4 days, 16 hrs live training

Certification exams in 2 weeks

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

Veysel Kocaman
Principal Data Scientist
veysel@johnsnowlabs.com

Spark NLP for Data Scientists

April 12-13, 2022

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

Welcome - We have a lot of things ahead of us

Day-1	60 min	- Intro to John Snow Labs, Spark NLP and NLP Theory - Spark NLP Pretrained Pipelines basics and JVM/Spark concepts
	10 min	Break
	50 min	- Text Preprocessing and composing custom pipeline in Spark NLP
	10 min	Break
	50 min	- Showcase of the 5000+ pretrained Models for 200+ languages in Spark NLP
		Break
	50 min	- Train Named Entity Recognizers (NER) and Classifier models - Upload to Modelshub

Welcome - We have a lot of things ahead of us

Day-2	50 min	- Unsupervised Keyword Extraction with YAKE - Sentence Detection - Spell Checking - Graph triplet extraction
	10 min	Break
	60 min	- Sequence Classification with Transformers - Token Classification with Transformers - GPT2 - Conditional Text generation - Text Style Transfer and SQL code generation from Natural Language Text with T5
	10 min	Break
	60 min	- Question Answering, Summarization and other T5 applications - Multilingual NLP - Train only on English data and predict for 100+ languages - Huggingface and Tensorflow Hub to Spark NLP export
	10 min	Break
	50 min	- The Python NLU library basics, use any of the 4000+ models in 1 line of code - NLU & Streamlit - NLP Server - NLU OCR

Part - I (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

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

John Snow LABS

Home Docs Learn Models Demo

John Snow Labs NLP Documentation

Spark NLP

Healthcare NLP

Spark OCR

Annotation Lab

NLP Server

NLU

https://nlp.johnsnowlabs.com/docs

John Snow LABS

Home Docs Learn Models Demo

John Snow Labs NLP Documentation

Spark NLP

Healthcare NLP

Spark OCR

Annotation Lab

NLP Server

NLU

https://nlp.johnsnowlabs.com/docs

Setup

pip install spark-nlp

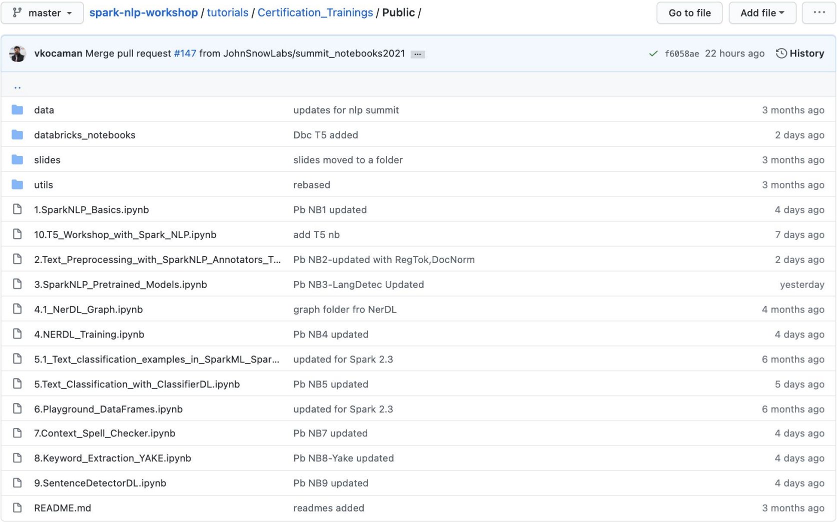
Open in Colab

```
!pip install pyspark==3.1.2 spark-nlp==3.4.2
```

RUNNING CODE:

https://github.com/JohnSnowLabs/spark-nlp-workshop/blob/master/tutorials/Certification_Trainings/Public

[How to set up Google Colab]



master · spark-nlp-workshop / tutorials / Certification_Trainings / Public / · Go to file · Add file

vkocaman Merge pull request #147 from JohnSnowLabs/summit_notebooks2021 … ✓ f6058ae 22 hours ago History

..		
data	updates for nlp summit	3 months ago
databricks_notebooks	Dbc T5 added	2 days ago
slides	slides moved to a folder	3 months ago
utils	rebased	3 months ago
1.SparkNLP_Basics.ipynb	Pb NB1 updated	4 days ago
10.T5_Workshop_with_Spark_NLP.ipynb	add T5 nb	7 days ago
2.Text_Preprocessing_with_SparkNLP_Annotators_T...	Pb NB2-updated with RegTok,DocNorm	2 days ago
3.SparkNLP_Pretrained_Models.ipynb	Pb NB3-LangDetec Updated	yesterday
4.1_NerDL_Graph.ipynb	graph folder fro NerDL	4 months ago
4.NERDL_Training.ipynb	Pb NB4 updated	4 days ago
5.1_Text_classification_examples_in_SparkML_Spar...	updated for Spark 2.3	6 months ago
5.Text_Classification_with_ClassifierDL.ipynb	Pb NB5 updated	5 days ago
6.Playground_DataFrames.ipynb	updated for Spark 2.3	6 months ago
7.Context_Spell_Checker.ipynb	Pb NB7 updated	4 days ago
8.Keyword_Extraction_YAKE.ipynb	Pb NB8-Yake updated	4 days ago
9.SentenceDetectorDL.ipynb	Pb NB9 updated	4 days ago
README.md	readmes added	3 months ago

BOOKMARK:

https://nlp.johnsnowlabs.com/
https://nlp.johnsnowlabs.com/docs/en/quickstart
spark-nlp.slack.com

https://github.com/JohnSnowLabs/spark-nlp-workshop/blob/master/tutorials/1hr_workshop/SparkNLP_openSource_workshop_1hr.ipynb

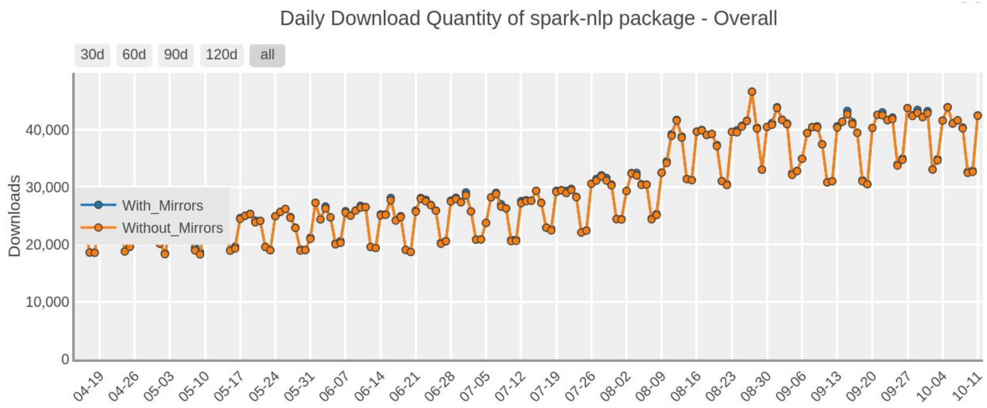
Introducing Spark NLP

Total downloads	14,558,265
Total downloads - 30 days	1,371,781
Total downloads - 7 days	318,804

downloads 14M

downloads/month 1M

downloads/week 318k

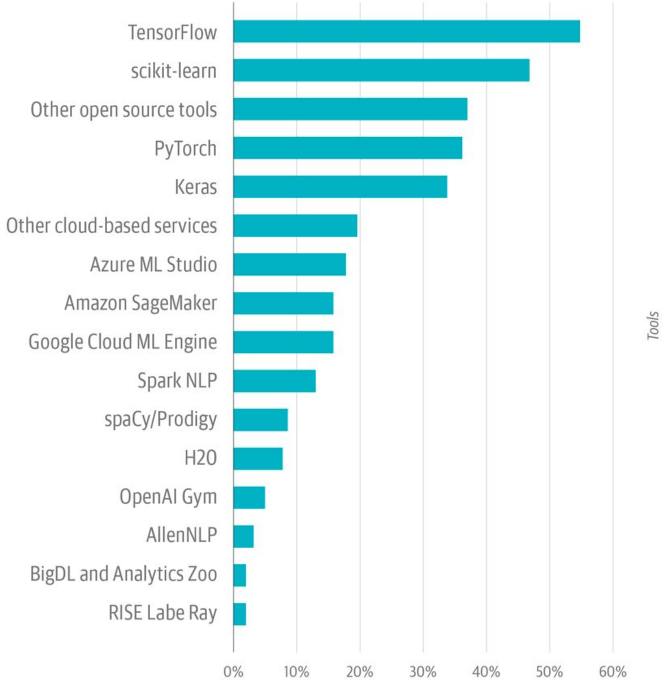


- Spark NLP is an open-source natural language processing library, built on top of Apache Spark and Spark ML. (initial release: Oct 2017)
  - A single unified solution for all your NLP needs
  - Take advantage of transfer learning and implementing the latest and greatest **SOTA** algorithms and models in NLP research
  - The most widely used NLP library in industry (5 yrs in a row)
  - Delivering a mission-critical, enterprise grade NLP library (used by multiple Fortune 500)
  - Full-time development team (30 new releases in 2021, 26 new releases in 2020, 30 new releases in 2019.)

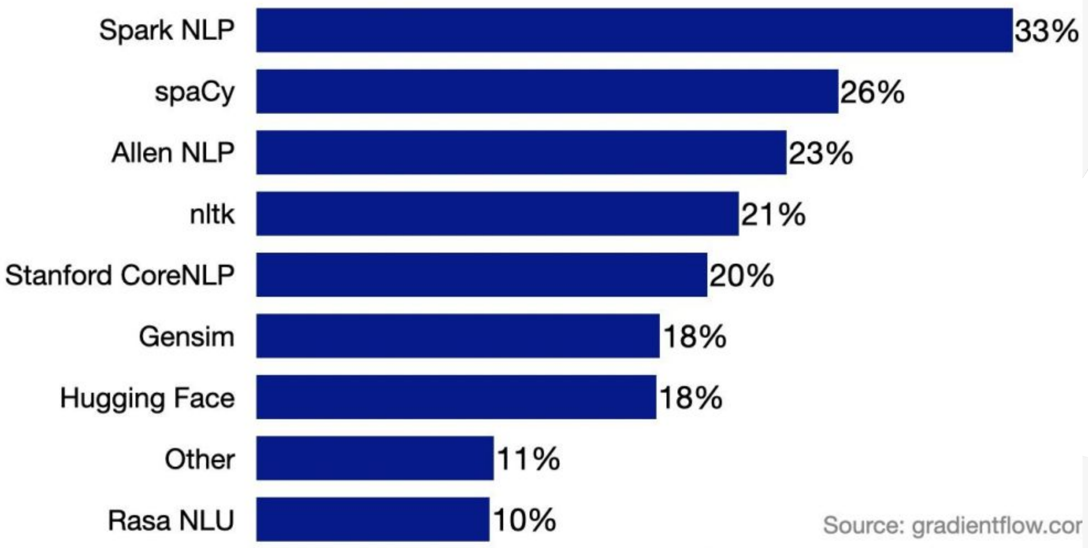
https://medium.com/spark-nlp/introduction-to-spark-nlp-foundations-and-basic-components-part-i-c83b7629ed59

Spark NLP in Industry

Which of the following AI tools do you use?



Which NLP libraries does your organization use?



NLP Industry Survey by Gradient Flow,
an independent data science research & insights company, September 2020

TRUSTED BY

Microsoft	Google	amazon	intel
IBM	databricks	verizon	indeed
Capital One	viacom	McKESSON	MERCK
Roche	selectdata	UiPath Reboot Work.	ASCO CANCER·LINQ DISCOVERY
cnrs	Imperial College London	Georgia Tech	STANFORD UNIVERSITY

Recognized by the Technology Experts

Global 100
2021
WINNER
John Snow LABS

ghp
2021 Global
Excellence Awards

2020
corporatevision
Best Healthcare
& Life Science AI
Development
Firm – USA

AI
ARTIFICIAL INTELLIGENCE
EXCELLENCE
AWARD
2020

International
DataScience
Awards
WINNER 2019

Strata
Data
Awards
2019
Most Significant
Open Source Project
WINNER
SPARK NLP

CIO Review
ARTIFICIAL
INTELLIGENCE
SOLUTION PROVIDER
OF THE YEAR - 2018

AI
PLATFORM
OF THE YEAR
CIO
APPLICATIONS
2019

Optimized, Tested, Supported Integrations

Java
python
jupyter
ANACONDA
kubernetes
APACHE Spark
Scala
R
docker
Google colab
CLOUDERA
oneAPI
Cloud Dataproc
intel
databricks
Amazon SageMaker
comet
amazon EMR
mlflow
kaggle
Google Cloud
Microsoft Azure
NVIDIA
Synapse ML
Simple and Distributed Machine Learning

Spark NLP

- 99 total releases
- Release every two weeks for the past 4 years
- A single unified library for all your NLP/NLU need
- Active community on Slack and GitHub

NLP Feature	Spark NLP	spaCy	NLTK	CoreNLP	Hugging Face
Tokenization	Yes	Yes	Yes	Yes	Yes
Sentence segmentation	Yes	Yes	Yes	Yes	No
Steeming	Yes	Yes	Yes	Yes	No
Lemmatization	Yes	Yes	Yes	Yes	No
POS tagging	Yes	Yes	Yes	Yes	No
Entity recognition	Yes	Yes	Yes	Yes	Yes
Dep parser	Yes	Yes	Yes	Yes	No
Text matcher	Yes	Yes	No	No	No
Date matcher	Yes	No	No	No	No
Sentiment detector	Yes	No	Yes	Yes	Yes
Text classification	Yes	Yes	Yes	No	Yes
Spell checker	Yes	No	No	No	No
Language detector	Yes	No	No	No	No
Keyword extraction	Yes	No	No	No	No
Pretrained models	Yes	Yes	Yes	Yes	Yes
Trainable models	Yes	Yes	Yes	Yes	Yes
Question Answering	Yes	Yes	No	No	Yes
Text Style Transfer	Yes	No	No	No	Yes
Finance models	Yes	No	No	No	Yes
200+ Languages supported	Yes	Yes	No	No	Yes
Summarize Test	Yes	Yes	No	No	Yes
Text Generation (GPT2, T5)	Yes	Yes	No	No	Yes

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

Accurate Clinical and Biomedical Named Entity Recognition at Scale

Anonymous NAACL-HLT 2021 submission

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-

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

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
{veysel, david}@johnsnowlabs.com

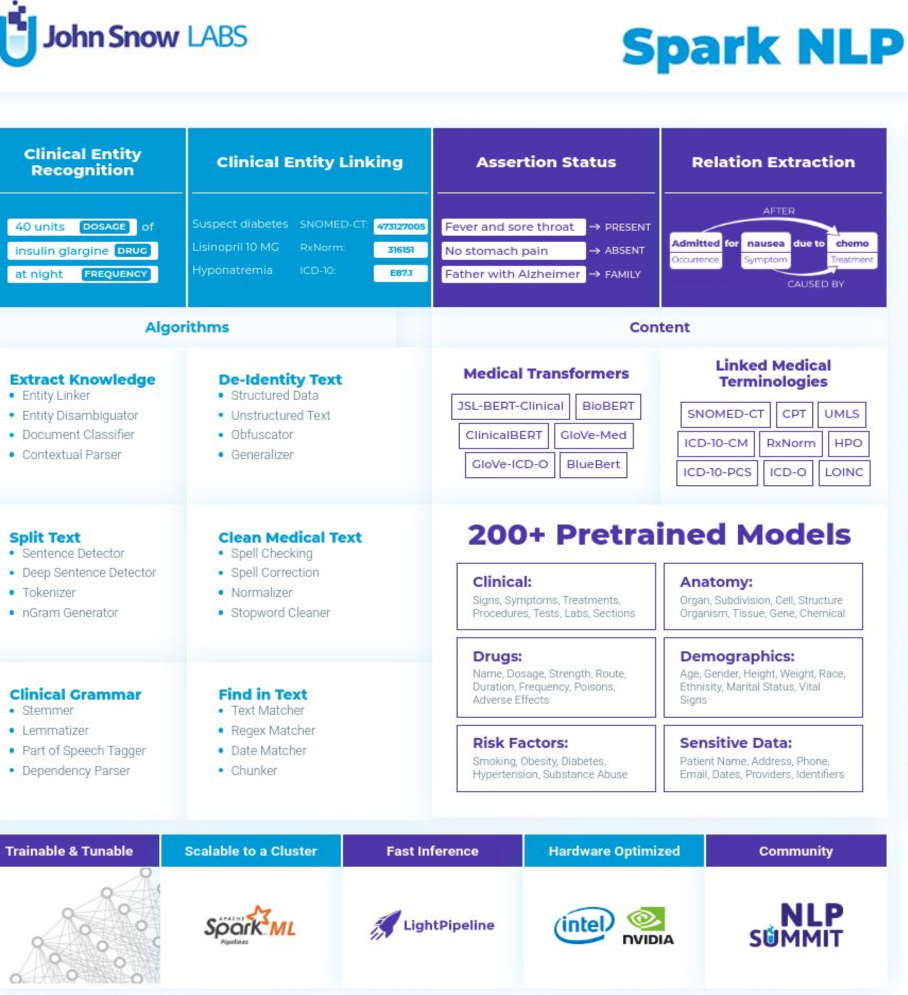
Abstract

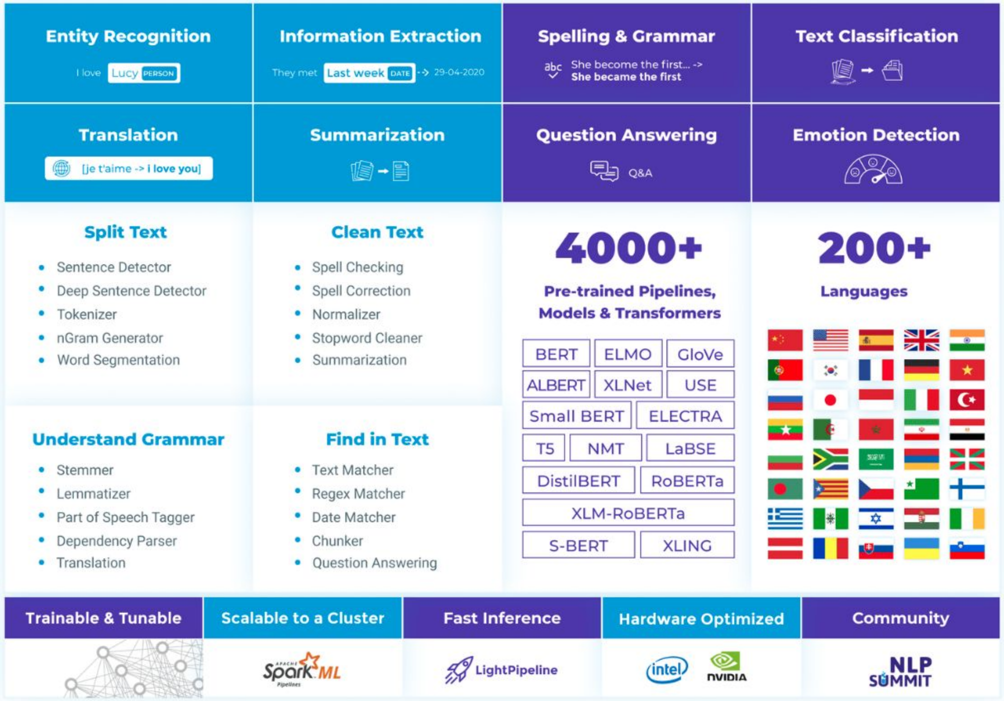
Following the global COVID-19 pandemic, the number of scientific papers studying the virus has grown massively, leading to increased interest in automated literate review. We present a clinical text mining system that improves on previous efforts in three ways. First, it can recognize over 100 different 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 conference papers on Spark NLP NER

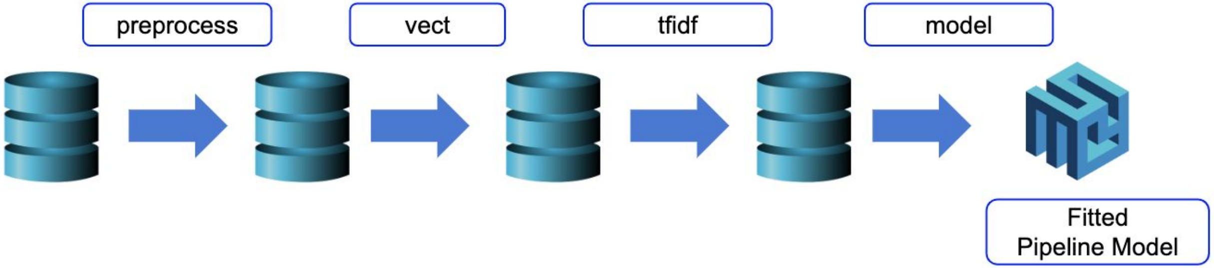


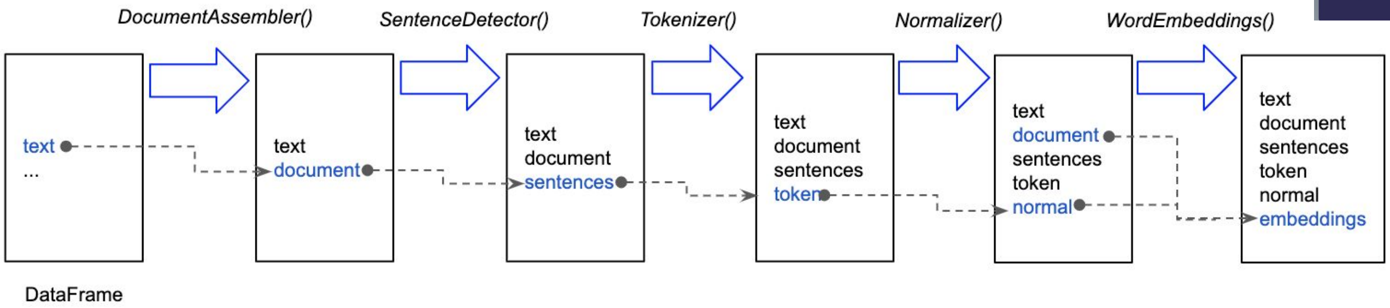


Spark NLP Modules (Enterprise and Public)

