Lingual NLU Webinar - Working with Chinese Text Data](#)
- [John Snow Labs NLU: Become a Data Science Superhero with One Line of Python code](#)
- [Python Web Def Conf - Python's NLU library: 4,000+ Models, 200+ Languages, State of the Art Accuracy, 1 Line of Code](#)
- [NYC/DC NLP Meetup with NLU](#)

# Session 4 (Day 2) - Coding Time

- ❖ [Notebook 13 - NLU Crash course, every Spark NLP model in 1 line](#)

# NLP Server - Powerful private deployable NLP Rest API

A screenshot of the AWS Marketplace search results for 'John Snow Labs'. The search bar at the top contains 'John Snow Labs'. The results page shows 15 items under the heading 'john Snow Labs (15 results) showing 1 - 15'. A red arrow points from the left margin towards the first item in the list, 'John Snow Labs - Annotation Lab'. The sidebar on the left has a 'Refine results' section with filters for Categories, Delivery methods, Publisher, Pricing model, Pricing unit, Operating system, End User License, Architecture, Region, and more. The main content area displays five product cards:

- John Snow Labs - Annotation Lab**: By John Snow Labs. Price: \$179 / 12 months. Description: An annotation lab is a robust data labeling and AI/ML solution for the cloud. It enables customers to annotate their data and generate their models in a simple, fast and efficient project based workflow. Features - Human-in-the-loop AI - High productivity annotation UI with keyboard shortcuts and more.
- John Snow Labs - NLP Server**: By John Snow Labs. Price: \$179 / 12 months. Description: Ready to use NLP Server for analyzing text documents using NLU library. All Spark NLP pre-trained models and pipelines are easy to use via a simple and intuitive UI, without writing a line of code. For more expert users and more complex tasks, NLP Server also provides a REST API that can be used to...
- Diagnosed Diabetes Prevalence 2004-2013**: By John Snow Labs. Price: \$179 / 12 months. Description: This dataset contains number and percentage of diabetes patients in the US during 2013 grouped by ZIP code.
- Respiratory Syncytial Virus Disease**: By John Snow Labs. Price: \$179 / 12 months. Description: This dataset provides replication data for the absence of the association between cord specific antibody levels and severe respiratory syncytial virus (RSV) disease in early infants from a case-control study from coastal Kenya.
- Dobutamine Stress Echocardiography Data**: By John Snow Labs. Price: \$179 / 12 months. Description: The dataset contains the complete data for the final study population for the Dobutamine Stress Echocardiography test which includes 554 subjects and 31 variables.
- Studies On Safe Drugs During Pregnancy**: By John Snow Labs. Price: \$179 / 12 months. Description: This dataset contains a list of the current Pregnancy Registries in research in the United States, their status and contact information. Pregnancy exposure registries are observational studies that focus on the outcome of pregnancies of women with medical conditions that require taking...

## John Snow Labs - NLP Server

By: [John Snow Labs](#) Latest Version: NLP Server 0.3.0

Ready to use NLP Server for analyzing text documents using NLU library. All Spark NLP pre-trained models and pipelines are easy to use via a simple and intuitive UI, without writing a line of code. For more expert users and more complex tasks, NLP Server also provides a REST API that can be used to...

[Show more](#)

Linux/Unix

[Continue to Subscribe](#)

[Save to List](#)

Typical Total Price

**\$0.384/hr**

Total pricing per instance for services hosted on m5.2xlarge in US East (N. Virginia). [View Details](#)

## Overview

## Pricing

## Usage

## Support

## Reviews

## Product Overview

Ready to use NLP Server for analyzing text documents using NLU library. All Spark NLP pre-trained models and pipelines are easy to use via a simple and intuitive UI, without writing a line of code. For more expert users and more complex tasks, NLP Server also provides a REST API that can be used to process high amounts of data.

With NLP Server you get access to state-of-the-art 3000+ models in over 200+ languages for problems like Sentiment Analysis, Spell Checking, Text Summarization, Translation, Named Entity Recognition, Question Answering, Spam Classifiers and much more! All those features are available in any programming language via simple Rest API calls.

Exploit the latest research developments in NLP from the Transformers world with LongFormers, XLM-RoBERTa, XLING, Electra, LaBSE, DistilBERT, XLNET , USE, T5, Marian and much more!