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)
```

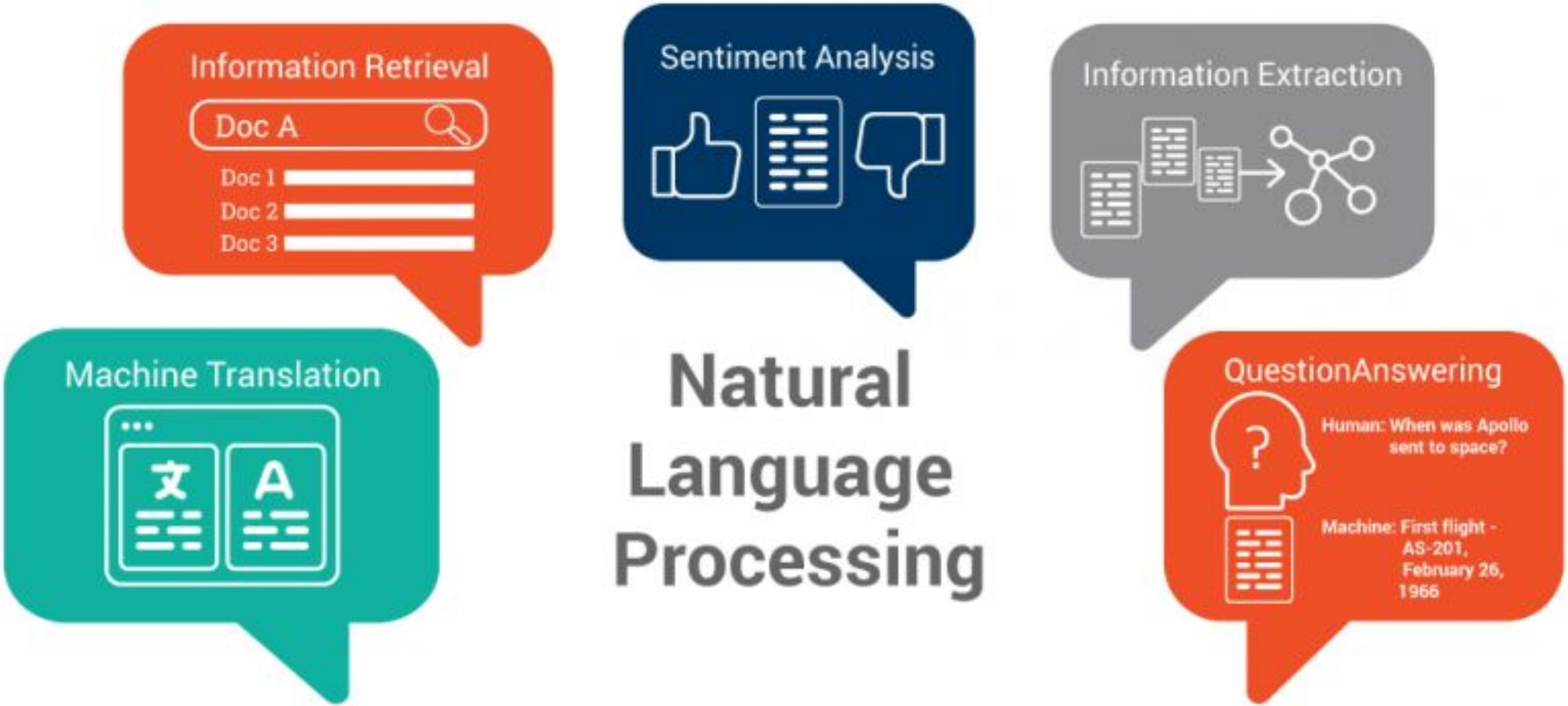
Introducing Spark NLP

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

Faster inference

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

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



Information Retrieval
Doc A
Doc 1
Doc 2
Doc 3
Sentiment Analysis
Information Extraction
Machine Translation
文
A
Natural Language Processing
QuestionAnswering
?
Human: When was Apollo sent to space?
Machine: First flight - AS-201, February 26, 1966

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
studying	-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

Remove stop words and apply stemming

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

Stopwords

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

(520 stopwords)

- 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.

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

NORMALIZATION

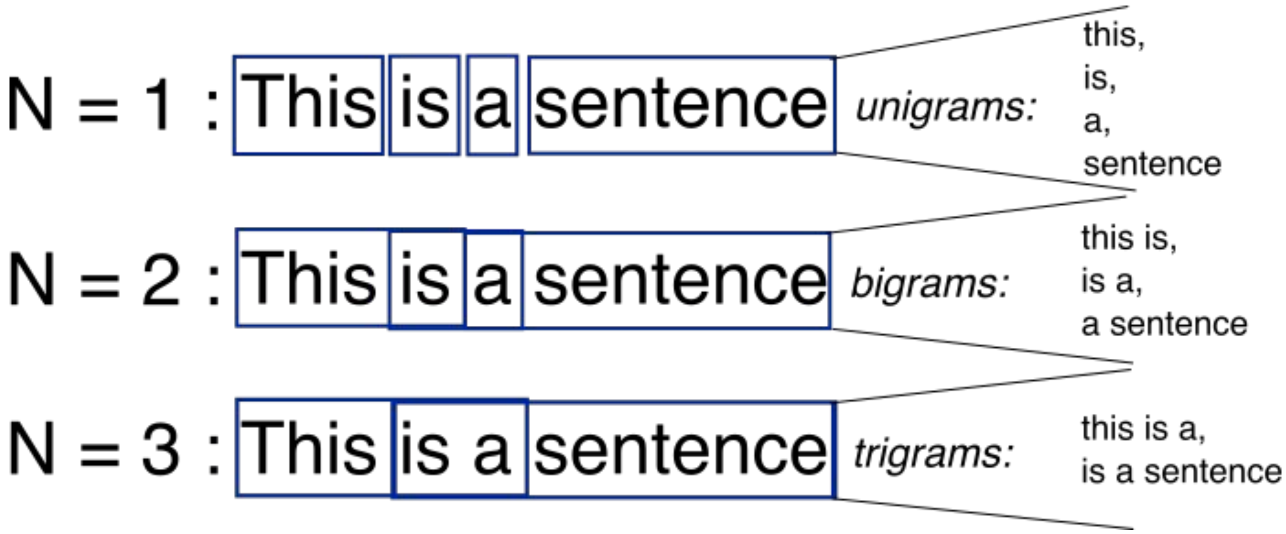
Remove or replace undesirable characters or regular expressions:

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

to: `Have a great birth day!`

Spark NLP also comes with a Slang normalizer:

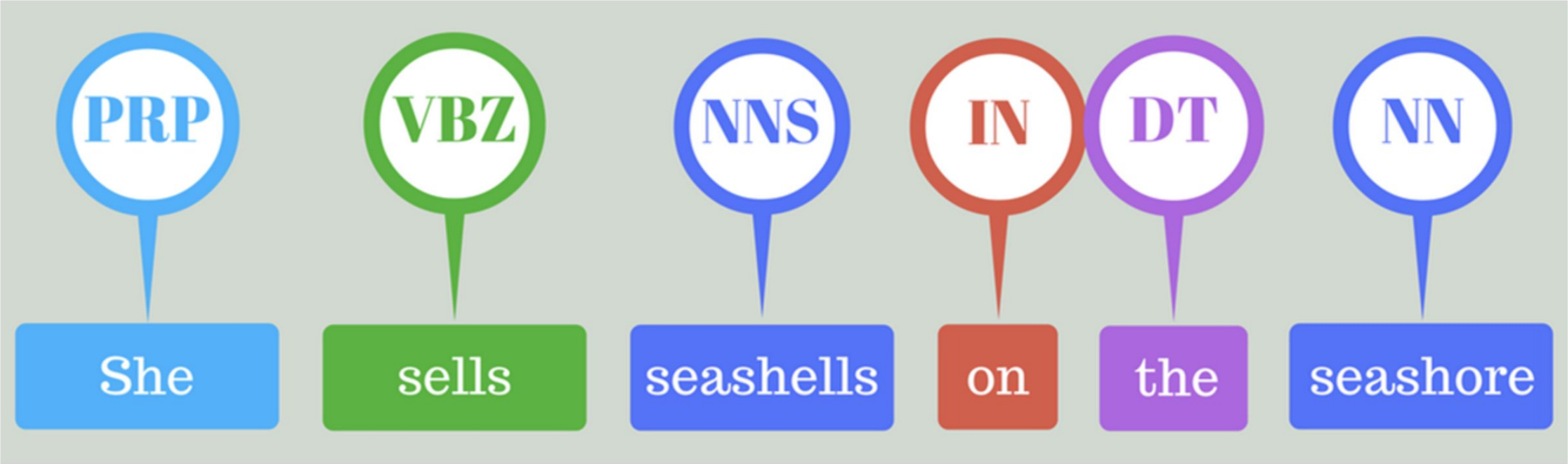
N-gram Tokenization



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

PART OF SPEECH TAGGING

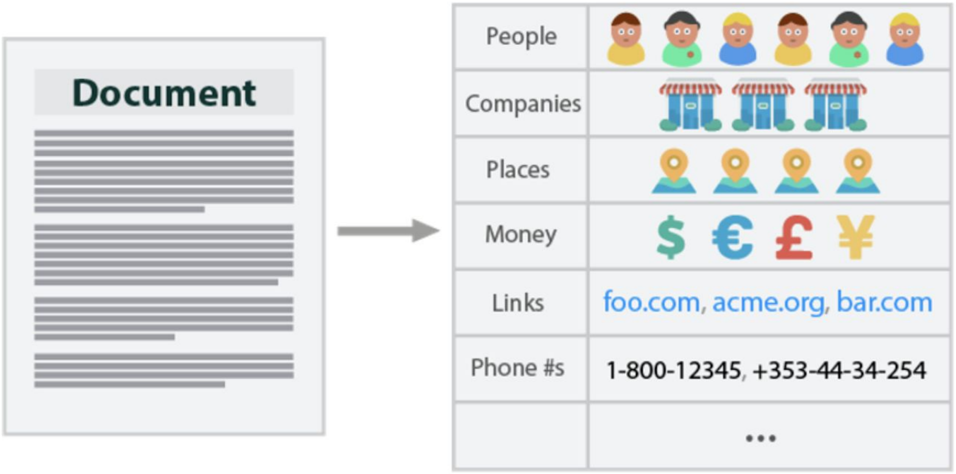
Often useful for recognizing named entities or word relationships.

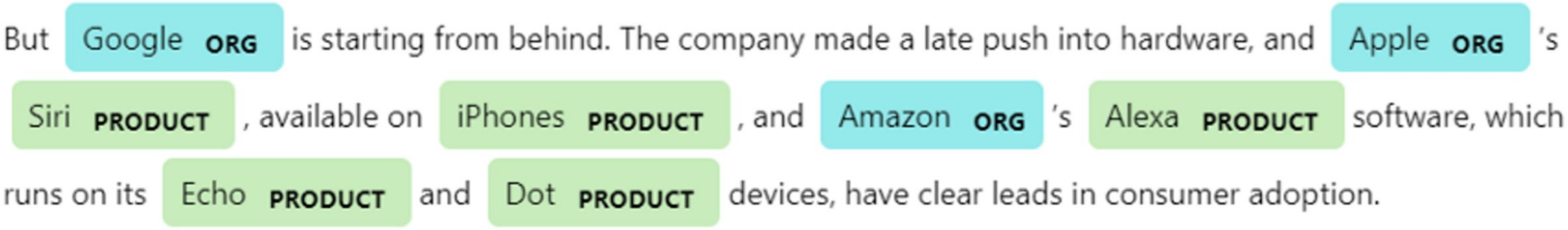


A **POS tag** (or **part-of-speech tag**) is a special label assigned to each token (word) in a text corpus to indicate the **part of speech** and often also other grammatical categories such as tense, number (plural/singular), case etc.

Named Entity Recognition (NER)

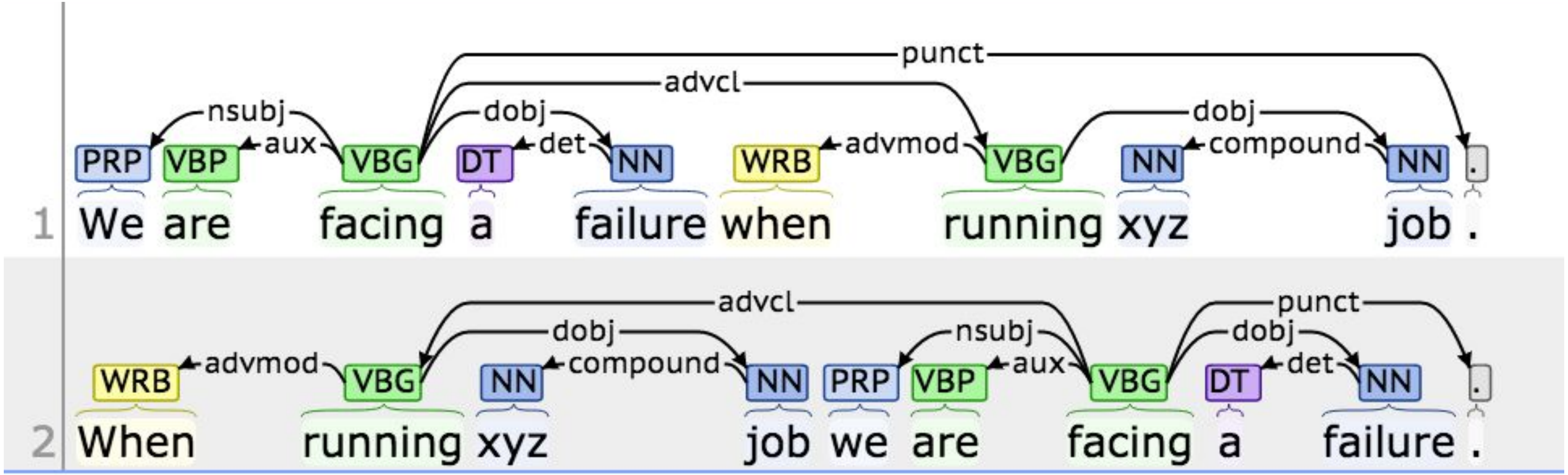
NER is a subtask of information extraction that seeks to locate and classify named entity mentioned in unstructured text into pre-defined categories such as person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc.





DEPENDENCY PARSING

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



Part - I (Day 1) - Coding Time

- [Notebook 1 - Spark NLP Basics]

Spark NLP for Data Scientists

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

Part - II (Day 1)

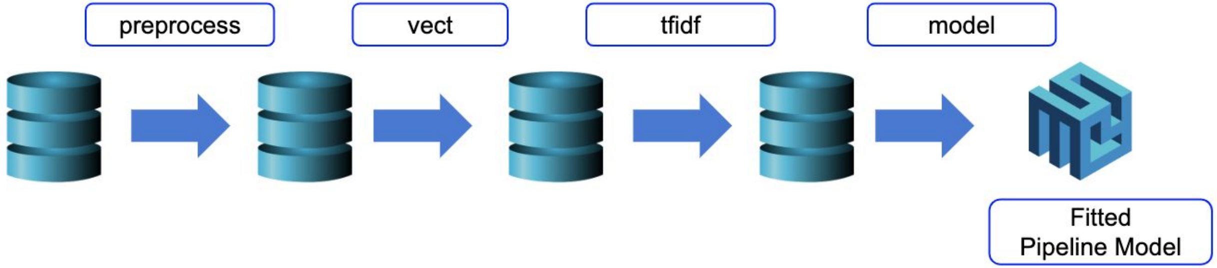
- Text Preprocessing
- Composing Pipelines
- Working with Spark Dataframes
- Notebook 2 Text Preprocessing and composing Pipelines

Spark NLP
for Data Scientists

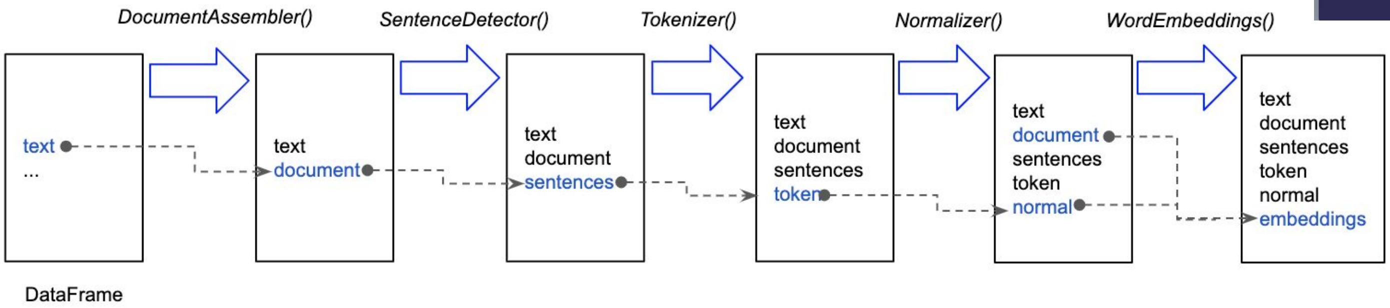
Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

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)
```



Part - II (Day 1) - Coding Time

- Notebook 2 Text Preprocessing and composing Pipelines

Spark NLP
for Data Scientists

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

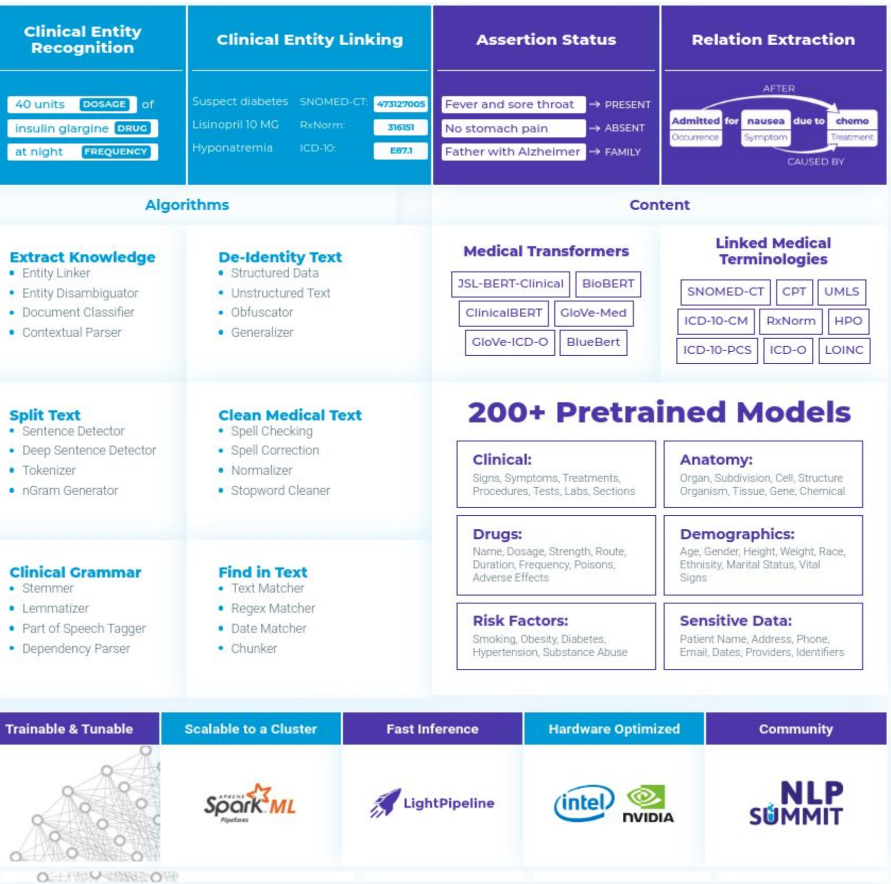
Part - III (Day 1)

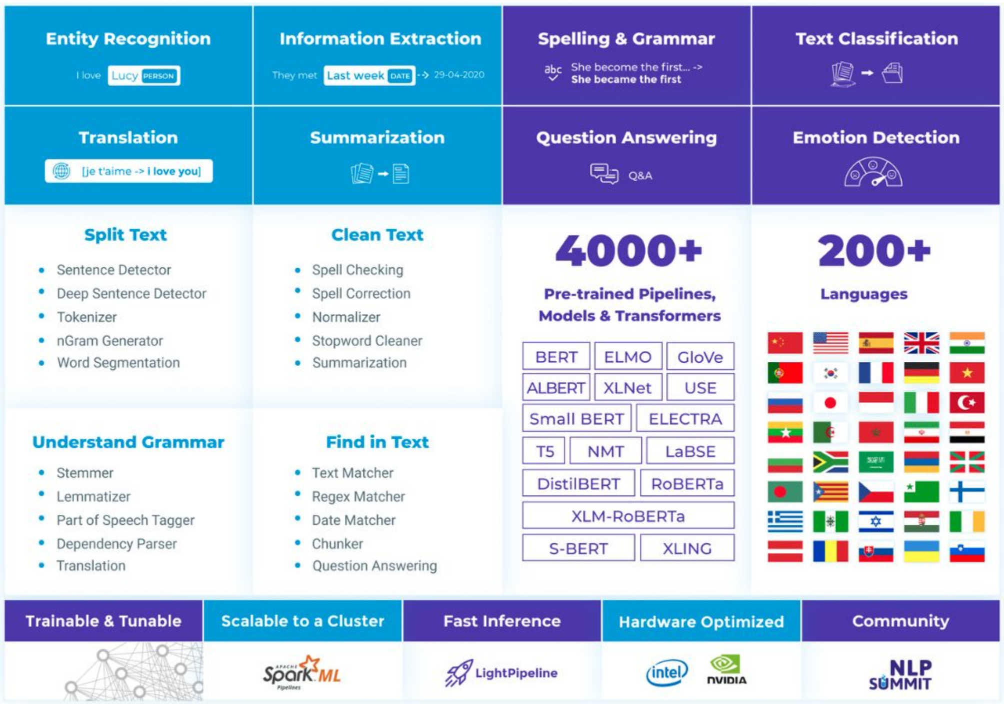
- Usage and overview of the 4000+ pretrained models for 200+ languages
- Notebook 3 Spark NLP pretrained models

Spark NLP for Data Scientists

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

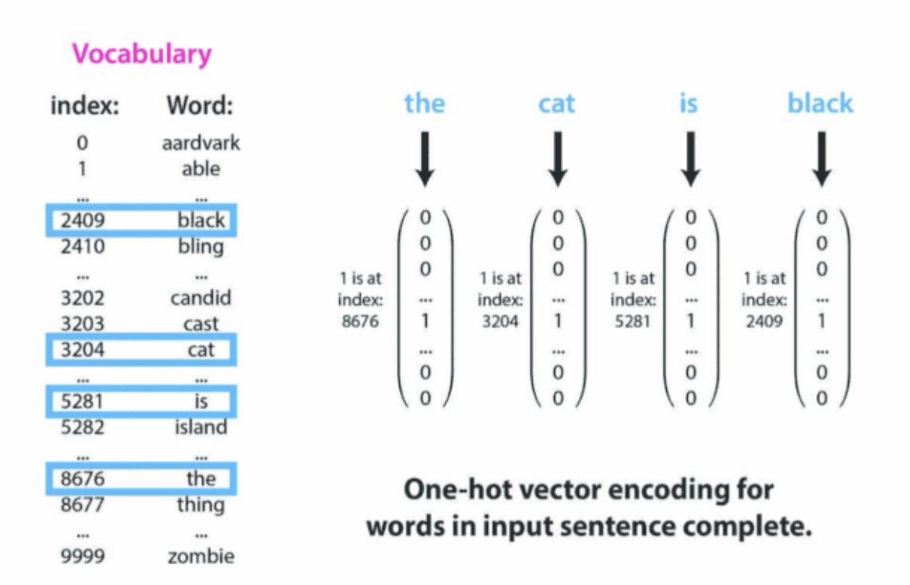
Spark NLP Modules





Spark NLP Modules (Enterprise and Public)

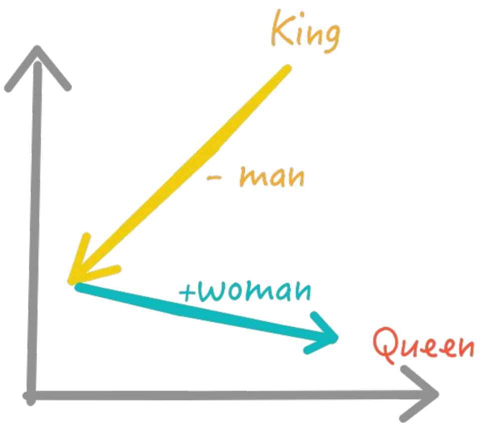
Word & Sentence Embeddings



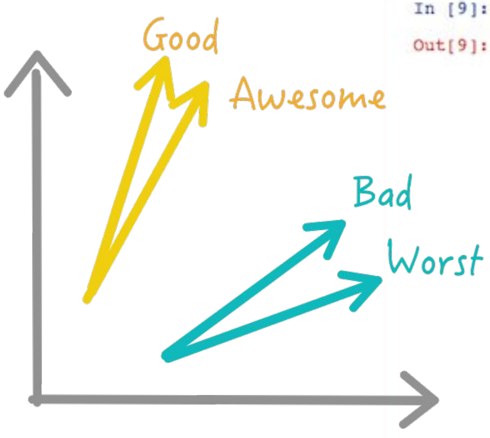

```
In [9]: doc[3].vector

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 ,
               -0.3617   , -0.18355  , -0.34717  , -0.3691   ,  0.16881  ,
                0.21018  , -0.38376  , -0.096909 , -0.36296  , -0.37319  ,
                0.0021152,  0.32512  ,  0.063977 ,  0.36249  , -0.26935  ,
               -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  ,  0.44076  ,  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  ,
               -0.19228  ,  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 , -0.10066  ,
               -0.4165   , -0.20252  ,  0.2794   ,  0.10852  , -0.40154  ,
```

- Words that are used in similar contexts will be given similar representations. That is, words that are used in similar ways will be placed close together within the high-dimensional semantic space–these points will cluster together, and their distance to each other will be low.

Word & Sentence Embeddings



a) Learns Analogy

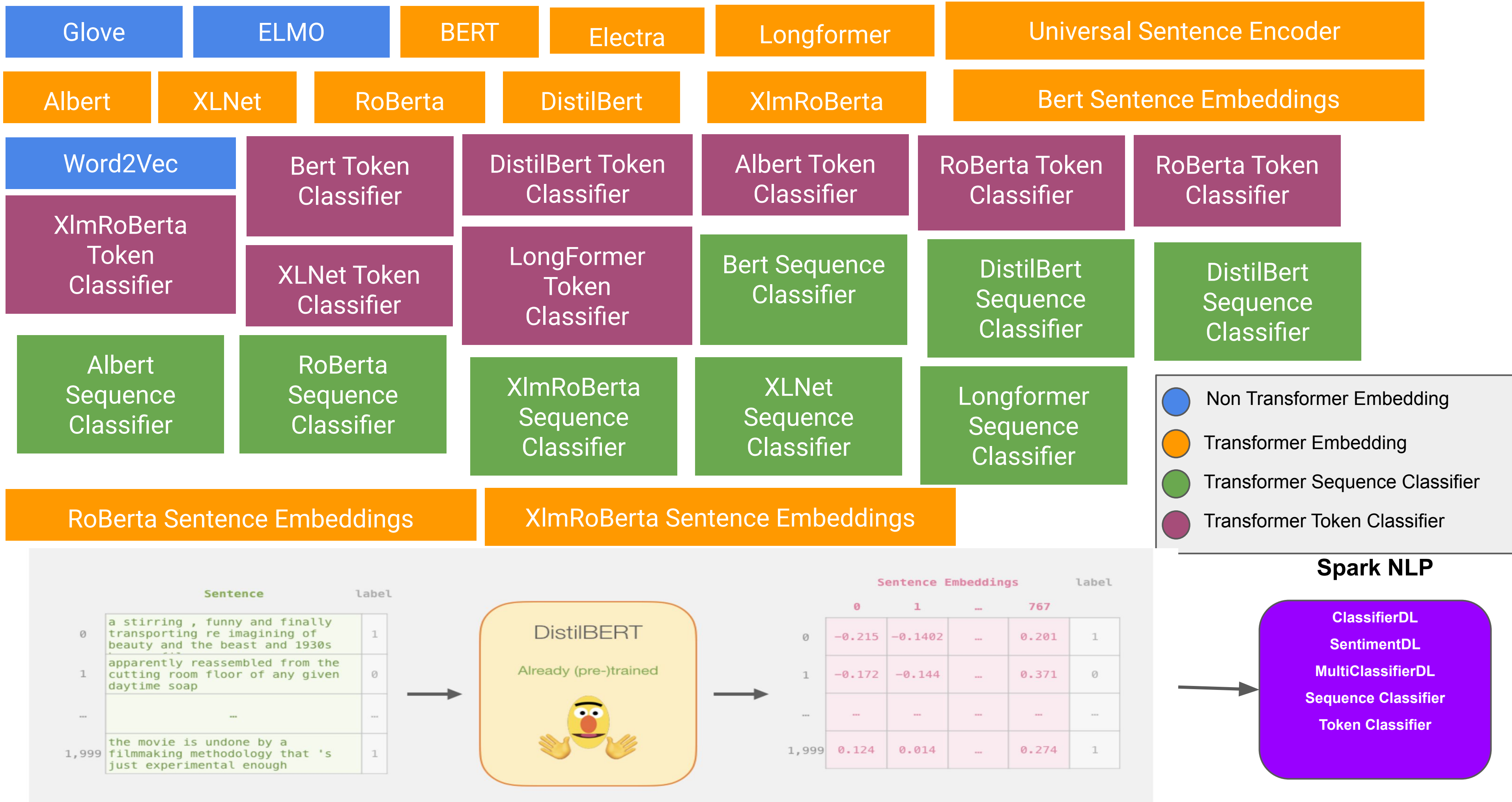


b) Similar Words have same angles

```
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 ,
               -0.3617   , -0.18355  , -0.34717  , -0.3691   ,  0.16881  ,
                0.21018  , -0.38376  , -0.096909 , -0.36296  , -0.37319  ,
                0.0021152,  0.32512  ,  0.063977 ,  0.36249  , -0.26935  ,
               -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  ,  0.44076  ,  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  ,
               -0.19228  ,  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 , -0.10066  ,
               -0.4165   , -0.20252  ,  0.2794   ,  0.10852  , -0.40154  ,
```

- Deep-Learning-based natural language processing systems.
- They 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.

Text Classification with Word & Sentence Embeddings & Transformers



Embeddings

in

Spark NLP

Part 1

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	tfhub_use	UniversalSentenceEncoder
7	en	tfhub_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_sciwiki_150d	WordEmbeddingsModel
21	es	embeddings_sciwiki_300d	WordEmbeddingsModel
22	es	embeddings_sciwiki_50d	WordEmbeddingsModel
23	en	embeddings_healthcare_100d	WordEmbeddingsModel
24	en	embeddings_biovec	WordEmbeddingsModel
25	en	bert_base_cased	BertEmbeddings

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_pmc_base_cased	BertEmbeddings
33	en	biobert_pubmed_base_cased	BertEmbeddings
34	en	biobert_pubmed_large_cased	BertEmbeddings
35	en	biobert_pubmed_pmc_base_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_pmc_base_cased	BertSentenceEmbeddings
44	en	sent_biobert_pubmed_base_cased	BertSentenceEmbeddings
45	en	sent_biobert_pubmed_large_cased	BertSentenceEmbeddings
46	en	sent_biobert_pubmed_pmc_base_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

Embeddings

in

Spark NLP

Part 2

Number	Language	Spark NLP Model Name	Model
51	en	sent_small_bert_L12_128	BertSentenceEmbeddings
52	en	sent_small_bert_L12_256	BertSentenceEmbeddings
53	en	sent_small_bert_L12_512	BertSentenceEmbeddings
54	en	sent_small_bert_L12_768	BertSentenceEmbeddings
55	en	sent_small_bert_L2_128	BertSentenceEmbeddings
56	en	sent_small_bert_L2_256	BertSentenceEmbeddings
57	en	sent_small_bert_L2_512	BertSentenceEmbeddings
58	en	sent_small_bert_L2_768	BertSentenceEmbeddings
59	en	sent_small_bert_L4_128	BertSentenceEmbeddings
60	en	sent_small_bert_L4_256	BertSentenceEmbeddings
61	en	sent_small_bert_L4_512	BertSentenceEmbeddings
62	en	sent_small_bert_L4_768	BertSentenceEmbeddings
63	en	sent_small_bert_L6_128	BertSentenceEmbeddings
64	en	sent_small_bert_L6_256	BertSentenceEmbeddings
65	en	sent_small_bert_L6_512	BertSentenceEmbeddings
66	en	sent_small_bert_L6_768	BertSentenceEmbeddings
67	en	sent_small_bert_L8_128	BertSentenceEmbeddings
68	en	sent_small_bert_L8_256	BertSentenceEmbeddings
69	en	sent_small_bert_L8_512	BertSentenceEmbeddings
70	en	sent_small_bert_L8_768	BertSentenceEmbeddings
71	en	small_bert_L10_128	BertEmbeddings
72	en	small_bert_L10_256	BertEmbeddings
73	en	small_bert_L10_512	BertEmbeddings
74	en	small_bert_L10_768	BertEmbeddings
75	en	small_bert_L12_128	BertEmbeddings

Number	Language	Spark NLP Model Name	Model
76	en	small_bert_L12_256	BertEmbeddings
77	en	small_bert_L12_512	BertEmbeddings
78	en	small_bert_L12_768	BertEmbeddings
79	en	small_bert_L2_128	BertEmbeddings
80	en	small_bert_L2_256	BertEmbeddings
81	en	small_bert_L2_512	BertEmbeddings
82	en	small_bert_L2_768	BertEmbeddings
83	en	small_bert_L4_128	BertEmbeddings
84	en	small_bert_L4_256	BertEmbeddings
85	en	small_bert_L4_512	BertEmbeddings
86	en	small_bert_L4_768	BertEmbeddings
87	en	small_bert_L6_128	BertEmbeddings
88	en	small_bert_L6_256	BertEmbeddings
89	en	small_bert_L6_512	BertEmbeddings
90	en	small_bert_L6_768	BertEmbeddings
91	en	small_bert_L8_128	BertEmbeddings
92	en	small_bert_L8_256	BertEmbeddings
93	en	small_bert_L8_512	BertEmbeddings
94	en	small_bert_L8_768	BertEmbeddings
95	en	covidbert_large_uncased	BertEmbeddings
96	en	electra_base_uncased	BertEmbeddings
97	en	electra_large_uncased	BertEmbeddings
98	en	electra_small_uncased	BertEmbeddings
99	en	sent_covidbert_large_uncased	BertSentenceEmbeddings
100	en	sent_electra_base_uncased	BertSentenceEmbeddings

Embeddings in Spark NLP Part 3

Number	Language	Spark NLP Model Name	Model
101	en	sent_electra_large_uncased	BertSentenceEmbeddings
102	en	sent_electra_small_uncased	BertSentenceEmbeddings
103	fi	bert_finnish_cased	BertEmbeddings
104	fi	bert_finnish_uncased	BertEmbeddings
105	de	w2v_cc_300d	WordEmbeddingsModel
106	en	biobert_clinical_base_cased	BertEmbeddings
107	en	biobert_discharge_base_cased	BertEmbeddings
108	en	biobert_pmc_base_cased	BertEmbeddings
109	en	biobert_pubmed_base_cased	BertEmbeddings
110	en	biobert_pubmed_large_cased	BertEmbeddings
111	en	biobert_pubmed_pmc_base_cased	BertEmbeddings
112	en	sent_biobert_clinical_base_cased	BertSentenceEmbeddings
113	en	sent_biobert_discharge_base_cased	BertSentenceEmbeddings
114	en	sent_biobert_pmc_base_cased	BertSentenceEmbeddings
115	en	sent_biobert_pubmed_base_cased	BertSentenceEmbeddings
116	en	sent_biobert_pubmed_large_cased	BertSentenceEmbeddings
117	en	sent_biobert_pubmed_pmc_base_cased	BertSentenceEmbeddings
118	en	labse	BertSentenceEmbeddings
119	pt	bert_portuguese_base_cased	BertEmbeddings
120	pt	bert_portuguese_large_cased	BertEmbeddings
121	en	sbiobert_base_cased_mli	BertSentenceEmbeddings
122	en	sbluebert_base_uncased_mli	BertSentenceEmbeddings
123	ur	urduvec_140M_300d	WordEmbeddingsModel
124	ar	arabic_w2v_cc_300d	WordEmbeddingsModel
125	fa	persian_w2v_cc_300d	WordEmbeddingsModel

Number	Language	Spark NLP Model Name	Model
126	xx	tfhub_use_multi_lg	UniversalSentenceEncoder
127	xx	tfhub_use_multi	UniversalSentenceEncoder
128	he	hebrew_cc_300d	WordEmbeddingsModel
129	hi	hindi_cc_300d	WordEmbeddingsModel
130	bn	bengali_cc_300d	WordEmbeddingsModel
131	xx	tfhub_use_multi_lg	UniversalSentenceEncoder
132	xx	tfhub_use_multi	UniversalSentenceEncoder
133	en	sbert_jsl_medium_umls_uncased	BertSentenceEmbeddings
134	en	sbert_jsl_medium_uncased	BertSentenceEmbeddings
135	en	sbert_jsl_mini_umls_uncased	BertSentenceEmbeddings
136	en	sbert_jsl_mini_uncased	BertSentenceEmbeddings
137	en	sbert_jsl_tiny_umls_uncased	BertSentenceEmbeddings
138	en	sbert_jsl_tiny_uncased	BertSentenceEmbeddings
139	en	sbiobert_jsl_cased	BertSentenceEmbeddings
140	en	sbiobert_jsl_umls_cased	BertSentenceEmbeddings
141	zh	bert_base_chinese	BertEmbeddings
142	nl	bert_base_dutch_cased	BertEmbeddings
143	de	bert_base_german_cased	BertEmbeddings
144	de	bert_base_german_uncased	BertEmbeddings
145	it	bert_base_italian_cased	BertEmbeddings
146	it	bert_base_italian_uncased	BertEmbeddings
147	xx	bert_base_multilingual_cased	BertEmbeddings
148	xx	bert_base_multilingual_uncased	BertEmbeddings
149	tr	bert_base_turkish_cased	BertEmbeddings
150	tr	bert_base_turkish_uncased	BertEmbeddings

Embeddings in Spark NLP Part 4

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

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

Embeddings

in

Spark NLP

Part 5

Number	Language	Spark NLP Model Name	Model
201	xx	sent_xlm_roberta_base	XlmRoBertaForTokenClassification
202	es	sent_bert_base_cased	BertSentenceEmbeddings
203	nl	sent_bert_base_cased	BertSentenceEmbeddings
204	sv	sent_bert_base_cased	BertSentenceEmbeddings
205	el	sent_bert_base_uncased	BertSentenceEmbeddings
206	es	sent_bert_base_uncased	BertSentenceEmbeddings
207	en	sent_bert_base_uncased_legal	BertSentenceEmbeddings
208	es	bert_base_cased	BertEmbeddings
209	nl	bert_base_cased	BertEmbeddings
210	sv	bert_base_cased	BertEmbeddings
211	el	bert_base_uncased	BertEmbeddings
212	es	bert_base_uncased	BertEmbeddings
213	en	bert_base_uncased_legal	BertEmbeddings
214	ja	japanese_cc_300d	WordEmbeddingsModel
215	ro	bert_base_cased	BertEmbeddings
216	de	sent_bert_base_cased	BertSentenceEmbeddings
217	am	xlm_roberta_base_finetuned_amharic	XlmRoBertaEmbeddings
218	ha	xlm_roberta_base_finetuned_hausa	XlmRoBertaEmbeddings
219	ig	xlm_roberta_base_finetuned_igbo	XlmRoBertaEmbeddings
220	rw	xlm_roberta_base_finetuned_kinyarwanda	XlmRoBertaEmbeddings
221	lg	xlm_roberta_base_finetuned_luganda	XlmRoBertaEmbeddings
222	xx	xlm_roberta_large	XlmRoBertaEmbeddings
223	am	sent_xlm_roberta_base_finetuned_amharic	XlmRoBertaSentenceEmbeddings
224	ha	sent_xlm_roberta_base_finetuned_hausa	XlmRoBertaSentenceEmbeddings
225	ig	sent_xlm_roberta_base_finetuned_igbo	XlmRoBertaSentenceEmbeddings

Number	Language	Spark NLP Model Name	Model
226	rw	sent_xlm_roberta_base_finetuned_kinyarwanda	XlmRoBertaSentenceEmbeddings
227	lg	sent_xlm_roberta_base_finetuned_luganda	XlmRoBertaSentenceEmbeddings
228	pcm	sent_xlm_roberta_base_finetuned_naija	XlmRoBertaSentenceEmbeddings
229	sw	sent_xlm_roberta_base_finetuned_swahili	XlmRoBertaSentenceEmbeddings
230	wo	sent_xlm_roberta_base_finetuned_wolof	XlmRoBertaSentenceEmbeddings
231	yo	sent_xlm_roberta_base_finetuned_yoruba	XlmRoBertaSentenceEmbeddings
232	lou	xlm_roberta_base_finetuned_luo	XlmRoBertaEmbeddings
233	pcm	xlm_roberta_base_finetuned_naija	XlmRoBertaEmbeddings
234	sw	xlm_roberta_base_finetuned_swahili	XlmRoBertaEmbeddings
235	wo	xlm_roberta_base_finetuned_wolof	XlmRoBertaEmbeddings
236	yo	xlm_roberta_base_finetuned_yoruba	XlmRoBertaEmbeddings
237	es	roberta_base_biomedical	RoBertaEmbeddings
238	en	doc2vec_gigaword_300	Doc2VecModel
239	en	doc2vec_gigaword_wiki_300	Doc2VecModel
240	te	distilbert_uncased	DistilBertEmbeddings
241	en	sbert_jsl_medium_rxnorm_uncased	BertSentenceEmbeddings
242	fi	bert_base_finnish_cased	BertSentenceEmbeddings
243	fi	bert_base_finnish_uncased	BertSentenceEmbeddings
244	en	sbert_jsl_medium_rxnorm_uncased	BertSentenceEmbeddings
245	en	word2vec_gigaword_300	Word2VecModel
246	en	word2vec_gigaword_wiki_300	Word2VecModel
247	en	electra_medal_acronym	BertEmbeddings
248	vi	distilbert_base_cased	DistilBertEmbeddings

Transformer **Sequence** Classifiers in Spark NLP

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_finbert	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

Transformer **Token** Classifiers in Spark NLP Part 1

Number	Language	Spark NLP Model Name	Model
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_peymaner	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

Number	Language	Spark NLP Model Name	Model
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

Transformer **Token** Classifiers in Spark NLP Part 2

Number	Language	Spark NLP Model Name	Model
51	es	roberta_token_classifier_bne_capitel_ner	RoBertaForTokenClassification
52	nl	bert_token_classifier_dutch_udlassy_ner	BertForTokenClassification
53	xx	bert_token_classifier_scandi_ner	BertForTokenClassification
54	en	bert_token_classifier_drug_development_trials	BertForTokenClassification
55	de	xlm_roberta_large_token_classifier_conll03	XlmRoBertaForTokenClassification
56	xx	xlm_roberta_large_token_classifier_hrl	XlmRoBertaForTokenClassification
57	hi	bert_token_classifier_hi_en_ner	BertForTokenClassification
58	id	roberta_token_classifier_pos_tagger	RoBertaForTokenClassification
59	en	roberta_token_classifier_ticker	RoBertaForTokenClassification
60	en	roberta_token_classifier_timex_semeval	RoBertaForTokenClassification
61	en	bert_token_classifier_ner_bionlp	MedicalBertForTokenClassifier
62	en	bert_token_classifier_ner_ade	MedicalBertForTokenClassifier
63	en	bert_token_classifier_ner_anatomy	MedicalBertForTokenClassifier
64	en	bert_token_classifier_ner_cellular	MedicalBertForTokenClassifier
65	en	bert_token_classifier_ner_chemicals	MedicalBertForTokenClassifier
66	en	bert_token_classifier_ner_chemprot	MedicalBertForTokenClassifier
67	en	bert_token_classifier_ner_clinical	MedicalBertForTokenClassifier
68	en	bert_token_classifier_ner_deid	MedicalBertForTokenClassifier
69	en	bert_token_classifier_ner_drugs	MedicalBertForTokenClassifier
70	en	bert_token_classifier_ner_jsl	MedicalBertForTokenClassifier
71	en	bert_token_classifier_ner_jsl_slim	MedicalBertForTokenClassifier
72	en	bert_token_classifier_ner_bacteria	MedicalBertForTokenClassifier

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.

Transformers & Embeddings

- **DOC2Vec :** https://arxiv.org/abs/1301.3781
- **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/**

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	PORTUGUESE
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	jv	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	GUJARATI	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	PUNJABI		
hr	CROATIAN	pl	POLISH		

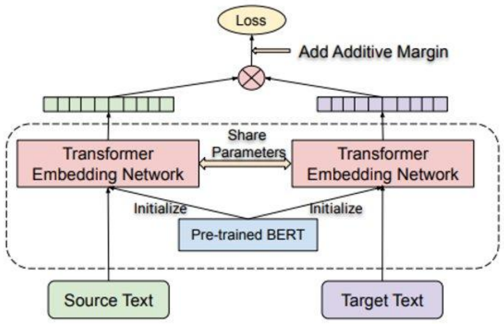


Figure 1: Dual encoder model with BERT based encoding modules.

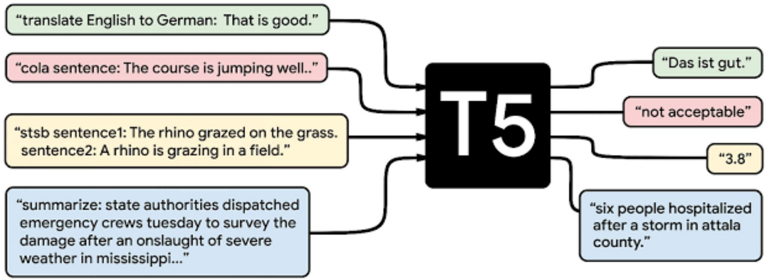
Translate between **200+ Languages**

With Marian: Fast Neural Machine Translation in C++

MARIANNMT

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')`
```




```python
# 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 deducted 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 disambigous 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

TensorFlow Hub

TF Hub to Spark NLP

Spark NLP	TF Hub Notebooks	Colab
BertEmbeddings	TF Hub in Spark NLP - BERT	Open in Colab
BertSentenceEmbeddings	TF Hub in Spark NLP - BERT Sentence	Open in Colab
AlbertEmbeddings	TF Hub in Spark NLP - ALBERT	Open in Colab

HuggingFace to Spark NLP

Spark NLP	HuggingFace Notebooks	Colab
BertEmbeddings	HuggingFace in Spark NLP - BERT	Open in Colab
BertSentenceEmbeddings	HuggingFace in Spark NLP - BERT Sentence	Open in Colab
DistilBertEmbeddings	HuggingFace in Spark NLP - DistilBERT	Open in Colab
RoBertaEmbeddings	HuggingFace in Spark NLP - RoBERTa	Open in Colab
XlmRoBertaEmbeddings	HuggingFace in Spark NLP - XLM-RoBERTa	Open in Colab
AlbertEmbeddings	HuggingFace in Spark NLP - ALBERT	Open in Colab
XlnetEmbeddings	HuggingFace in Spark NLP - XLNet	Open in Colab
LongformerEmbeddings	HuggingFace in Spark NLP - Longformer	Open in Colab
BertForTokenClassification	HuggingFace in Spark NLP - BertForTokenClassification	Open in Colab
DistilBertForTokenClassification	HuggingFace in Spark NLP - DistilBertForTokenClassification	Open in Colab
AlbertForTokenClassification	HuggingFace in Spark NLP - AlbertForTokenClassification	Open in Colab
RoBertaForTokenClassification	HuggingFace in Spark NLP - RoBertaForTokenClassification	Open in Colab
XlmRoBertaForTokenClassification	HuggingFace in Spark NLP - XlmRoBertaForTokenClassification	Open in Colab
BertForSequenceClassification	HuggingFace in Spark NLP - BertForSequenceClassification	Open in Colab
DistilBertForSequenceClassification	HuggingFace in Spark NLP - DistilBertForSequenceClassification	Open in Colab
AlbertForSequenceClassification	HuggingFace in Spark NLP - AlbertForSequenceClassification	Open in Colab
RoBertaForSequenceClassification	HuggingFace in Spark NLP - RoBertaForSequenceClassification	Open in Colab
XlmRoBertaForSequenceClassification	HuggingFace in Spark NLP - XlmRoBertaForSequenceClassification	Open in Colab
XlnetForSequenceClassification	HuggingFace in Spark NLP - XlnetForSequenceClassification	Open in Colab

https://nlp.johnsnowlabs.com/docs/en/transformers#example-notebooks

Part - III (Day 1) - Coding Time

- Notebook 3 Spark NLP pretrained models

Spark NLP for Data Scientists

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

Part - IV (Day 1)

- Training a Named Entity Recognizer
- Training a Deep Learning classifier
- Upload/Download your own model via John Snow Labs Models Hub
- Notebook 4 NERDL Training
- Notebook 5 ClassifierDL, SentimentDL, MultiClassifierDL Training

Spark NLP for Data Scientists

John Snow LABS

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

Not Secure | nlpprogress.com/eng...