## Highlights

- Spark NLP text annotation

### Version

**NLP Server 0.3.0**  
[Show other versions](#)

### By

**John Snow Labs**

### Categories

**Natural Language Processing**  
Text

### Operating System

**Linux/Unix, Ubuntu 20.04**

### Delivery Methods

**Amazon Machine Image**

[https://nlp.johnsnowlabs.com/docs/en/nlp\\_server/nlp\\_server](https://nlp.johnsnowlabs.com/docs/en/nlp_server/nlp_server)

4000+ Models in 200 languages available as API and GUI with a few clicks in the AWS marketplace.

This enables you to use all of these Models with any programming Language by simple using REST API calls

### API Usage in 3 Steps :

- Query - Send Data request
- Poll - Is server done processing?
- Get result - Get result when server done

#### 1. Choose a spell

en.class

**en.classify.snips**  
Detect actions in general commands related to music, restaurant, movies.

**en.classify.spam**  
Spam Classifier

**en.classify.spam.use**  
Spam Classifier

**en.classify.toxic**  
Toxic Comment Classification

**en.classify.toxic.sm**  
Toxic Comment Classification - Small

**en.classify.trec50**  
TREC(50) Question Classifier

**en.classify.trec50.relp**

#### 1. Choose a spell

en.ner

Recognize Entities DL Pipeline for English

#### 2. Enter the Text to analyze

The president of the United States is th... affiliated with a political party.[2]

There are five living former presidents. The most recent to die was George H. W. Bush, on November 30, 2018.]



Group results by:  No grouping  Document  Sentence  Entity  Word

#### 3. Preview the Results:

index	entities_class	document	entities	word_embedding_ner	entities_confidence
0	LOC	The president of the United States is th...	United States	-0.0381940007,-0.244870007,0.728120029,-...	0.95000005
0	LOC	The president of the United States is th...	United States	-0.0381940007,-0.244870007,0.728120029,-...	0.90639997
0	MISC	The president of the United States is th...	American	-0.0381940007,-0.244870007,0.728120029,-...	0.9992
0	ORG	The president of the United States is th...	Electoral College	-0.0381940007,-0.244870007,0.728120029,-...	0.95365
0	ORG	The president of the United States is th...	United States Armed Forces	-0.0381940007,-0.244870007,0.728120029,-...	0.829475

## Step 1 : Post a request

- Post a data processing request
- Returns UUID status

POST /api/results/

Request samples

Payload

Content type  
application/json

Copy   Expand all   Collapse all

```
{  
    "spell": "tokenize",  
    "data": "I love NLU! <3"  
}
```

Response samples

200   404   422

Content type  
application/json

Copy   Expand all   Collapse all

```
{  
    "uuid": "njGeaDKl1VoRR7kjf002h"  
}
```

## Step 2 : Poll for result

- Poll Server for processing status

GET /api/results/{uuid}/status

▼

Response samples

200 404 422

Content type  
application/json

Copy Expand all Collapse all

```
{  
  - "status": {  
      "code": "progress",  
      "message": "Predicting..."  
    }  
}
```

## Step 3 : Get result as JSON

- Get result for UUID
- Contains NLP predictions

GET /api/results/{uuid}

Response samples

200 404 422

Content type  
application/json

Copy Expand all Collapse all

```
{  
  - "sentence": {  
      + "0": [ ... ]  
    },  
  - "document": {  
      "0": "I love NLU! <3"  
    },  
  - "token": {  
      + "0": [ ... ]  
    },  
  - "text": {  
      "0": "I love NLU! <3"  
    }  
}
```

# All of Spark NLP features behind a simple to use API. deployable in a few clicks on AWS

```
NLP_server_cheatsheet.py

import requests
# Invoke Processing with tokenization spell
r = requests.post(f'http://localhost:5000/api/results',json={"spell": "tokenize","data": "I love NLU!
<3"})
# Use the uuid to get your processed data
uuid = r.json()['uuid']

# Get status of processing
r = requests.get(f'http://localhost:5000/api/results/{uuid}/status').json
>>> {'status': {'code': 'success', 'message': None}}

# Get results
r = requests.get(f'http://localhost:5000/api/results/{uuid}').json()
>>> {'sentence': {'0': ['I love NLU! <3']}, 'document': {'0': 'I love NLU! <3'}, 'token': {'0': ['I',
'love', 'NLU', '!', '<3']}}
```

- [https://nlp.johnsnowlabs.com/docs/en/nlp\\_server/nlp\\_server](https://nlp.johnsnowlabs.com/docs/en/nlp_server/nlp_server)
- [Youtube Tutorial for NLP Server Deployment on AWS](#)

# Thank You!