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 (Ghaddar 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

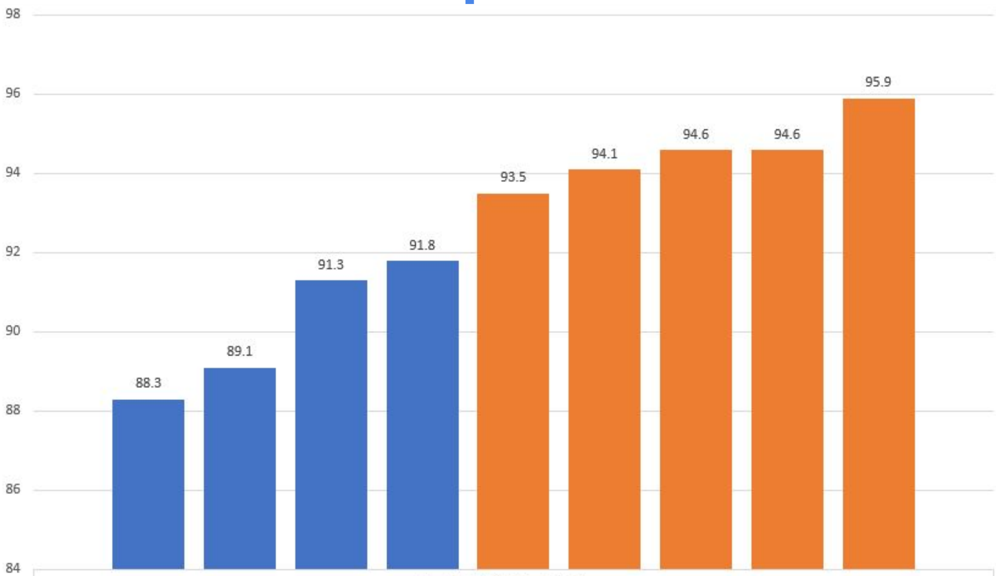
93.3 %
Test Set

Bert

NerDLApproach

John Snow LABS

NER-DL in Spark NLP



	CoNLL 2003 - F1 Dev
spaCy 2.2	88.3
spaCy 2.2 + en_trf_bertbaseuncased_lg	89.1
spaCy 2.2 + en_core_web_md	91.3
spaCy 2.2 + en_core_web_lg	91.8
Spark NLP + GloVe 100d	93.5
Spark NLP + GloVe 300d	94.1
Spark NLP + ELMO	94.6
Spark NLP + BERT Base Cased	94.6
Spark NLP + BERT Large Cased	95.9

NER Systems

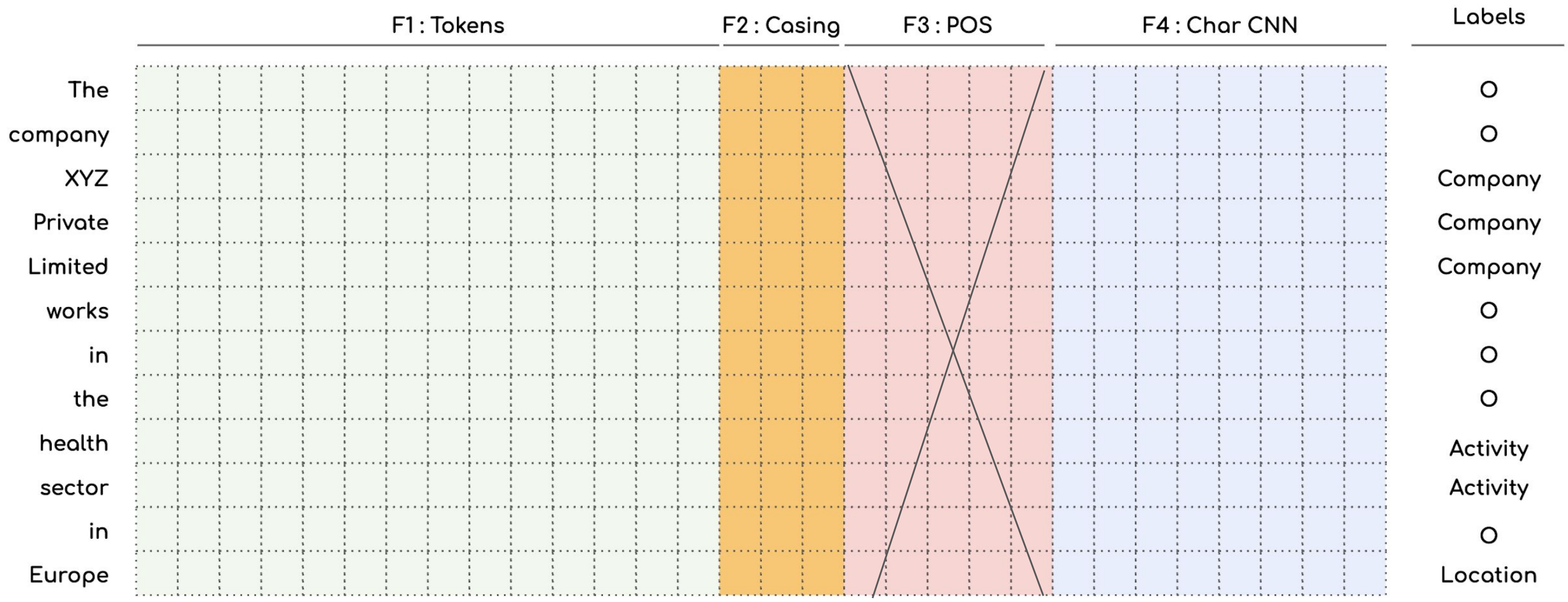
Feature-engineered machine learning systems	Dict	SP	DU	EN	GE
Carreras et al. (2002) binary AdaBoost classifiers	Yes	81.39	77.05	-	-
Malouf (2002) - Maximum Entropy (ME) + features	Yes	73.66	68.08	-	-
Li et al. (2005) SVM with class weights	Yes	-	-	88.3	-
Passos et al. (2014) CRF	Yes	-	-	90.90	-
Ando and Zhang (2005a) Semi-supervised state of the art	No	-	-	89.31	75.27
Agerri and Rigau (2016)	Yes	**84.16**	**85.04**	**91.36**	**76.42**
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	**76.12**
Collobert et al. (2011) Conv-CRF (SENNA+Gazetteer)	Yes	-	-	89.59	-
Huang et al. (2015) Bi-LSTM+CRF+ (SENNA+Gazetteer)	Yes	-	-	**90.10**	-
Feature-inferring neural network character models					
Gillick et al. (2015) – BTS	No	**82.95**	**82.84**	**86.50**	**76.22**
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	**85.19**	91.26	-
Luo (2015)	Yes	-	-	91.20	-
Chiu and Nichols (2015)	Yes	-	-	**91.62**	-
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	**78.76**
Bharadwaj et al. (2016)	Yes	**85.81**	-	-	-
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	**87.26**	**87.54**	90.86	**79.01**

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)

John Snow LABS

NER-DL in Spark NLP

Char-CNN-BiLSTM



NER-DL in Spark NLP

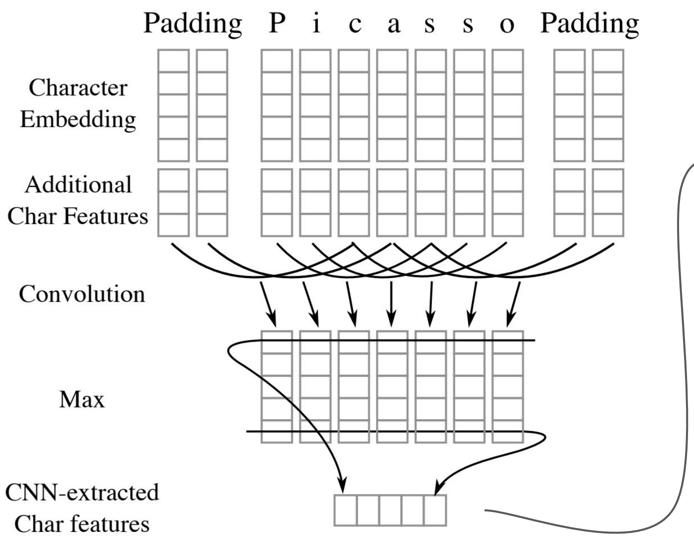


Figure 2: The convolutional neural network extracts character features from each word. The character embedding and (optionally) the character type feature vector are computed through lookup tables. Then, they are concatenated and passed into the CNN.

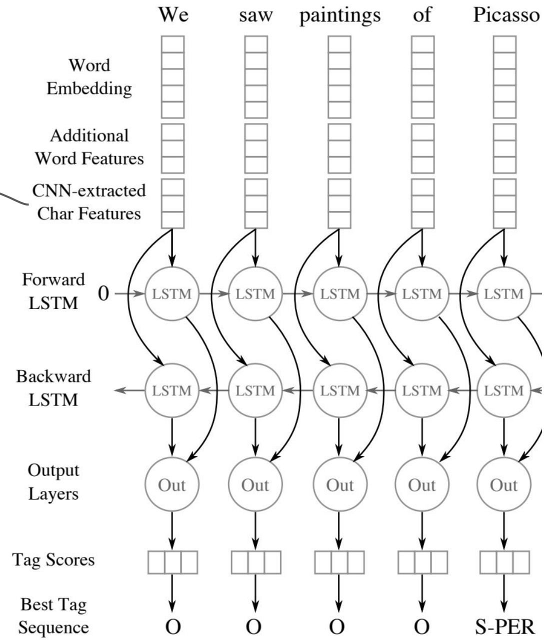


Figure 1: The (unrolled) BLSTM for tagging named entities. Multiple tables look up word-level feature vectors. The CNN (Figure 2) extracts a fixed length feature vector from character-level features. For each word, these vectors are concatenated and fed to the BLSTM network and then to the output layers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

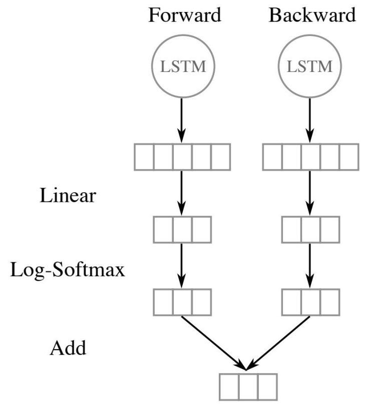


Figure 3: The output layers ("Out" in Figure 1) decode output into a score for each tag category.

Char-CNN-BiLSTM

John Snow LABS

NER-DL in Spark NLP

CoNNL2003 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

SentimentDL, ClassifierDL, and MultiClassifierDL

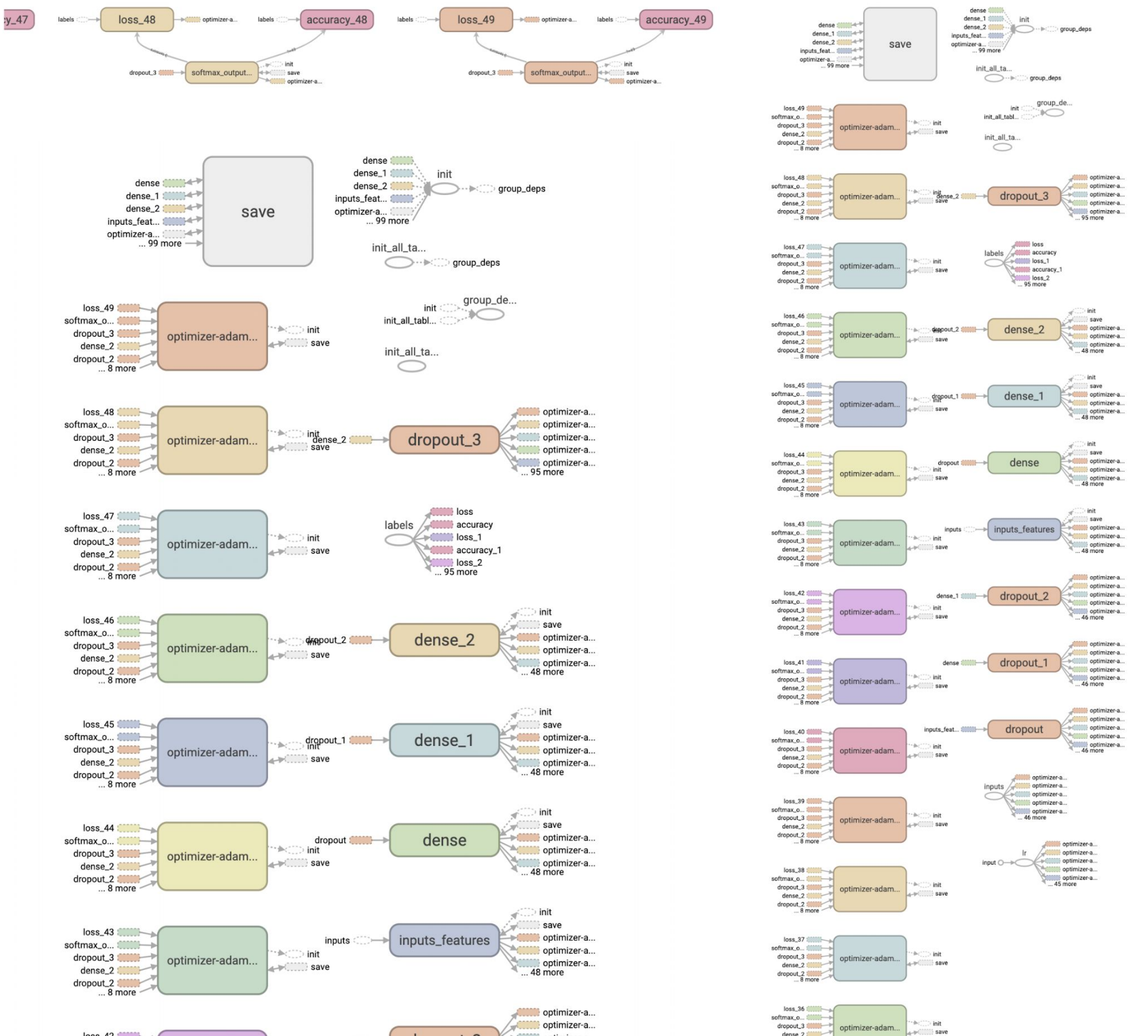
- BERT
- Small BERT
- BioBERT
- CovidBERT
- LaBSE
- ALBERT
- ELECTRA
- XLNet
- ELMO
- Universal Sentence Encoder
- GloVe

- 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

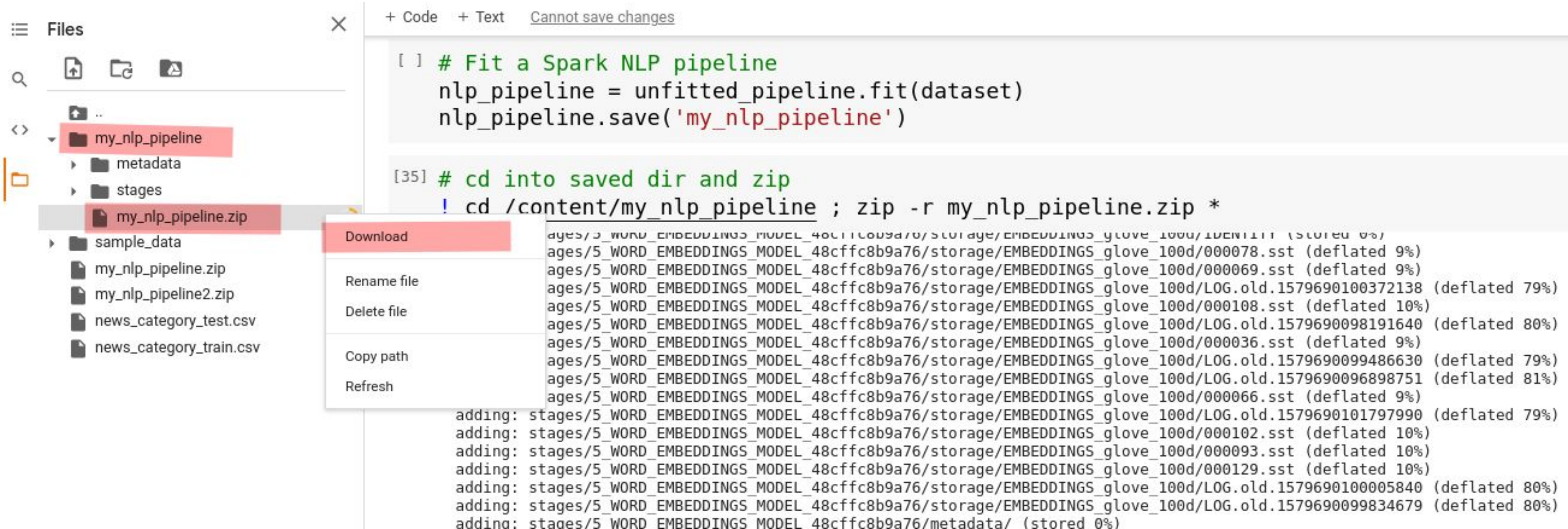
- tfhub_ues
- tfhub_use_lg
- glove_6B_100
- glove_6B_300
- glove_840B_300
- bert_base_cased
- bert_base_uncased
- bert_large_cased
- bert_large_uncased
- bert_multi_uncased
- electra_small_uncased
- elmo
- … 100+ Word & Sentence models

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



Part - IV (Day 1) - Coding Time

- Notebook 4 NERDL Training
- Notebook 5 ClassifierDL, SentimentDL, MultiClassifierDL Training

Spark NLP for Data Scientists

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

End of Day 1
See you tomorrow same time same place :)

Spark NLP
for Data Scientists

John Snow LABS

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

Welcome - We have a lot of things ahead of us

Day-2	50 min	- Unsupervised Keyword Extraction with YAKE - Sentence Detection - Spell Checking - Graph triplet extraction
	10 min	Break
	60 min	- Sequence Classification with Transformers - Token Classification with Transformers - GPT2 - Conditional Text generation - Text Style Transfer and SQL code generation from Natural Language Text with T5
	10 min	Break
	60 min	- Question Answering, Summarization and other T5 applications - Multilingual NLP - Train only on English data and predict for 100+ languages - Huggingface and Tensorflow Hub to Spark NLP export
	10 min	Break
	50 min	- The Python NLU library basics, use any of the 4000+ models in 1 line of code - NLU & Streamlit - NLP Server - NLU OCR

Part - I (Day 2)

- Notebook 7 Context Spell Checker
- Notebook 8 YAKE
- Notebook 9 Sentence Detection
- Notebook 12 RDF Graph Extraction

Spark NLP
for Data Scientists

John Snow LABS

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

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']
```

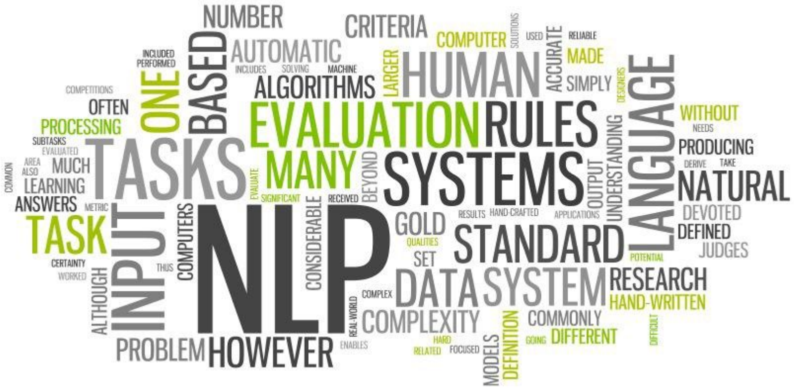
Notice that in the first example, 'siter' is indeed a valid English word,
https://www.merriam-webster.com/dictionary/siter

Notebook 7_ Spell Checker

Unsupervised Keyword Extraction

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

[Notebook 8 YAKE]()



Contextual Aware DL Sentence Detector

9 SentenceDetectorDL

`SentenceDetectorDL` (SDDL) is based on a general-purpose neural network model for sentence boundary detection. The task of sentence boundary detection is to identify sentences within a text. Many natural language processing tasks take a sentence as an input unit, such as part-of-speech tagging, dependency parsing, named entity recognition or machine translation.

In this model, we treated the sentence boundary detection task as a classification problem using a DL CNN architecture. We also modified the original implemenation a little bit to cover broken sentences and some impossible end of line chars.

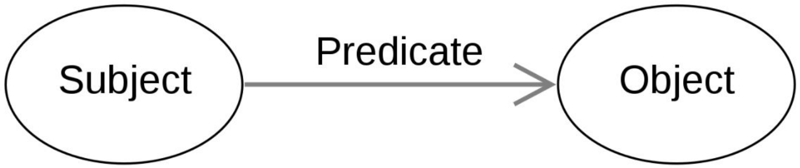
We are releasing two pretrained SDDL models: `english` and `multilanguage` that are trained on `SETimes corpus (Tyers and Alperen, 2010)` and `Europarl. Wong et al. (2014)` datasets.

Here are the test metrics on various languages for `multilang` model

[Notebook 9 Sentence Detector DL]

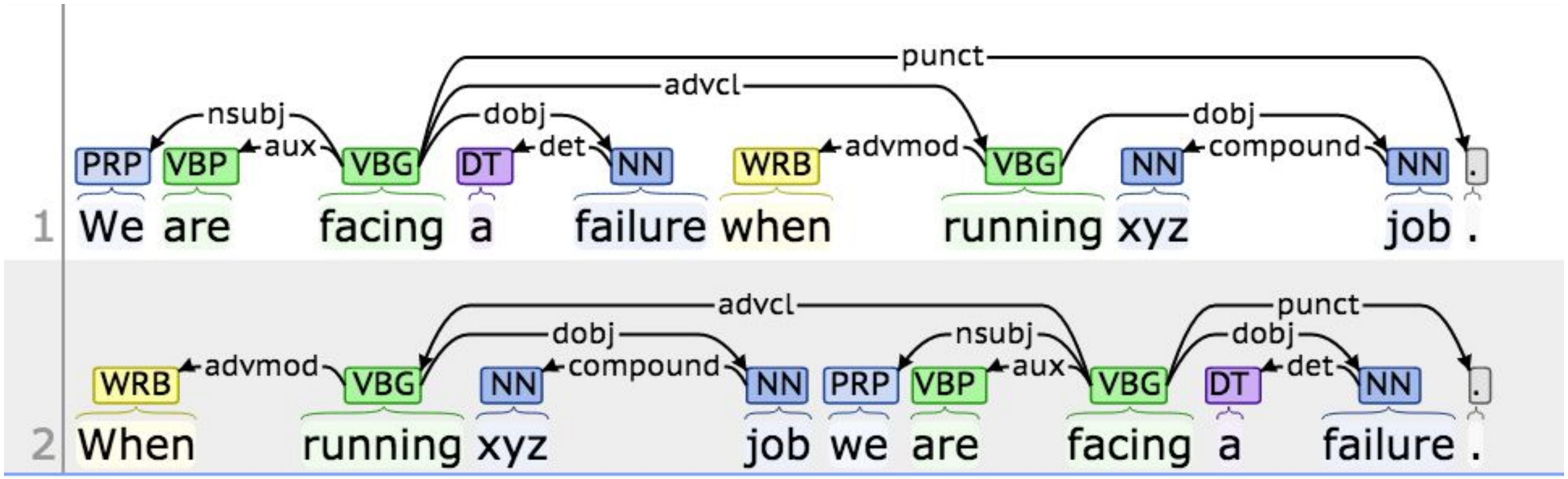
Language	Accuracy	Recall	Precision	F1
bg	0.96	1.00	0.93	0.96
bs	0.98	1.00	0.96	0.98
de	0.97	0.99	0.94	0.97
el	0.97	1.00	0.94	0.97
en	0.98	1.00	0.96	0.98
hr	0.98	1.00	0.96	0.98
mk	0.96	0.99	0.93	0.96
ro	0.97	1.00	0.95	0.97
sq	0.98	1.00	0.96	0.98
sr	0.98	1.00	0.95	0.97
tr	0.98	0.99	0.96	0.98

Extract RDF Semantic Triplets



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

[Notebook 12 Graph Extraction]



Part - 1 (Day 2) - Coding Time

- Notebook 7 Context Spell Checker
- Notebook 8 YAKE
- Notebook 9 Sentence Detection
- Notebook 12 RDF Graph Extraction

Spark NLP for Data Scientists

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

Part - II (Day 2)

- Sequence Classification with Transformer Models
- Token Classification with Transformer Models
- GPT2 - Conditional Text Generation
- Generate SQL from natural language text and Text Style Transfer with T5
- Notebook 5.3 Transformers for Sequence Classification
- Notebook 14 Transformers for Token Classification
- Notebook 16 Text Generation with GPT2
- Notebook 10.2 - SQL generation and Style Transfer with T5

Spark NLP for Data Scientists

John Snow LABS

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

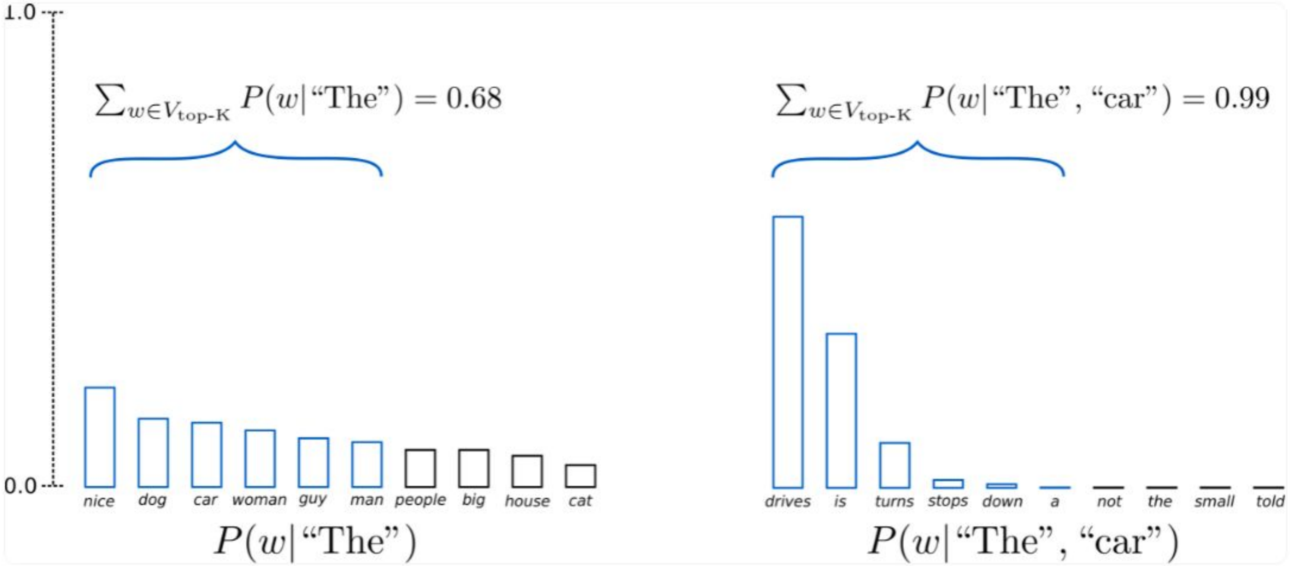
GPT2 - Conditional Text Generation

- **GPT2** is very capable of generating text, but introduces new engineering challenges, so called **Prompt Engineering**
- The outputs of GPT2 depend on the text sequence we feed it in the beginning, the so called "**Prompt". Choosing the right prompt for your problem is the biggest challenge
- **Tunable Parameters:**
  - **Min Output Length** : Minimum length of generated sequence
  - **Max Output Length** : Maximum length of generated sequence
  - **Repetition Penalty:** How much to penalize new tokens, based on their existing frequency in the text so far. Decreases the model's likelihood to repeat the same line verbatim
  - **Temperature** : Controls **Randomness**, lower values result in less random text generation. As the temperature approaches zero, the model will become deterministic and repetitive.
  - **Top P** : Controls diversity via nucleus sampling : 0.5 means half of all likelihood - weighted options are considered
  - Do Sampling
  - **Top K**: The number of highest probability vocabulary tokens to keep for top-k-filtering
  - **Number of Repeated N Grams :** How often a N Gram may be repeated. Setting it to N, all N Grams of size N can only occur once.

[Open AI GPT2]

Top-K Sampling

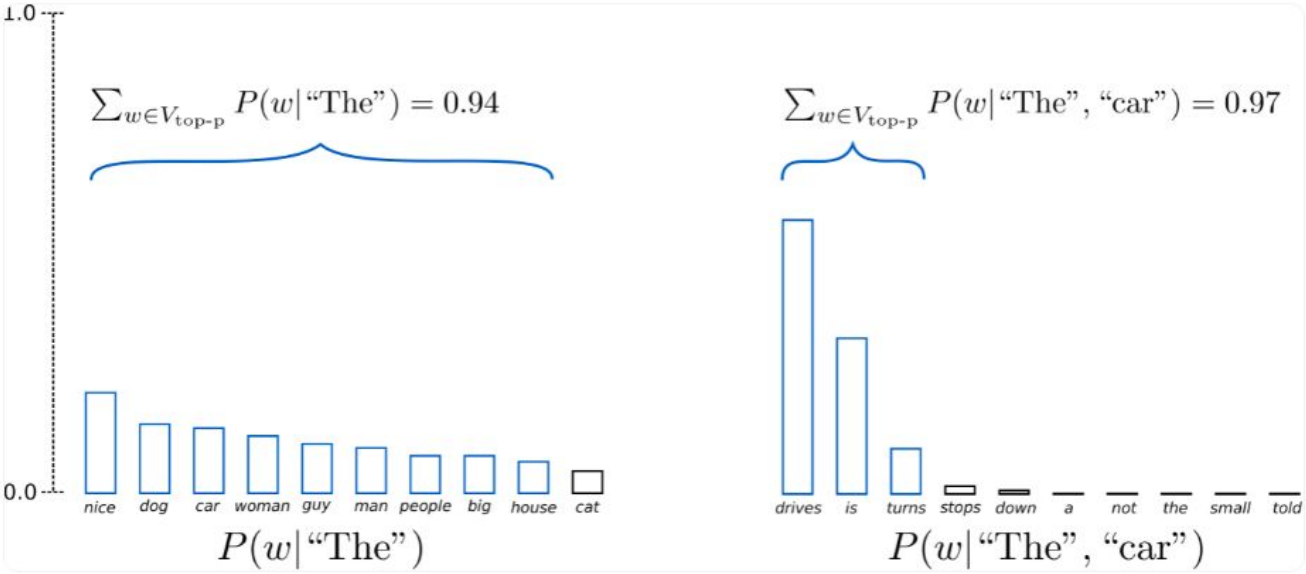
- With Top-K sampling, only the first K word candidates are used to find the next generated word candidate
- These words are filtered and the probability mass is redistributed among those K words



[Open AI GPT2]

Top-P Sampling

- With Top-P sampling, the model will choose next word candidates, by finding the smallest possible subset of words, whose cumulative probability exceed the probability **p**
- The probability mass is redistributed among this subset of words



[Open AI GPT2]

GPT2 - Good vs Bad Prompts

37]
```
# Bad prompting, the input text we condition GPT2 yields bad output, it does not understand the pattern we want from the original input
gpt2_pipe.predict("Suggest me a good Sci-Fi movie")
```

1 entry Filter ?

index	document	generated
0	Suggest me a good Sci-Fi movie	generate Suggest me a good Sci-Fi movie. I'm not sure if I'm going to be able to do this, but I'm sure I'll be able. (I'm sure you're going to want to do it.) So, I'm not going to do that. . , .

Show 25 per page
Like what you see? Visit the [data table notebook]() to learn more about interactive tables.

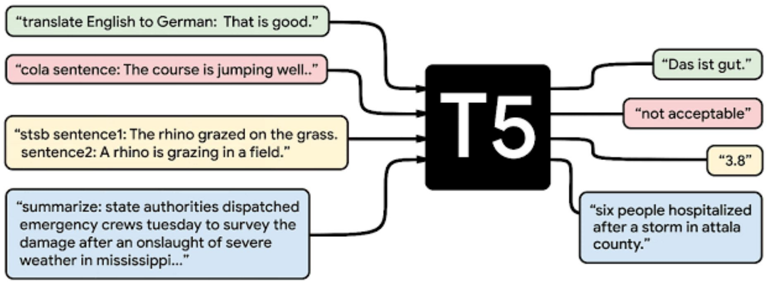
```
# Good prompting. help GPT2 out and by giving it a few samples in the prompt we condition it on
gpt2_pipe.predict("""Generate a top 10 movie list: \n
1.The Matrix \n
2.Terminator \n
3. """)
```

1 entry Filter ?

index	document	generated
0	Generate a top 10 movie list: 1.The Matrix 2.Terminator 3.	generate Generate a top 10 movie list: 1.The Matrix 2.Terminator 3.The Hunger Games 4.The Dark Knight Rises 5.The Lord of the Rings 6.The Hobbit 7.The Last Crusade 8.The Lion King 9.The Lego Movie 10.The LEGO Movie 11.The King of the Hill 12.The Jungle Book 13.The Legend of Zelda 14.The Little Mermaid 15.The Princess Bride 16.The Pirates of the Caribbean 17.The Twilight Saga 18

Show 25 per page
Like what you see? Visit the [data table notebook]() to learn more about interactive tables.

[Open AI GPT2]()




```python
# 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 deducted 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 disambigous 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

Transformer Based Token and Sequence Classifcation

```
37] # Bad prompting, the input text we condition GPT2 yields bad output, it does not understand the pattern we want from the original input
gpt2_pipe.predict("Suggest me a good Sci-Fi movie")
```

1 entry Filter

index	document	generated
0	Suggest me a good Sci-Fi movie	generate Suggest me a good Sci-Fi movie. I'm not sure if I'm going to be able to do this, but I'm sure I'll be able. (I'm sure you're going to want to do it.) So, I'm not going to do that. . , .

Show 25 per page

Like what you see? Visit the [data table notebook]() to learn more about interactive tables.

```
# Good prompting. help GPT2 out and by giving it a few samples in the prompt we condition it on
gpt2_pipe.predict("""Generate a top 10 movie list: \n
1.The Matrix \n
2.Terminator \n
3. """)
```

1 entry Filter

index	document	generated
0	Generate a top 10 movie list: 1.The Matrix 2.Terminator 3.	generate Generate a top 10 movie list: 1.The Matrix 2.Terminator 3.The Hunger Games 4.The Dark Knight Rises 5.The Lord of the Rings 6.The Hobbit 7.The Last Crusade 8.The Lion King 9.The Lego Movie 10.The LEGO Movie 11.The King of the Hill 12.The Jungle Book 13.The Legend of Zelda 14.The Little Mermaid 15.The Princess Bride 16.The Pirates of the Caribbean 17.The Twilight Saga 18

Show 25 per page

Like what you see? Visit the [data table notebook]() to learn more about interactive tables.

[Open AI GPT2]()

Part - II (Day 2) - Coding Time

- Notebook 5.3 Transformers for Sequence Classification
- Notebook 14 Transformers for Token Classification
- Notebook 16 Text Generation with GPT2
- Notebook 10.2 - SQL generation and Style Transfer with T5

Spark NLP for Data Scientists

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

Part - III (Day 2)

- Question Answering, Summarization and more with T5
- Train a Multi-Lingual Classifier for over 100 languages
- Import embedding from Huggingface and TF-Hub to Spark NLP
- Notebook 5.2 Training Multilingual Classifier
- Notebook 10 Question Answering and Summarization with T5
- Notebook 15 Import Transformers from Huggingface

Spark NLP for Data Scientists

John Snow LABS

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

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	PORTUGUESE
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	jv	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	GUJARATI	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	PUNJABI		
hr	CROATIAN	pl	POLISH		

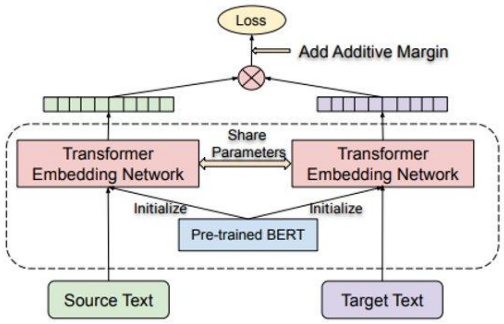
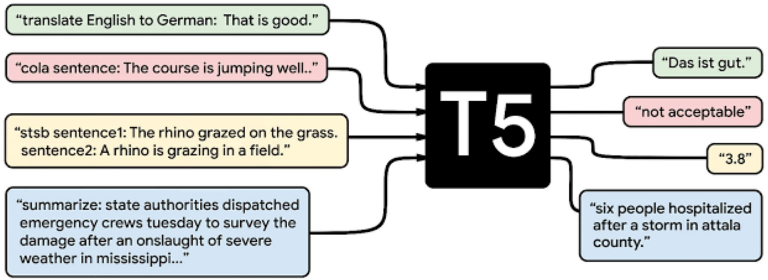


Figure 1: Dual encoder model with BERT based encoding modules.




```python
# 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 deducted 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 disambigous 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

HuggingFace to Spark NLP

Spark NLP	HuggingFace Notebooks	Colab
BertEmbeddings	HuggingFace in Spark NLP - BERT	Open in Colab
BertSentenceEmbeddings	HuggingFace in Spark NLP - BERT Sentence	Open in Colab
DistilBertEmbeddings	HuggingFace in Spark NLP - DistilBERT	Open in Colab
RoBertaEmbeddings	HuggingFace in Spark NLP - RoBERTa	Open in Colab
XlmRoBertaEmbeddings	HuggingFace in Spark NLP - XLM-RoBERTa	Open in Colab
AlbertEmbeddings	HuggingFace in Spark NLP - ALBERT	Open in Colab
XlnetEmbeddings	HuggingFace in Spark NLP - XLNet	Open in Colab
LongformerEmbeddings	HuggingFace in Spark NLP - Longformer	Open in Colab
BertForTokenClassification	HuggingFace in Spark NLP - BertForTokenClassification	Open in Colab
DistilBertForTokenClassification	HuggingFace in Spark NLP - DistilBertForTokenClassification	Open in Colab
AlbertForTokenClassification	HuggingFace in Spark NLP - AlbertForTokenClassification	Open in Colab
RoBertaForTokenClassification	HuggingFace in Spark NLP - RoBertaForTokenClassification	Open in Colab
XlmRoBertaForTokenClassification	HuggingFace in Spark NLP - XlmRoBertaForTokenClassification	Open in Colab
BertForSequenceClassification	HuggingFace in Spark NLP - BertForSequenceClassification	Open in Colab
DistilBertForSequenceClassification	HuggingFace in Spark NLP - DistilBertForSequenceClassification	Open in Colab
AlbertForSequenceClassification	HuggingFace in Spark NLP - AlbertForSequenceClassification	Open in Colab
RoBertaForSequenceClassification	HuggingFace in Spark NLP - RoBertaForSequenceClassification	Open in Colab
XlmRoBertaForSequenceClassification	HuggingFace in Spark NLP - XlmRoBertaForSequenceClassification	Open in Colab
XlnetForSequenceClassification	HuggingFace in Spark NLP - XlnetForSequenceClassification	Open in Colab

TensorFlow Hub

TF Hub to Spark NLP

Spark NLP	TF Hub Notebooks	Colab
BertEmbeddings	TF Hub in Spark NLP - BERT	Open in Colab
BertSentenceEmbeddings	TF Hub in Spark NLP - BERT Sentence	Open in Colab
AlbertEmbeddings	TF Hub in Spark NLP - ALBERT	Open in Colab

https://nlp.johnsnowlabs.com/docs/en/transformers#example-notebooks

Part - III (Day 2) - Coding Time

- Notebook 5.2 training Multilingual Classifier
- Notebook 10 Question Answering and Summarization with T5
- Notebook 15 Import Transformers from Huggingface

Spark NLP for Data Scientists

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

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/winning-in-health-care-ai-with-small-data/#1b2fc2555664
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-in-enterprises/
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

Part - IV (Day 2)

- Introduction to the Python NLU Library
- Introduction to NLP Server
- Notebook 13 - NLU Crash course, every Spark NLP model in 1 line

Spark NLP for Data Scientists

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

What is NLU?

- All of the 4000+ 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, 1...]	[-0.003953204490244389, -1.5821468830108643, -...]	[-0.7420049905776978, -0.8647691011428833, 0.1...]	[0.04002974182367325, -0.43536433577537537, -0...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
all	[-0.2182299941778183, 0.6919900178909302, 0.70...]	[1.1014549732208252, -0.43204769492149353, -0...]	[0.31148090958595276, -1.0986182689666748, 0.3...]	[-0.8933112025260925, 0.44822725653648376, -0...]	[0.17885173857212067, 0.045830272138118744, -0...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
of	[-0.15289995446205014, -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, -0.009035333991050072, -0...]	[-0.11999595910310745, 0.012994140386581421, -...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
once	[-0.23837999999523163, 0.221670001745224, 0.35...]	[-0.39100387692451477, -0.8297092914581299, 2...]	[-0.46001458168029785, -1.2062749862670898, 0...]	[0.2988640069961548, 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, -0.5257778167724609, -0.4...]	[-1.3351283073425293, 0.6296550035476685, -1.4...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
Watch	[-0.38264000415802, -0.08968199789524078, 0.02...]	[-0.10218311846256256, -0.4334276020526886, 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	[-1.875599980354309, -0.40814998745918274, -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, -0.1428000032901764, 0.5...]	[-0.22261293232440948, 0.21325691044330597, 0...]	[-1.3547197580337524, 0.43423181772232056, -1...]	[0.46073707938194275, 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 arrray', ' 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

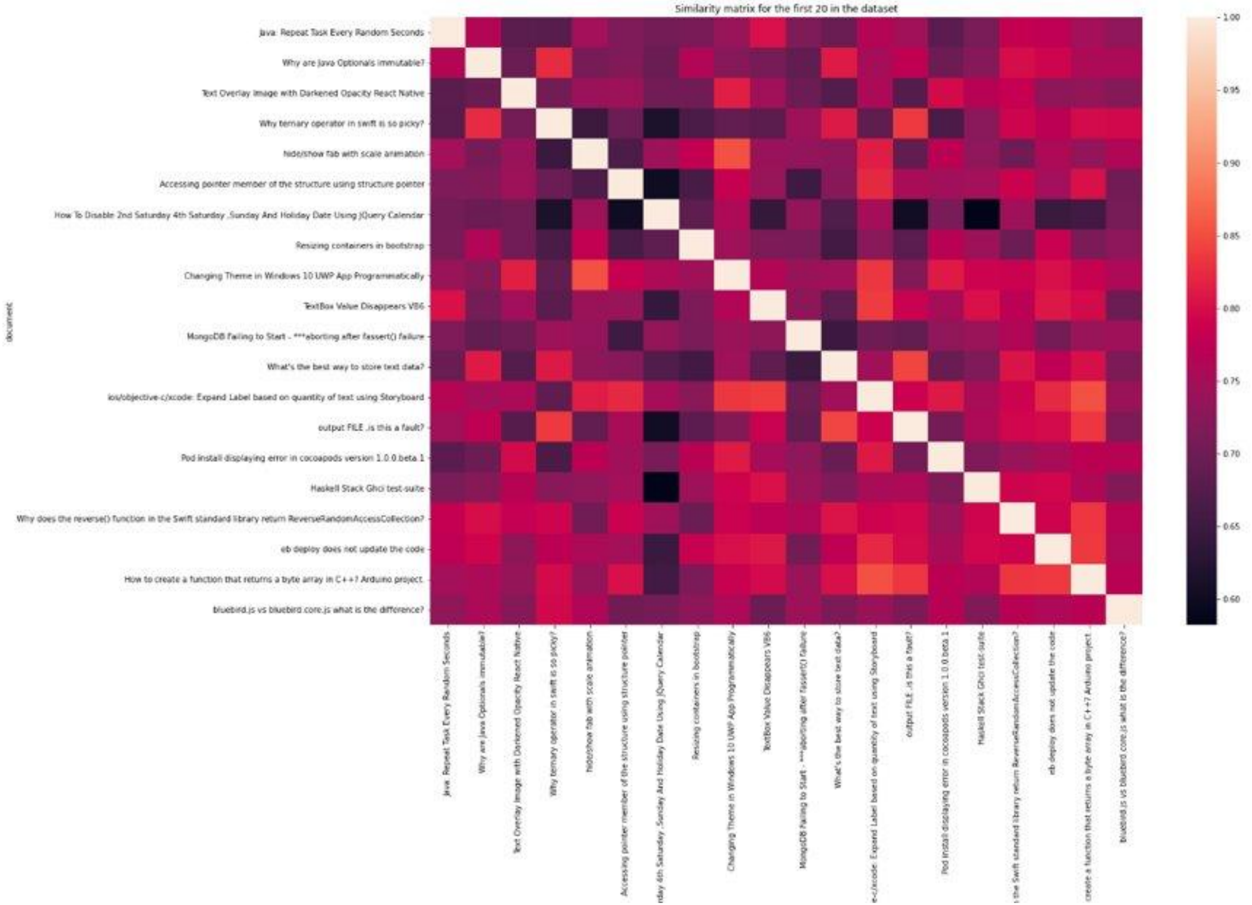
```
# Multiple binary sentiment classifiers trained on various datasets
nlu.load('classify.sentiment').predict('I love NLU and Python WebDev Conf 2021!')
nlu.load('classify.sentiment.imdb').predict('The Matrix was a pretty good movie')
nlu.load('classify.sentiment.twitter').predict('@elonmusk Tesla stock 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 withouth 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('BERTolgy Word embeddings!')
nlu.load('bert elmo albert glove').predict('Multiple BERTolgy Word embeddings!')
nlu.load('embed_sentence.bert ').predict('BERTolgy 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!

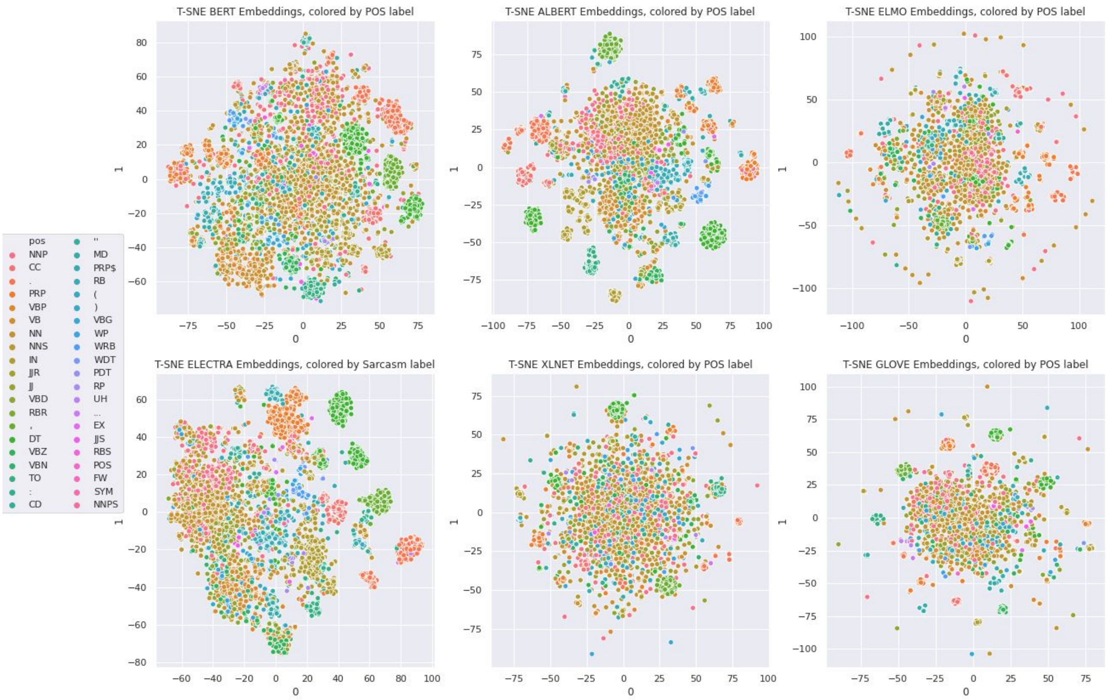
```python
# Multiple binary sentiment classifiers trained on various datasets
nlu.load('classify.sentiment').predict('I love NLU and Python WebDev Conf 2021!')
nlu.load('classify.sentiment.imdb').predict('The Matrix was a pretty good movie')
nlu.load('classify.sentiment.twitter').predict('@elonmusk Tesla 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 withouth 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('BERTolgy Word embeddings!')
nlu.load('bert elmo albert glove').predict('Multiple BERTolgy Word embeddings!')
nlu.load('embed_sentence.bert ').predict('BERTolgy 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')
```

Sentence Similarity With BERTology Embeds or T5



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

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]

Visualize NER results

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

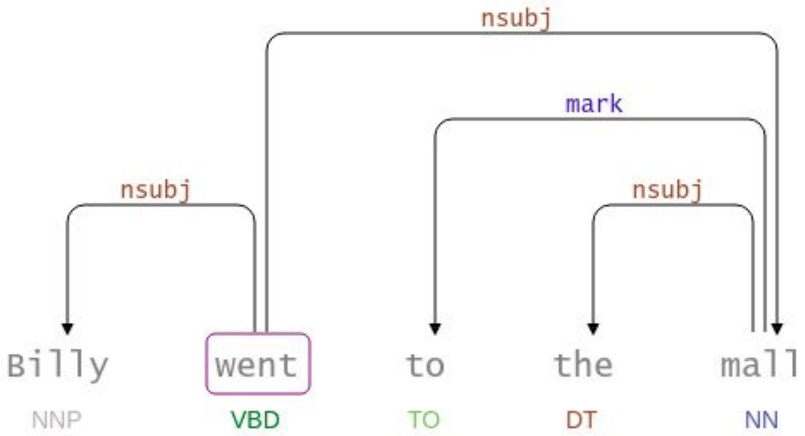
```
onto_recognize_entities_sm download started this may take some time.
Approx size to download 160.1 MB
[OK!]
```

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

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

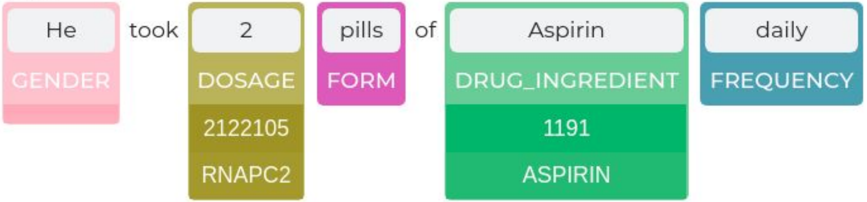
```
nlu.load('med_ner.clinical assert').viz("The MRI scan showed no signs of cancer in the left lung")
```

```
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!]
```

Visualize Resolution

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

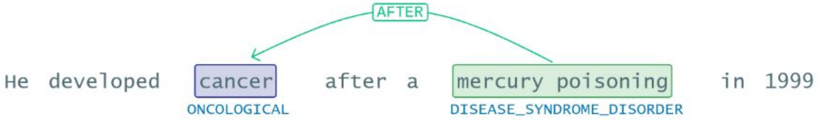
```
jsl_ner_wip_clinical download started this may take some time.
Approximate size to download 14.5 MB
[OK!]
chunkresolve_rxnorm_in_clinical download started this may take some time.
Approximate size to download 26.9 MB
[OK!]
embeddings_clinical download started this may take some time.
Approximate 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!]
```



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_viz_cheatsheet.py

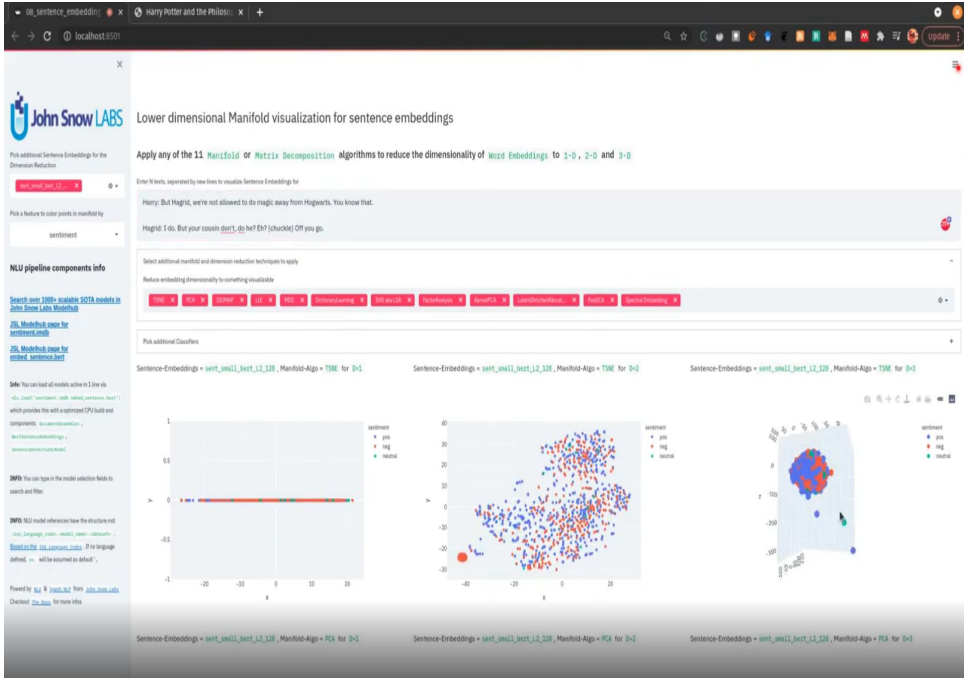
```python
data = 'I want some pretty visualizations please!'

# 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)
```

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

- Compare Multiple similarity Metrics And Embeddings at the same time
- Supported similarities :
    - Cosine
    - Cityblock
    - Euclidean
    - L2
    - L1
    - Manhattan

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



nlu_streamlit_cheatsheet.py

```python
# 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! <3','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()
```

NER Domains and Models Overview

Domain	Description	Sample NLU Spells	Sample Entities	Sample Predicted Labels	Reference Links
ADE (Adverse Drug Events)	Find adverse drug event (ADE) related entities	med_ner.ade_biobert	Aspirin , vomiting	DRUG , ADE	CADEC, Twimed
Anatomy	Find body parts, anatomical sites a nd reference related entities	med_ner.anatomy	tubules, nasopharyngeal aspirates, embryoid bodies, NK cells, Mitochondrial, tracheoesophageal fistulas, heart, colon cancer, cervical, central nervous system	Tissue structure, Organism substance, Developing anatomical structure, Cell, Cellular component, Immaterial_anatomical_entity, organ, Pathological formation, Organism_subdivision, Anatomical_system	AnEM
Cellular/ Molecular Biology	Find Genes, Molecules, Cell or general Biology related entities	med_ner.cellular.biobert	human T-cell leukemia virus type 1 Tax-responsive , primary T lymphocytes, E1A-immortalized, Spi-B mRNA, zeta-globin	DNA, Cell type, Cell_line, RNA, Protein	JNLPBA
Chemical/Genes/ Proteins	Find Chemical, Gene and Protein related entities	med_ner.chemprot.clinical	nitrogen , β-amyloid , NF-kappaB	CHEMICAL, GENE-Y, GENE-N	ChemProt
Chemical Compounds	Find general chemical compound related entities	med_ner.chemicals	resveratrol , β-polyphenol	CHEM	Dataset by John Snow Labs
Drug Chemicals	Find chemical and drug related entities	med_ner.drugs	potassium , anthracyclines, taxanes	DrugChem. DrugChem.DrugChem	i2b2 + FDA
Posology/Drugs	Find posology and drug related entities	med_ner.posology.biobert	5000 units, Aspirin, 14 days, tablets, daily, topically, 30 mg	DOSAGE, DRUG, DURATION, FORM, FREQUENCY, ROUTE, STRENGTH.	i2b2 + FDA
Risk Factors	Find risk factor of patient related entities	med_ner.risk_factors.biobert	coronary artery disease, hypertension, Smokes 2 packs of cigarettes per day, morbid obesity, Actos, Works in School, diabetic, diabetic	CAD, HYPERTENSION, SMOKER, OBESE, FAMILY HIST, MEDICATION, PHI, HYPERLIPIDEMIA, DIABETES	De-identification and Heart Disease Risk Factors Challenge datasets
cancer Genetics	Find cancer and genetics related entities	med_ner.cancer	human, Kir 3.3, GIRK3, potassium, GIRK, chromosome 1q21-23, pancreas, tissues, fat andskeletal muscle, KCNJ9, Type II, breast cancer, patients, anthracyclines, taxanes, vinorelbine, patients, breast, vinorelbine inpatients, anthracyclines	Amino acid, Anatomical_system, cancer, Cell, Cellular component, Developing anatomical Structure , Gene or gene product, Immaterial anatomical_entity, Multi-tissue structure, Organ, Organism , Organism_subdivision, Simple_chemical, Tissue	CG TASK of BioNLP 2013
Diseases	Find disease related entities	med_ner.diseases.biobert	the cyst, a large Prolene suture, a very small incisional hernia, the hernia cavity, omentum, the hernia, the wound lesion, The lesion, the existing scar, the cyst, the wound, this cyst down to its base, a small incisional hernia, The cyst	Disease	CG TASK of BioNLP 2013
Bacterial Species	Find bacterial species related entities	med_ner.bacterial_species	Neisseria wadsworthii, N. bacilliformis, Spirochaeta litoralis	SPECIES	Dataset by John Snow Labs
Medical Problem/Test/Treatment	Find medical problem,test and treatment related entities	med_ner.healthcare	respiratory tract infection , Ourexpression studies, atorvastatin	PROBLEM, TEST, TREATMENT	i2b2
Clinical Admission Events	Find clinical admission event related entities	med_ner.admission_events	2007, 12 AM, Headache, blood sample, presented, emergency room, daily	DATE, TIME, PROBLEM, TEST, TREATMENT, OCCURENCE, CLINICAL DEPT, EVIDENTIAL, DURATION, FREQUENCY, ADMISSION, DISCHARGE	Custom i2b2, enriched with Events
Genetic Variants	Find genetic variant related entities	en.med_ner.genetic_variants	rs1061170, p.S45P, T13046C	DNAMutation, ProteinMutation, SNP	TMVAR
PHI (Protected Healthcare Information)	Find PHI(Protected Healthcare) related entities	en.med_ner.deid	2093-01-13, David Hale, Hendrickson,</br> Ora, 7194334, 01/13/93, Oliveira, 25-year-old, 1-11-2000, Cocke County Baptist Hospital, 0295 Keats Street., (302) 786-5227, Brothers Coal-Mine	MEDICALRECORD, ORGANIZATION, DOCTOR, USERNAME, PROFESSION, HEALTHPLAN, URL, CITY, DATE, LOCATION-OTHER, STATE, PATIENT, DEVICE, COUNTRY, ZIP, PHONE, HOSPITAL, EMAIL, IDNUM, SREET, BIOID, FAX, AGE	n2c2 i2b2-PHI
Social Determinants / Demographic Data	Find Social Determinants and Demographic Data Related Entities	med_ner.jsl.enriched	21-day-old, male, congestion, mom, suctioning yellow discharge, she, problems with his breathing, perioral cyanosis, retractions, mom, Tylenol, His, his, respiratory congestion, He, tired, fussy, albuterol	Age, Diagnosis, Dosage, Drug Name, Frequency, Gender, Lab_Name, Lab_Result, Symptom_Name	Dataset by John Snow Labs
General Clinical	Find General Clinical Entities	med_ner.jsl.wip.clinical.modifier	28-year-old, female, gestational, diabetes, mellitus, eight, years, prior, type, two, diabetes, mellitus, T2DM, HTG-induced, pancreatitis, three, years, prior, acute, hepatitis, obesity, body, mass, index, BMI, kg/m2, polyuria, polydipsia, poor, appetite, vomiting, Two, weeks, prior, she, five-day, course	Injury or Poisoning, Direction, Test, Admission Discharge, Death Entity, Relationship Status, Duration, Respiration, Hyperlipidemia, Birth Entity, Age, Labour_Delivery, Family History_Header, BMI, Temperature, Alcohol, Kidney Disease, Oncological, Medical History Header, Cerebrovascular_Disease, Oxygen_Therapy, O2 Saturation, Psychological Condition, Heart Disease, Employment, Obesity, Disease Syndrome Disorder, Pregnancy, ImagingFindings, Procedure, Medical Device, Race Ethnicity, Section Header, Symptom, Treatment, Substance, Route, Drug Ingredient, Blood Pressure, Diet, External body part or region, LDL, VS Finding, Allergen, EKG Findings, Imaging Technique, Triglycerides, RelativeTime, Gender, Pulse, Social History Header, Substance_Quantity, Diabetes, Modifier, Internal organ or component, Clinical Dept, Form, Drug_BrandName, Strength, Fetus NewBorn, RelativeDate, Height, Test Result, Sexually Active or Sexual Orientation, Frequency, Time, Weight, Vaccine, Vital Signs Header, Communicable Disease, Dosage, Overweight, Hypertension, HDL, Total_Cholesterol, Smoking,`	Dataset by John Snow Labs
			Bilateral, breast, ultrasound,		

NER Domains and Models Overview

Radiology	Find Radiology related entities	med_ner.radiology.wip_clinical	Bilateral, breast, ultrasound, ovoid mass, 0.5 x 0.5 x 0.4, cm, anteromedial aspect, left, shoulder, mass, isoechoic 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	Dataset by John Snow Labs, MIMIC-CXR and MT Radiology texts
Radiology Clinical JSL-V1	Find radiology related entities in clinical setting	med_ner.radiology.wip_greedy_biobert	Bilateral, breast, ultrasound, ovoid mass, 0.5 x 0.5 x 0.4, cm, anteromedial aspect, left, shoulder, mass, isoechoic 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	Dataset by John Snow Labs,
Genes and Phenotypes	Find Genes and Phenotypes (the observable physical properties of an organism) related entities	med_ner.human_phenotype.gene_biobert	APOC4 , polyhydramnios	GENE, PHENOTYPE	PGR_1, PGR_2
Normalized Genes and Phenotypes	Find Normalized Genes and Phenotypes (the observable physical properties of an organism) related entities	med_ner.human_phenotype.go_biobert	protein complex oligomerization , defective platelet aggregation	GO, HP	PGR_1, PGR_2
Radiology Clinical JSL-V2	Find radiology related entities in clinical setting	med_ner.jsl.wip.clinical.rd		Kidney Disease, HDL, Diet, Test, Imaging Technique, Triglycerides, Obesity, Duration, Weight, Social History Header, ImagingTest, Labour_Delivery, Disease Syndrome Disorder, Communicable Disease, Overweight, Units, Smoking, Score, Substance Quantity, Form, Race Ethnicity, Modifier, Hyperlipidemia, ImagingFindings, Psychological Condition, OtherFindings, Cerebrovascular Disease, Date, Test_Result, VS Finding, Employment, Death Entity, Gender, Oncological, Heart Disease, Medical Device, Total Cholesterol, ManualFix, Time, Route, Pulse, Admission Discharge, RelativeDate , O2 Saturation, Frequency, RelativeTime, Hypertension, Alcohol, Allergen, Fetus NewBorn, Birth Entity, Age, Respiration, Medical History Header, Oxygen Therapy, Section Header, LDL, Treatment, Vital_Signs_Header, Direction, BMI, Pregnancy, Sexually 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	Dataset by John Snow Labs,
General Medical Terms	Find general medical terms and medical entities.	med_ner.medmentions		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 Activity, Gene or Genome, Indicator_Reagent_or_Diagnostic_Aid, Biologic Function, Chemical, Mammal, Molecular Function, Quantitative Concept, Prokaryote, Mental or Behavioral Dysfunction, Injury or Poisoning, Body Location or Region, Spatial Concept, Nucleotide_Sequence, Tissue, Pathologic Function, Body Substance, Fungus, Mental Process, Medical Device, Plant, Health Care Activity, Clinical Attribute, Genetic Function, Food, Therapeutic_or_Preventive_Procedure, Body Part 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, Population Group, Disease or Syndrome, Group, Sign or Symptom, Body System	MedMentions

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`	- `Confirmed` : X-Ray scan shows `cancer` in lung. - `Negative` : X-Ray scan shows no sign of `cancer` in lung. - `Suspected` :X-Ray raises `suspicion of cancer` in lung but does not confirm it.	Internal Dataset by Annotated by John Snow Labs
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`, `Hypotetical`	- `Present` : Patient diagnosed with `cancer` in 1999 - `Absent` : No sign of `cancer` was shown by the scans - `Possible` : Tests indicate patient might have `cancer` - `Planned` : `CT-Scan` is scheduled for 23.03.1999 - `Someoneelse` : The patient gave `Aspirin` to daugther. - `Past` : The patient has no more `headaches` since the operation - `Family` : The patients father has `cancer`. - `Hypotetical` : `Death` could be possible.	2010 i2b2 + Data provided by JSL
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`	- `present` : Patient diagnosed with `cancer` in 1999 - `absent` : No sign of `cancer` was shown by the scans - `possible` : Tests indicate patient might have `cancer` - `planned` : `CT-Scan` is scheduled for 23.03.1999 - `someoneelse` : The patient gave `Aspirin` to daugther - `past` : The patient has no more `headaches` since the operation	2010 i2b2 + Data provided by JSL
Healthcare/Clinical classic	Predict status of Healthcare/Clinical related entities	`assert.biobert`	`present`, `absent`, `possible`, `conditional`, `associated_with_someone_else`, `hypothetical`	- `present` : Patient diagnosed with `cancer` in 1999 - `absent` : No sign of `cancer` was shown by the scans - `possible` : Tests indicate patient might have `cancer` - `conditional` If the test is positive, patient has `AIDS` - `associated with someone_else` : The patients father has `cancer`. - `hypothetical` : `Death` could be possible.	2010 i2b2

Resolution Domains Models Overview

Domain/Terminology	Description	Sample NLU Spells	Sample Entities	Sample Predicted Codes	Reference Links
ICD-10 / ICD-10-CM (International Classification of Diseases - Clinical Modification)	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	ICD-10-CM, WHO ICD-10-CM
ICD-10-PCS (International Classification of Diseases - Procedure Coding System)	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 maintaining by Centers for Medicare & Medicaid Services	resolve.icd10pcs	hypertension, gastritis	DWY18ZZ, 04723Z6	ICD10-PCS, CMS ICD-10-PCS
ICD-O (International Classification of Diseases, Oncollogy) Topography & Morphology codes	Get ICD-0 codes of Medical and Clinical Entities. The International Classification of Diseases for Oncology (ICD-0), 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	9050/3 + C38.3, 8001/3 + C39.8	ICD-O Histology Behaviour dataset
HCC (Hierachical Conditional Categories)	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	HCC
ICD-10-CM + HCC Billable	Get ICD-10-CM and HCC codes of Medical and Clinical Entities.	resolve.icd10cm.augmented_billable	metastatic lung cancer	C7800 + ['1', '1', '0']	ICD10-CM HCC
CPT (Current Procedural Terminology)	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/diagonstics. 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	CPT
LOINC (Logical Observation Identifiers Names and Codes)	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	LOINC
HPO (Human Phenotype Ontology)	Get HPO codes of Medical and Clinical Entities.	resolve.HPO	cancer, bipolar disorder	0002664, 0007302, 0100753	HPO
UMLS (Unified Medical Language System) CUI	Get UMLS codes of Medical and Clinical Entities.	resolve.umls.findings	vomiting, polydipsia, hepatitis	C1963281, C3278316, C1963279	UMLS
SNOMED International (Systematized Nomenclature of Medicine)	Get SNOMED (INT) codes of Medical and Clinical Entities.	resolve.snomed.findings_int	hypertension	148439002	SNOMED
SNOMED CT (Clinical Terms)	Get SNOMED (CT) codes of Medical and Clinical Entities.	resolve.snomed.findings	hypertension	73578008	SNOMED
SNOMED Conditions	Get SNOMED Conditions codes of Medical and Clinical Entities.	resolve.snomed_conditions	schizophrenia	58214004	SNOMED
RxNorm and RxCUI (Concept Uinque Indentifier)	Get Normalized RxNorm and RxCUI codes of Medical, Clinical and Drug Entities.	resolve.rxnorm	50 mg of eltrombopag oral	825427	[RxNorm Overview] [November 2020 RxNorm Clinical Drugs ontology graph]

Relation Extraction Domains and Models examples

Domain	Description	Sample NLU Spells	Predictable Relationships and Explanation
Dates and Clinical Entities	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
Body Parts and Directions	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
Body Parts and Problems	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
Body Parts and Procedures	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
Adverse Effects between drugs (ADE)	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
Phenotype abnormalities,Genes and Diseases	Predict binary caused by relationship between Phenotype Abnormality Entities, Gene Entities and Disease Entities	relation.humen_phenotype_gene	- 1 for Gene Entity and Phenotype Entity are related - 0 for Gene Entity and Phenotype Entity are not related
Temporal events	Predict multi-class temporal relationship between Time Entities and Event Entities	relation.temporal_events	- AFTER if Any Entity occured after Another Entity - BEFORE if Any Entity occured before Another Entity - OVERLAP if Any Entity during Another Entity
Dates and Tests/Results	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
Clinical Problem, Treatment and Tests	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 - TrWP : 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 - TrNAP : 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
DDI Effects of using Multiple Drugs (Drug Drug Interaction)	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.
Posology (Drugs, Dosage, Duration, Frequency, Strength)	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
Chemicals and Proteins	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	Internal Dataset by Annotated by John Snow Labs
Body Parts and Directions	MRI demonstrated infarction in the upper - brain stem , left cerebellum and right basil ganglia	- 1 for uppper and brain stem - 0 for upper and cerebellum - 1 for left and cerebellum	Internal Dataset by Annotated by John Snow Labs
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	Internal Dataset by Annotated by John Snow Labs
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	Internal Dataset by Annotated by John Snow Labs
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 Liptor - 0 for sever fatigue and voltaren - 0 for cramps and Liptor - 1 for cramps and voltaren	Internal Dataset by Annotated by John Snow Labs
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	PGR aclAntology
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 addmitted and Mayo Clinic - BEFORE for admitted and discharged	Temporal JSL Dataset and n2c2
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	Internal Dataset by Annotated by John Snow Labs
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 : Azotemia 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 Azotemia and sepsis	2010 i2b2 relation challenge
DDI Effects of using Multiple Drugs (Drug Drug Interaction)	- DDI-advise : 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 theobromine - DDI-false : Aspirin does not interact with Chlorthalidone	- DDI-advise 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	DDI Extraction corpus
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	Magge, Scotch, Gonzalez-Hernandez (2018)
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 agnonists 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 IFNα/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 palinurin 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 Ibandronate 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 IFNα/β - CPR:9 (Substrate) for ZIP9 and Cd(2+) cells - CPR:10 (Not Related) for GSK-3β and ATP	ChemProt Paper

NLU OCR

- Extract text from
  - **Images**
  - **PDFs**
  - **DOCX**

NLU Spell	Transformer Class
nlu.load(`img2text`)	ImageToText
nlu.load(`pdf2text`)	PdfToText
nlu.load(`doc2text`)	DocToText

https://nlu.johnsnowlabs.com/docs/en/nlu_for_ocr

Image to Text

Sample image:

“The Old Pond” by Matsuo Bashō

An old silent pond

A frog jumps into the pond—

Splash! Silence again.

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

Output of IMG OCR:

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

PDF to Text

haiku.pdf

Sample PDF:

“Lighting One Candle” by Yosa Buson

The light of a candle

Is transferred to another candle—

Spring twilight

```
nlu.load('pdf2text').predict('path/to/haiku.pdf')
```

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

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.

```
nlu.load('doc2text').predict('path/to/haiku.docx')
```

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.

Combine OCR and NLP models

Sample image containing named entities **from U.S. Presidents Wikipedia**:

Four presidents died in office of natural causes (William Henry Harrison, Zachary Taylor, Warren G. Harding, and Franklin D. Roosevelt), four were assassinated (Abraham Lincoln, James A. Garfield, William McKinley and John F. Kennedy), and one resigned (Richard Nixon, facing impeachment).[9] John Tyler was the first vice president to assume the presidency during a presidential term, and set the precedent that a vice president who does so becomes the fully functioning president with his presidency, as opposed to a caretaker president.[10] The Twenty-fifth Amendment to the Constitution put Tyler's precedent into law in 1967. It also established a mechanism by which an intra-term vacancy in the vice presidency could be filled. Richard Nixon was the first president to fill a vacancy under this provision when he selected Gerald Ford for the office following Spiro Agnew's resignation in 1973. The following year, Ford became the second to do so when he chose Nelson Rockefeller to succeed him after he acceded to the presidency. As no mechanism existed for filling an intra-term vacancy in the vice presidency before 1967, the office was left vacant until filled through the next ensuing presidential election and subsequent inauguration.[11]

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

Output of image OCR and NER NLP :

entities_ner	entities_ner_class	entities_ner_confidence
Four	CARDINAL	0.9986
Abraham Lincoln	PERSON	0.705514
John F. Kennedy),	PERSON	0.966533
one	CARDINAL	0.9457
Richard Nixon,	PERSON	0.71895
John Tyler	PERSON	0.9929
first	ORDINAL	0.9811
The Twenty-fifth Amendment	LAW	0.548033
Constitution	LAW	0.9762
Tyler's	CARDINAL	0.5329
1967	DATE	0.8926
Richard Nixon	PERSON	0.99515
first	ORDINAL	0.9588
Gerald Ford	PERSON	0.996

More NLU Ressources

- Join our Slack
- NLU Website
- NLU Githuab
- 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
- Indepth 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+ Healtchare models in NLU
- Multi Lingual NLU Webinar - Tutorial on Chinese News dataset
- 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

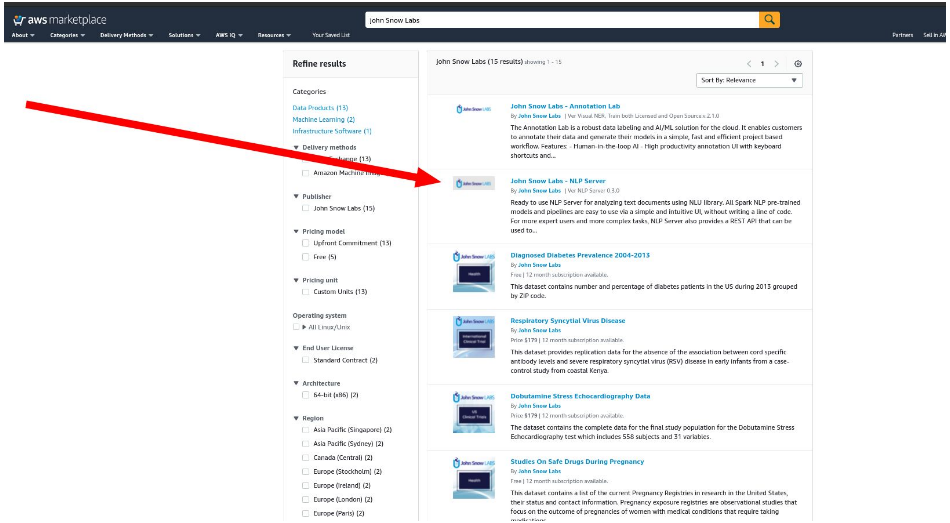
Part - IV (Day 2) - Coding Time

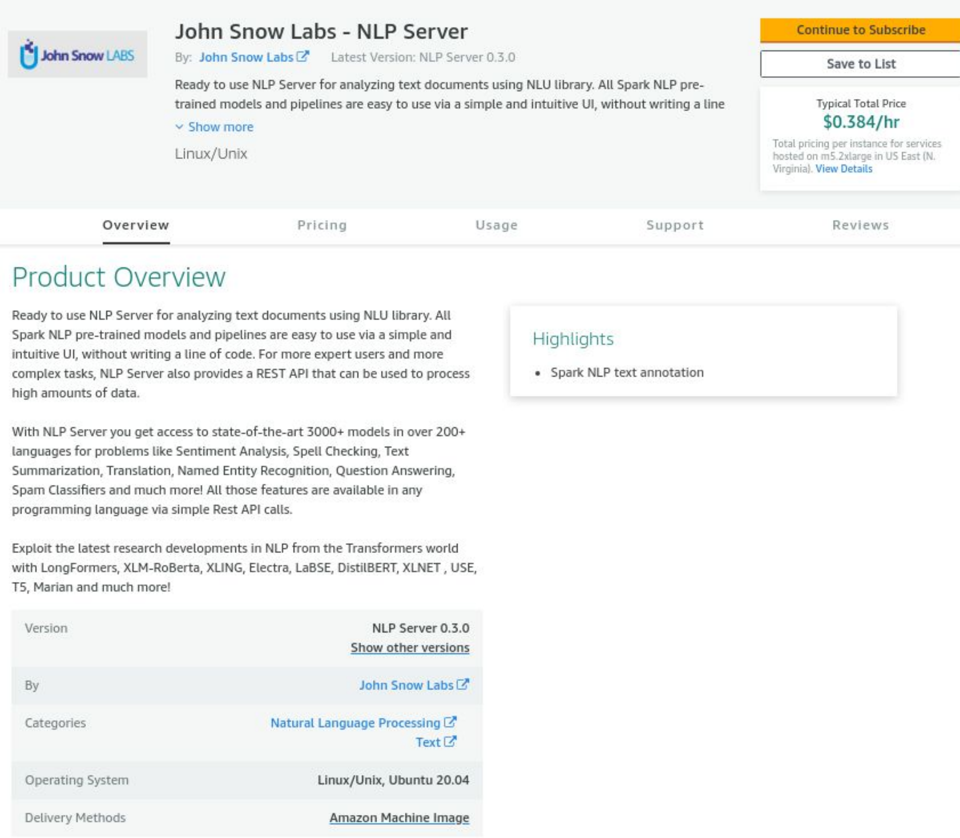
- [Notebook 13 - NLU Crash course, every Spark NLP model in 1 line]

Spark NLP
for Data Scientists

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

NLP Server - Powerful private deployable NLP Rest API

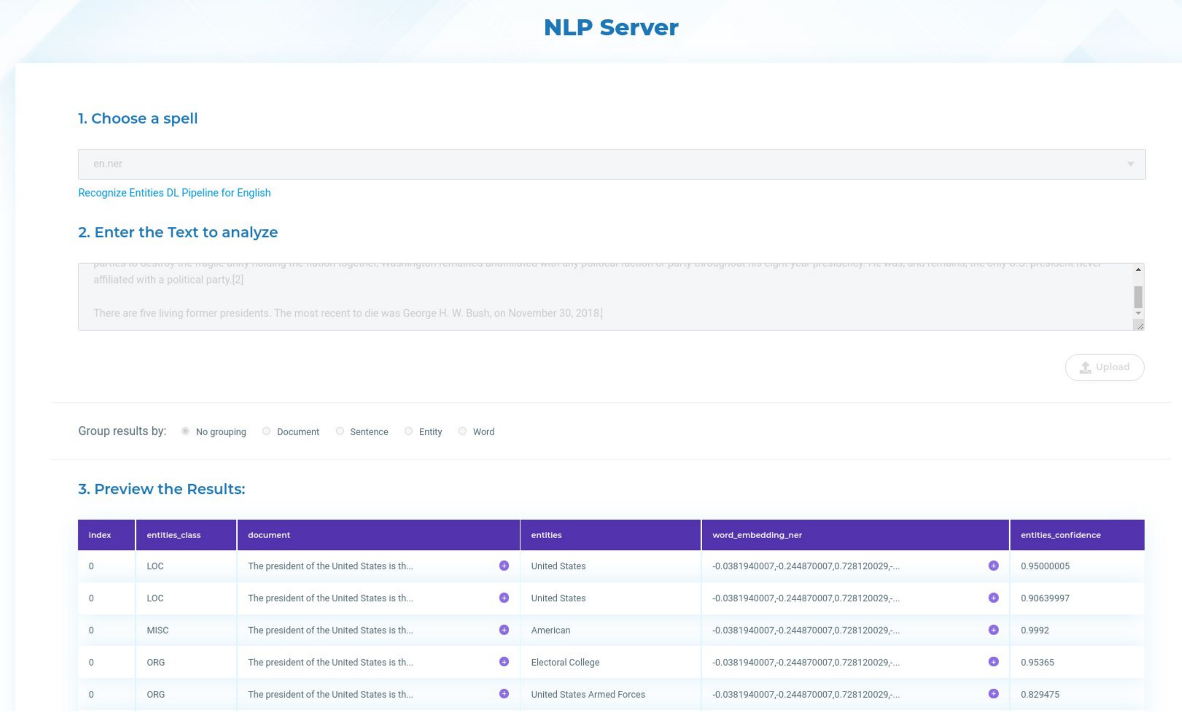




https://nlp.johnsnowlabs.com/docs/en/nlp_server/nlp_server

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

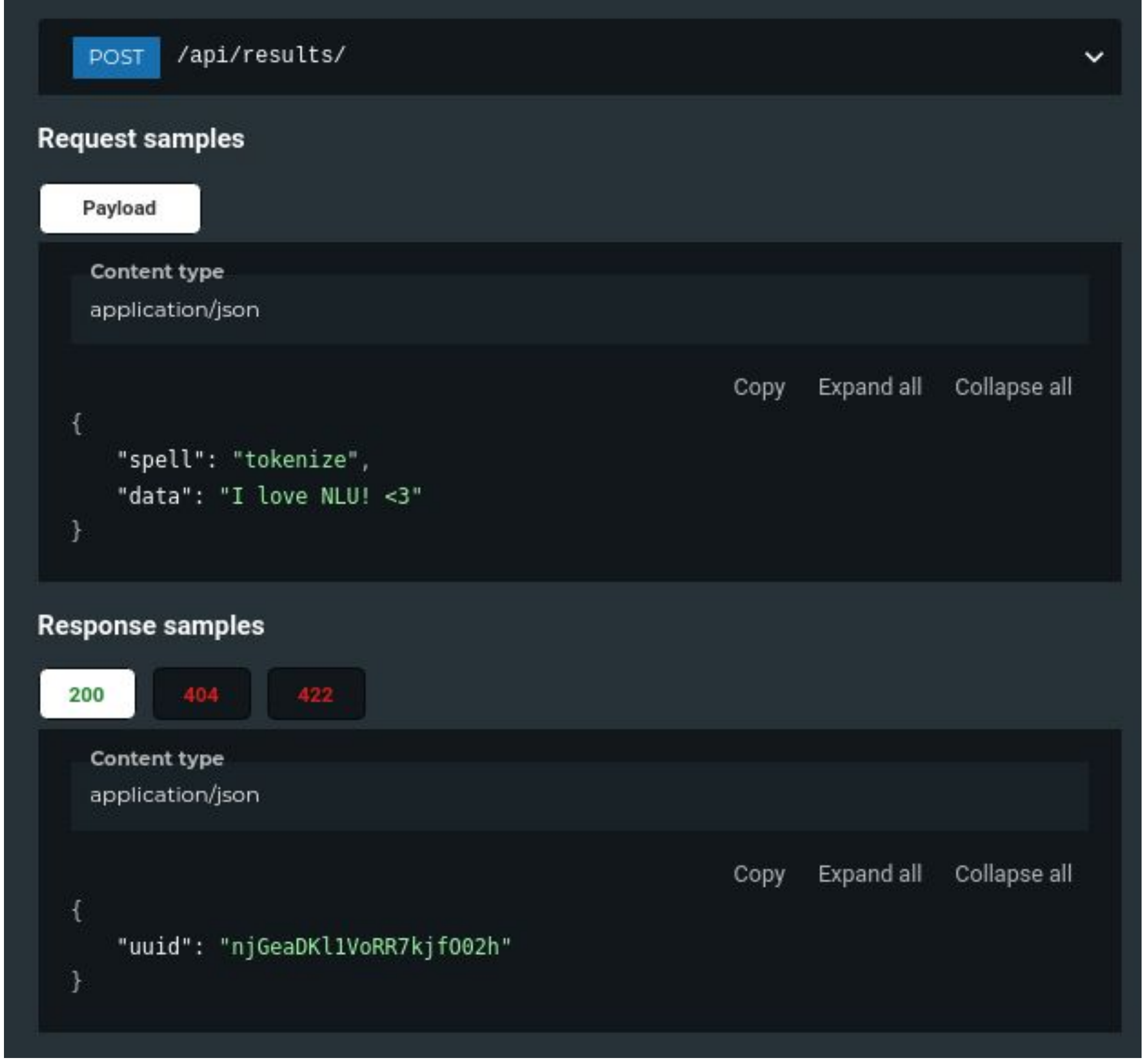


API Usage in 3 Steps :

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

Step 1 : Post a request

- Post a data processing request
- Returns UUID status

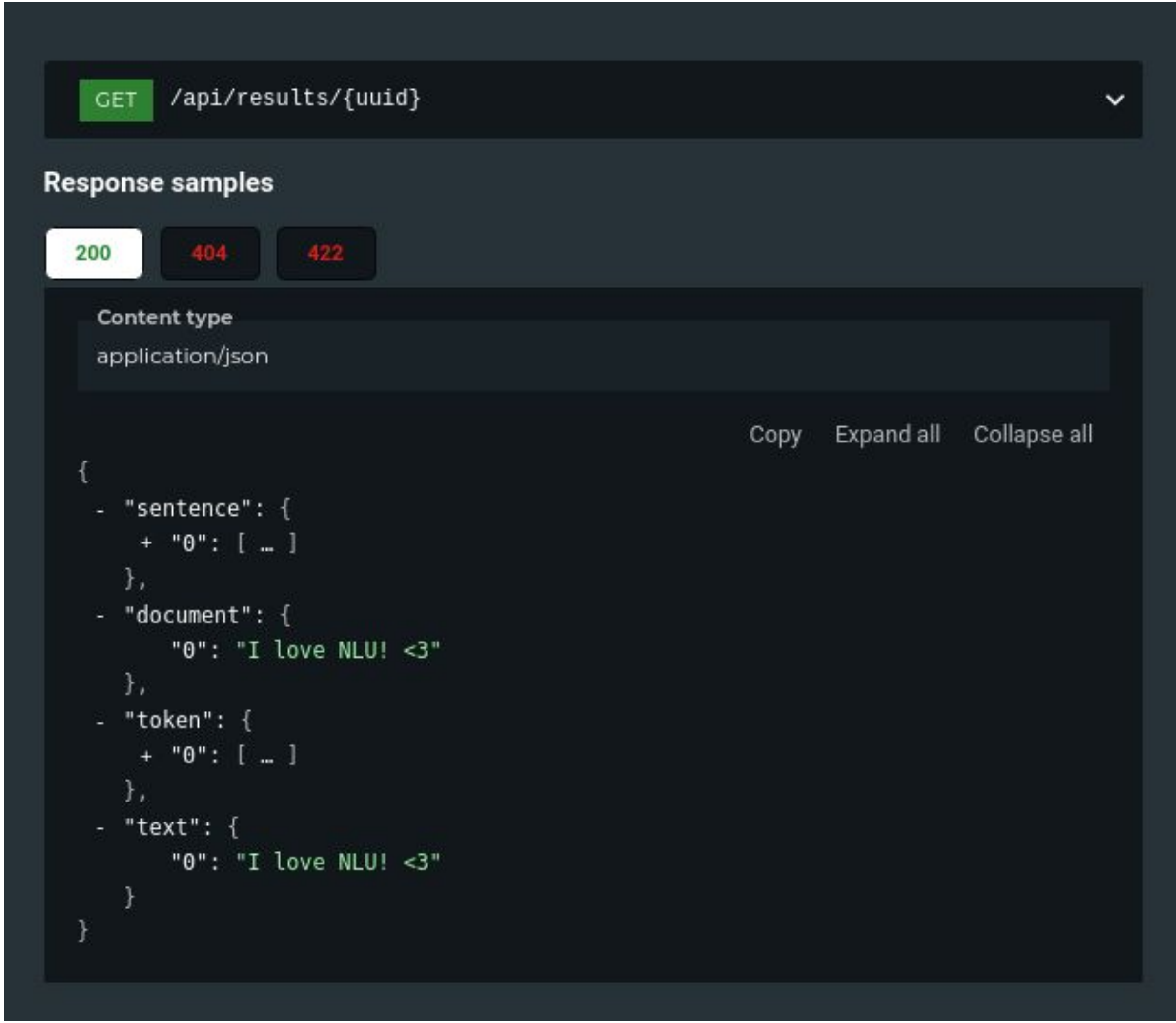


Step 2 : Poll for result

- Poll Server for processing status

Step 3 : Get result as JSON

- Get result for UUID
- Contains NLP predictions



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

Youtube Tutorial for NLP Server Deployment on AWS

Thank you.

christian@JohnSnowLabs.com

@ckl_it

In/Christian-Kasim-Loan

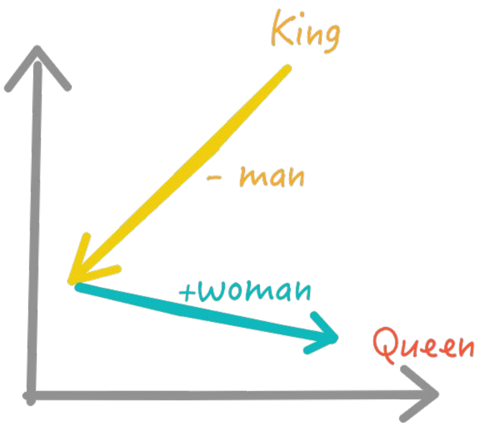
Medium.com/@Christian.Kasim.Loan

Bonus:

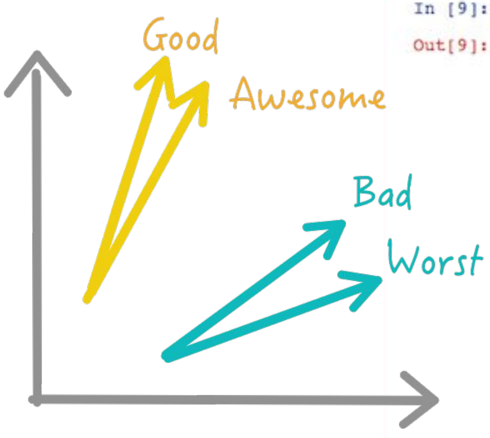
- Semantic Search and Similarity with Textual Embeddings
- Visualize Embeddings via Manifold and Decomposition Algorithms
- Notebook 11 Similarities and Dimension Reduction

Christian Kasim Loan
Lead Data Scientist
christian@johnsnowlabs.com

Word & Sentence Embeddings - reminder



a) Learns Analogy



b) Similar Words have same angles

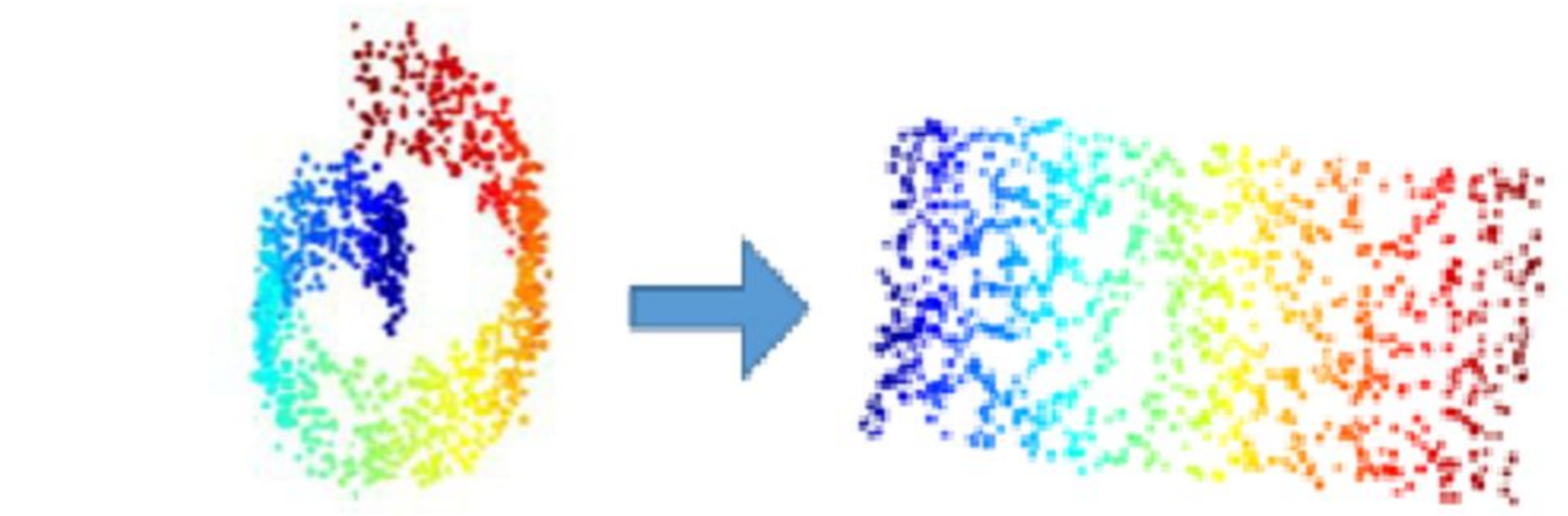
```
In [9]: doc[3].vector
Out[9]: array([ 0.037103 , -0.31259  , -0.17857  ,  0.30001  ,  0.078154 ,
                0.17958  ,  0.12048  , -0.11879  , -0.20601  ,  1.2849   ,
               -0.20409  ,  0.80613  ,  0.34344  , -0.19191  , -0.084511 ,
                0.17339  ,  0.042483 ,  2.0282   , -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 ,
               -0.3617   , -0.18355  , -0.34717  , -0.3691   ,  0.16881  ,
                0.21018  , -0.38376  , -0.096909 , -0.36296  , -0.37319  ,
                0.0021152,  0.32512  ,  0.063977 ,  0.36249  , -0.26935  ,
               -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  ,  0.44076  ,  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  ,
               -0.19228  ,  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 , -0.10066  ,
               -0.4165   , -0.20252  ,  0.2794   ,  0.10852  , -0.40154  ,
```

- Deep-Learning-based natural language processing systems.
- They 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.

Data Manifolds

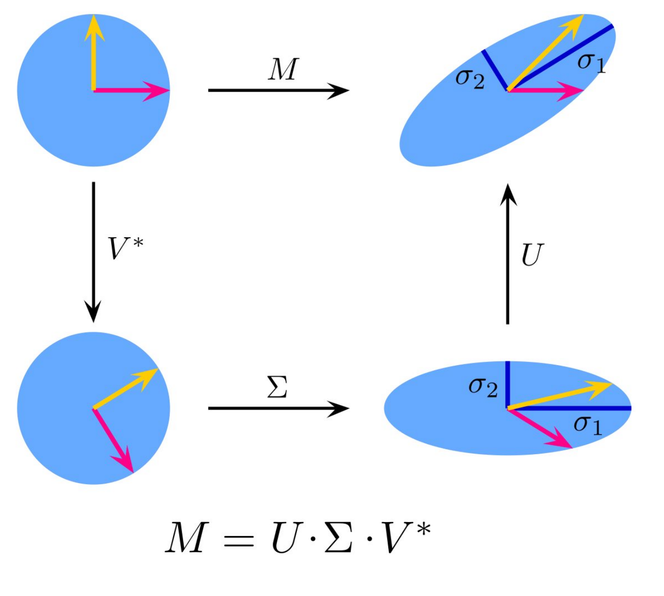
Reduce dimensionality by learning some lower dimensional structure

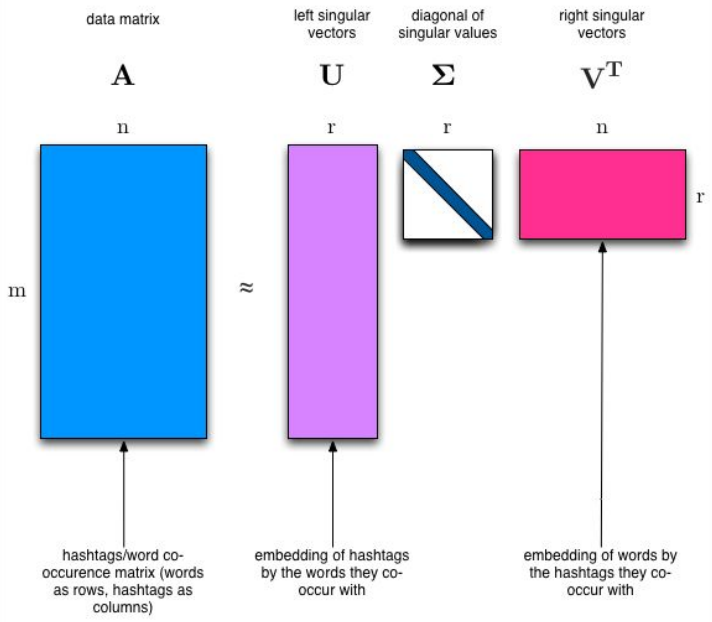
Which preserves relationship between of original hyperspace



Matrix Decompositions

Reduce dimensionality by decomposing Embedding Matrix into lower dimensions which partitions data shape, potentially revealing hidden structure





How to find your way in Spark NLP Universe?

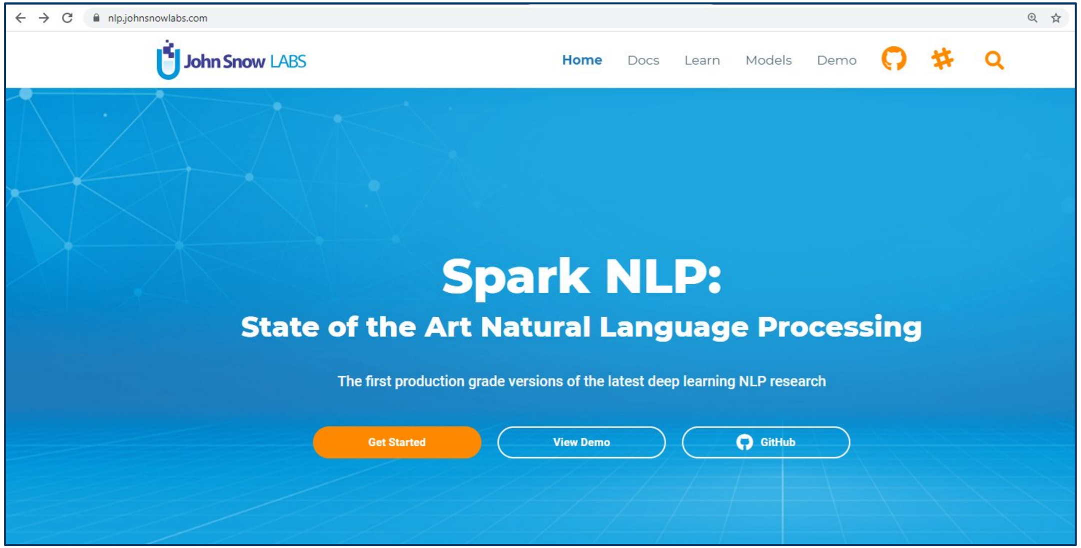
A guidebook that contains all the helpful blogs, GitHub repos, documents, and demos of **Spark NLP**.

The Table of Contents

- Models Hub
- Streamlit Demos
- GitHub Repositories
- Content of the Certification Training
- Slack Channels
- Docs
- Blog Posts
- Youtube Channel

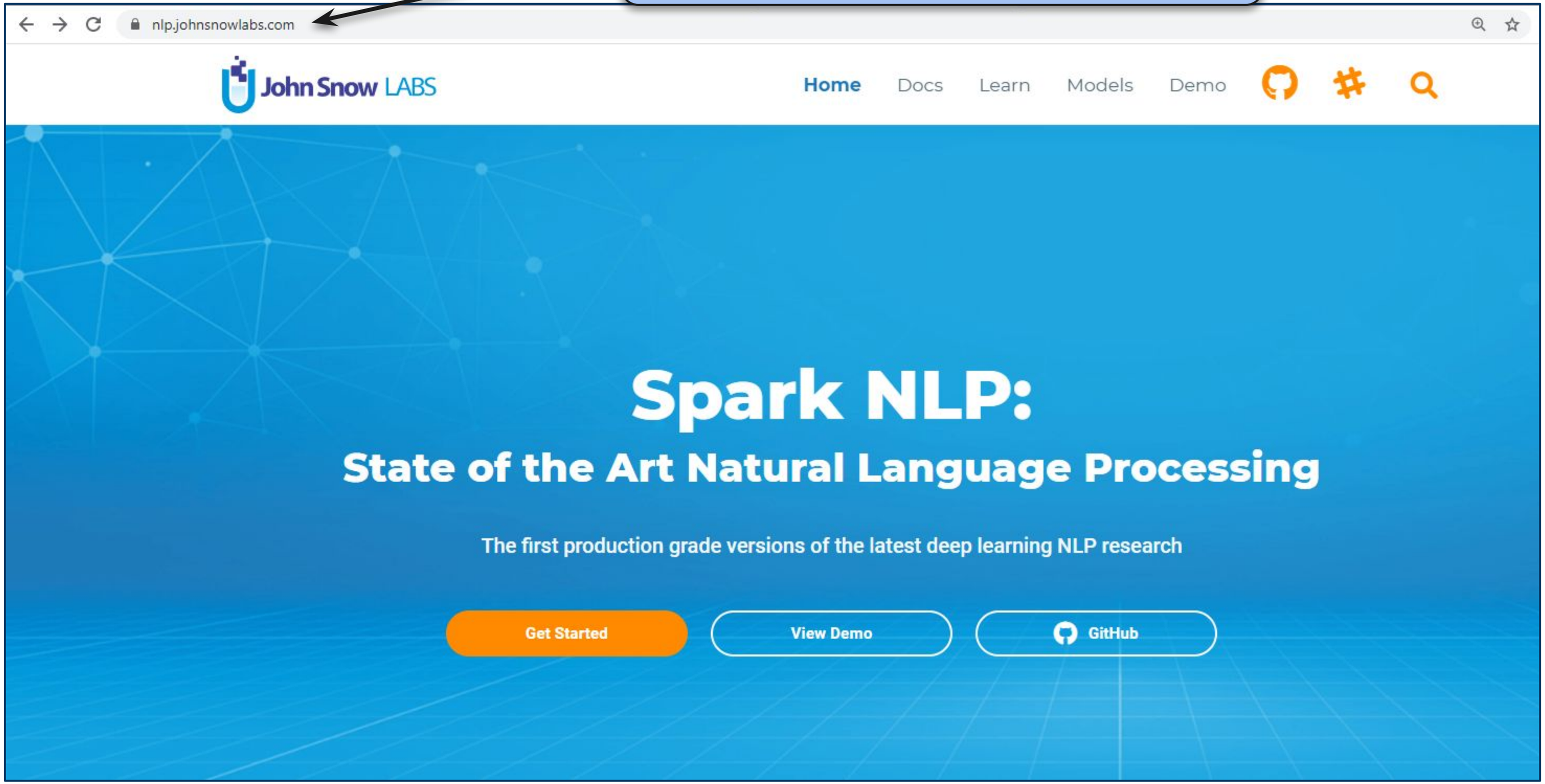
Models Hub

- Main Page
- Models Hub
- Description Pages
  - How to Use
  - Demo Pages
  - Colab Notebooks
  - Sample Code Snippets
  - Benchmarking (Metrics)



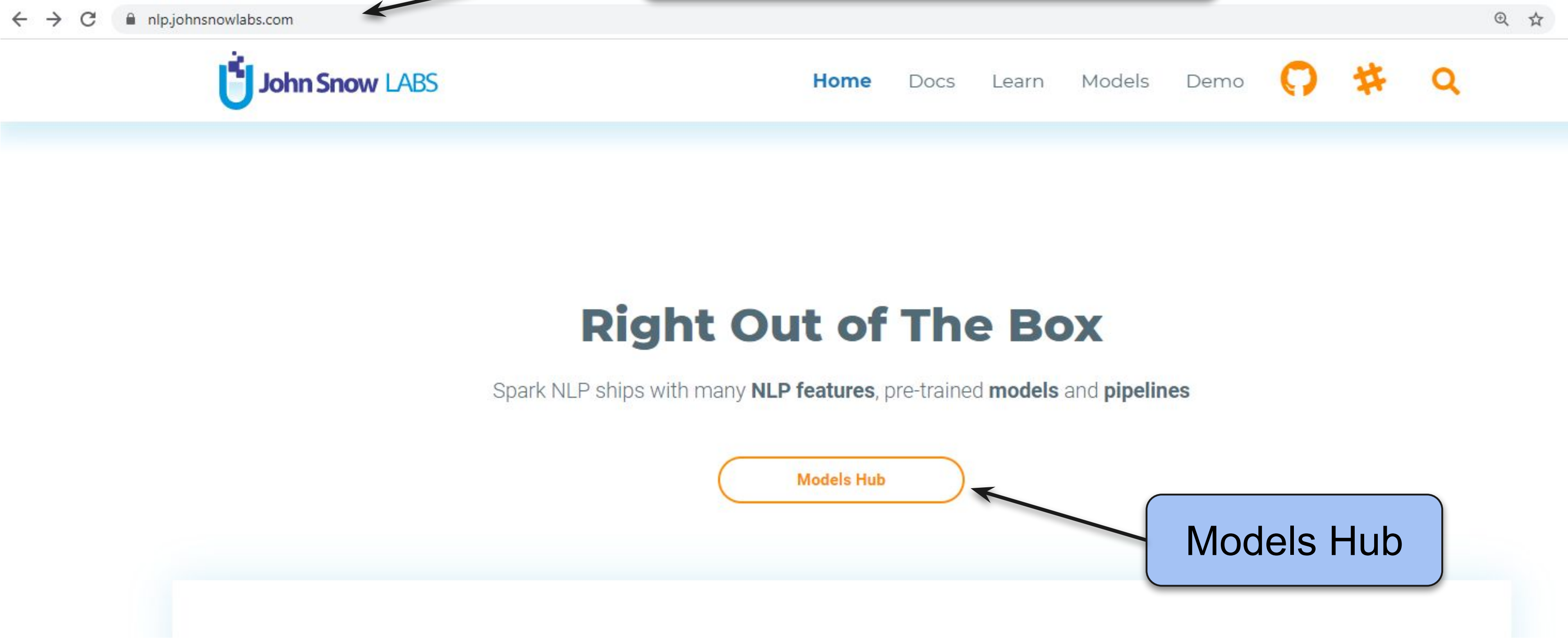
Main Page

http://nlp.johnsnowlabs.com/



Models Hub

http://nlp.johnsnowlabs.com/



Models Hub

Models Hub

nlp.johnsnowlabs.com/models

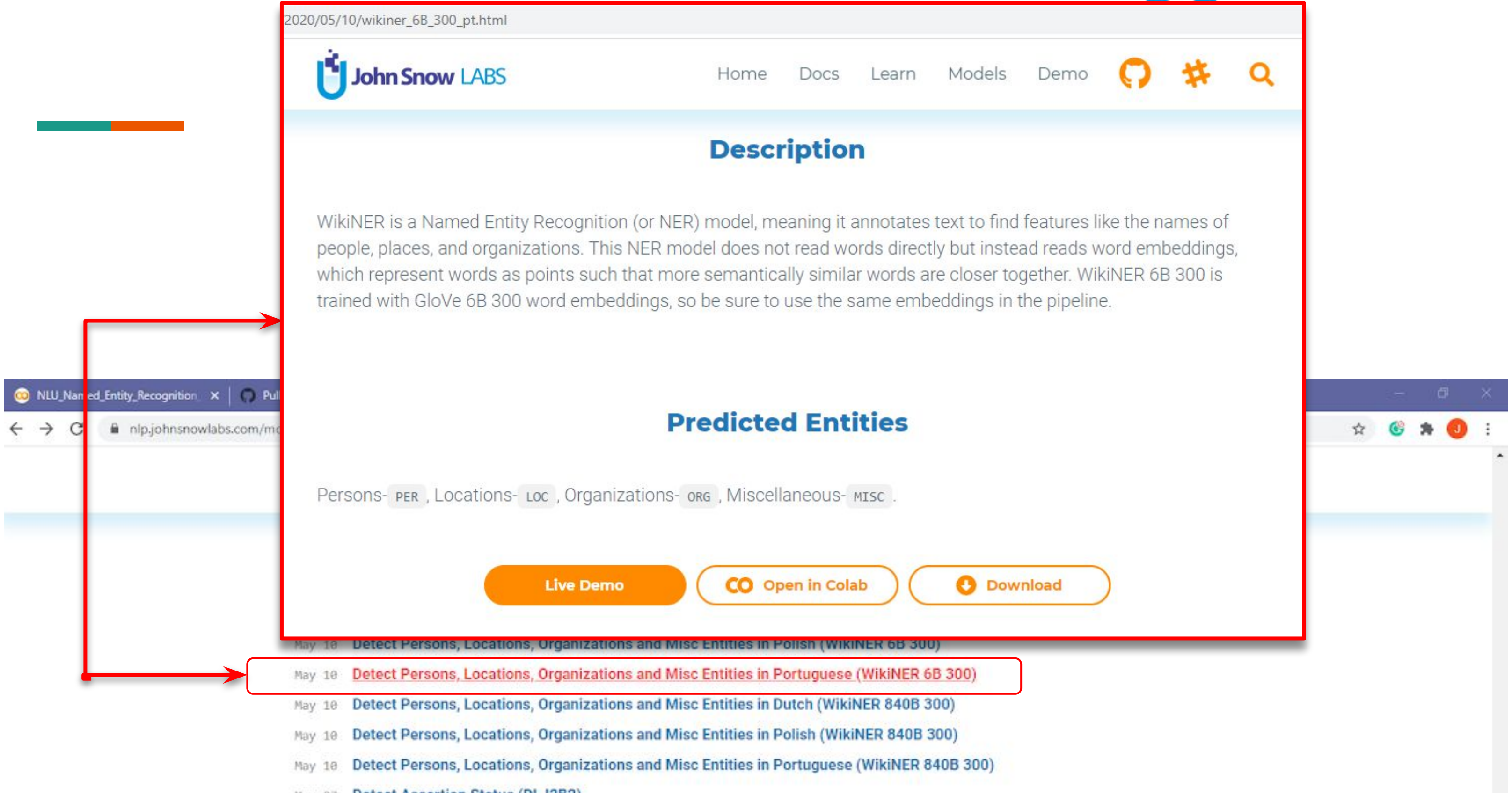
John Snow LABS

Home Docs Learn Models Demo

Available Models and Pipelines

Show All 1036, aav 4, af 5, afa 4, alv 4, am 2, ar 9, art 2, ase 2, assertion 2, az 4, bat 4, bcl 4, bem 4, ber 4,
bg 7, bh 2, bho 2, bi 4, bn 5, bnt 4, br 3, bzs 4, ca 7, cau 2, ccs 2, ceb 4, cel 4, chk 4, classifier 19,
clinical 101, cn 7, cpf 4, cpp 4, crs 4, cs 7, cus 4, cy 4, da 6, de 12, deidentify 4, dra 4, ee 4, efi 4, el 5,
embeddings 55, en 809, entity_resolution 29, eo 5, es 22, et 6, eu 7, euq 4, fa 5, fi 7, fiu 4, fj 4, fr 7, ga 7,
gaa 4, gem 4, gil 4, gl 7, gmq 4, gmw 4, grk 4, guw 4, gv 4, ha 5, he 7, hi 7, hil 4, ho 4, ht 4, hu 7,
hy 7, id 7, ig 4, iir 4, ilo 4, inc 4, ine 4, is 4, iso 4, it 7, itc 4, ja 7, jap 4, ka 2, kab 2, kg 4, kj 4, kl 2,
ko 6, kqn 4, kwn 4, kwy 4, la 3, language_detection 16, lemmatizer 41, lg 4, licenced 1, licensed 82, ln 4, loz 4,
lu 4, lua 4, lue 4, lun 4, luo 4, lus 4, lv 5, map 2, mfe 4, mg 4, mh 4, mk 4, mkh 4, ml 4, mos 4, mr 7,
mt 4, mul 4, nb 2, ner 79, ng 4, nic 4, niu 4, nl 9, nn 1, nso 4, ny 4, nyk 4, om 4, open_source 810, pa 2,
pag 4, pap 4, phi 4, pipeline 351, pis 4, pl 8, pon 4, pos 46, poz 2, pqe 4, pqw 2, pt 6, question_answering 1,
re 3, relation extraction 2, relation_extraction 4, rn 4, rnd 4, ro 5, roa 4, ru 10, run 4, rw 4, sal 4, sem 4,
sentence_detection 5, sentiment 10, seq2seq 649, sg 4, sit 2, sk 7, sl 3, sla 4, sm 4, sn 4, so 1, sq 4, srn 2

Description Page



How to Use

How to use

Python | Scala

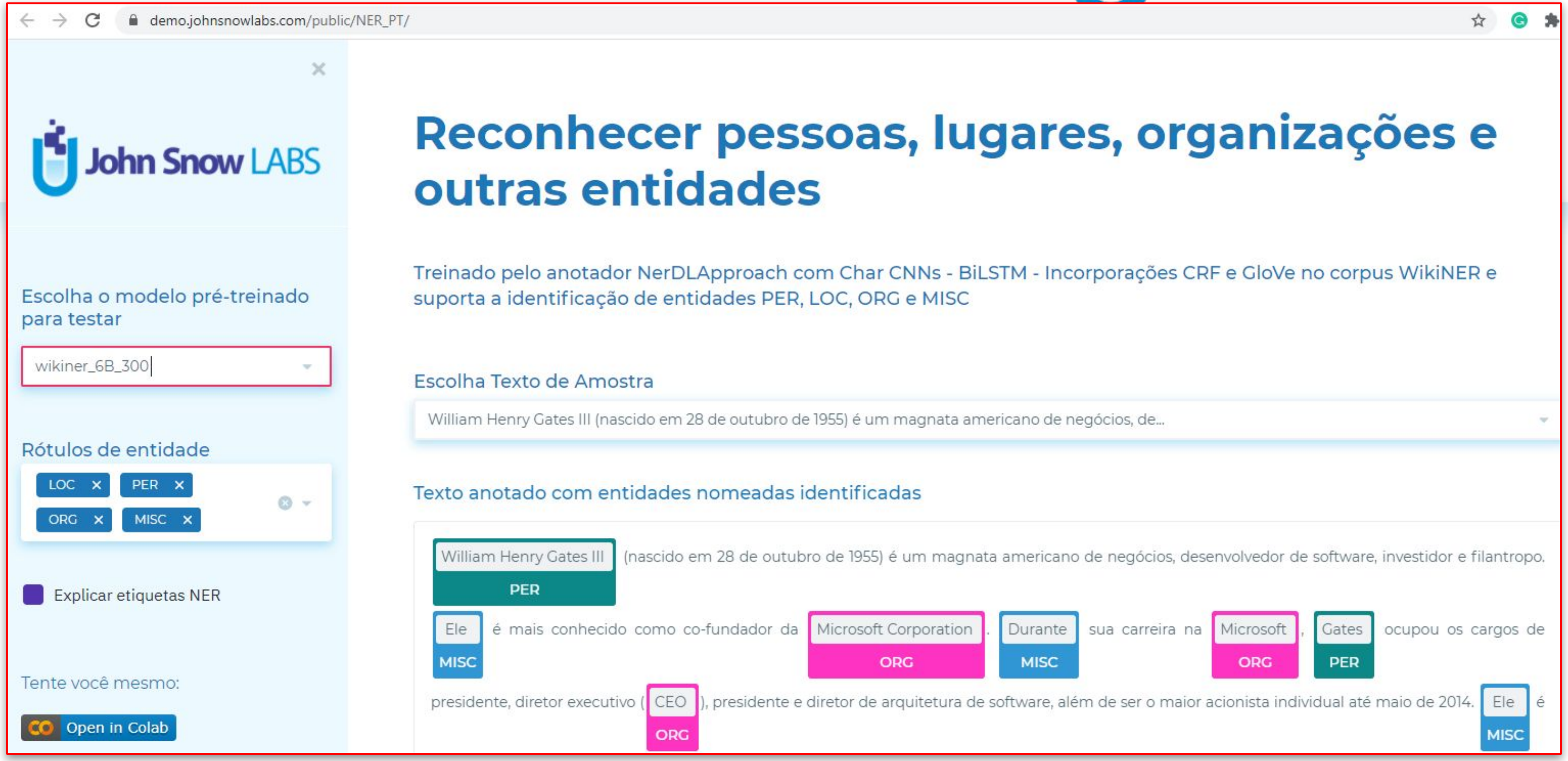
```
...
ner_model = NerDLModel.pretrained("wikiner_6B_300", "pt") \
        .setInputCols(["document", "token", "embeddings"]) \
        .setOutputCol("ner")
...
nlp_pipeline = Pipeline(stages=[document_assembler, sentence_detector, tokenizer, embeddings, ner_model, ner_converter])
pipeline_model = nlp_pipeline.fit(spark.createDataFrame([['']]).toDF('text'))

result = pipeline_model.transform(spark.createDataFrame(pd.DataFrame({'text': ["""William Henry Gates III (nascido em 28 d
```

How to use

Python | Scala

```
...
val ner_model = NerDLModel.pretrained("wikiner_6B_300", "pt")
        .setInputCols(Array("document", "token", "embeddings"))
        .setOutputCol("ner")
...
val pipeline = new Pipeline().setStages(Array(document_assembler, sentence_detector, tokenizer, embeddings, ner_model, ner

val result = pipeline.fit(Seq.empty["William Henry Gates III (nascido em 28 de outubro de 1955) é um magnata americano de
```

Demo Page

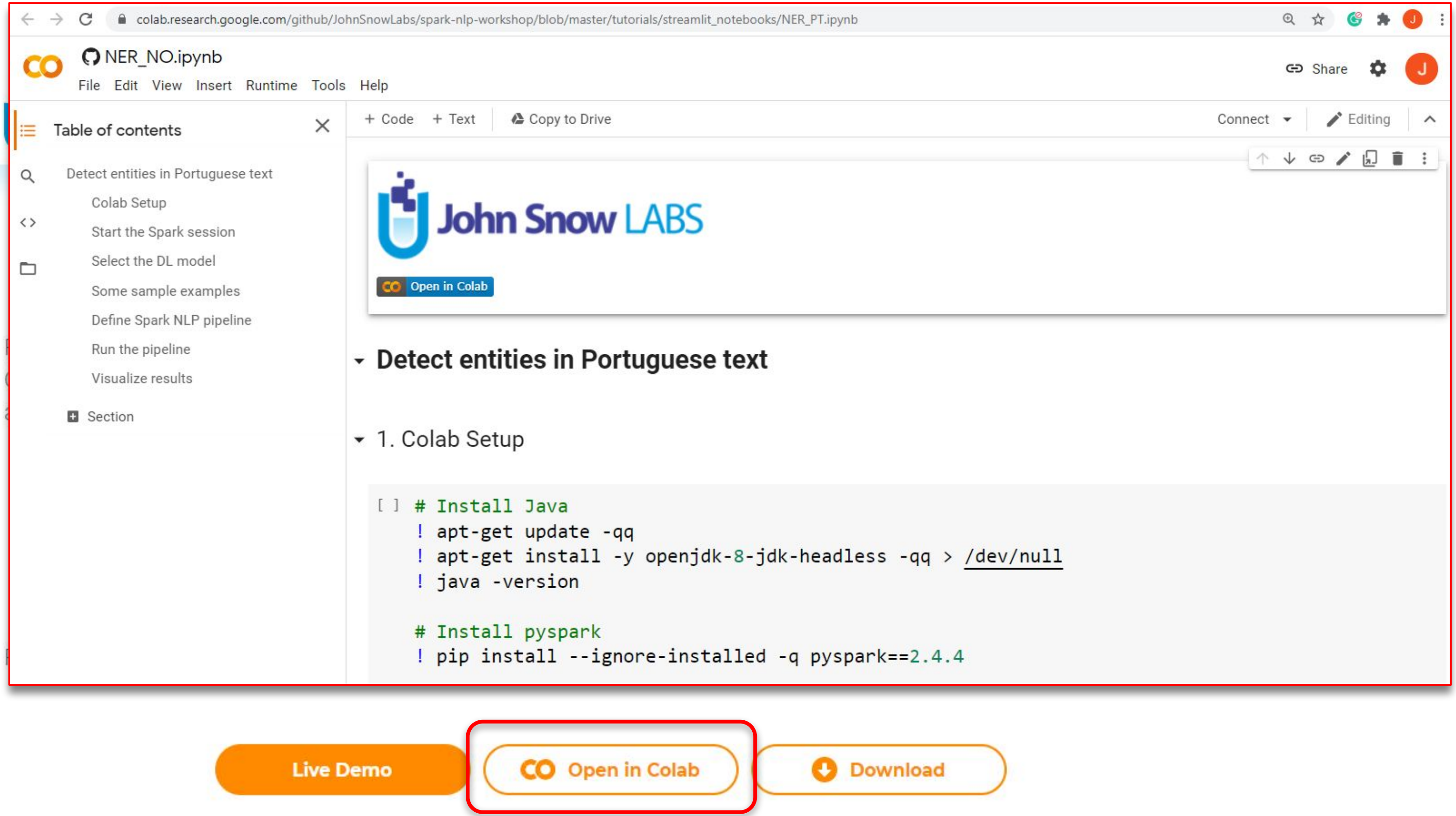


Live Demo

Open in Colab

Download

Colab Notebooks



Benchmarking (Metrics)

Benchmarking

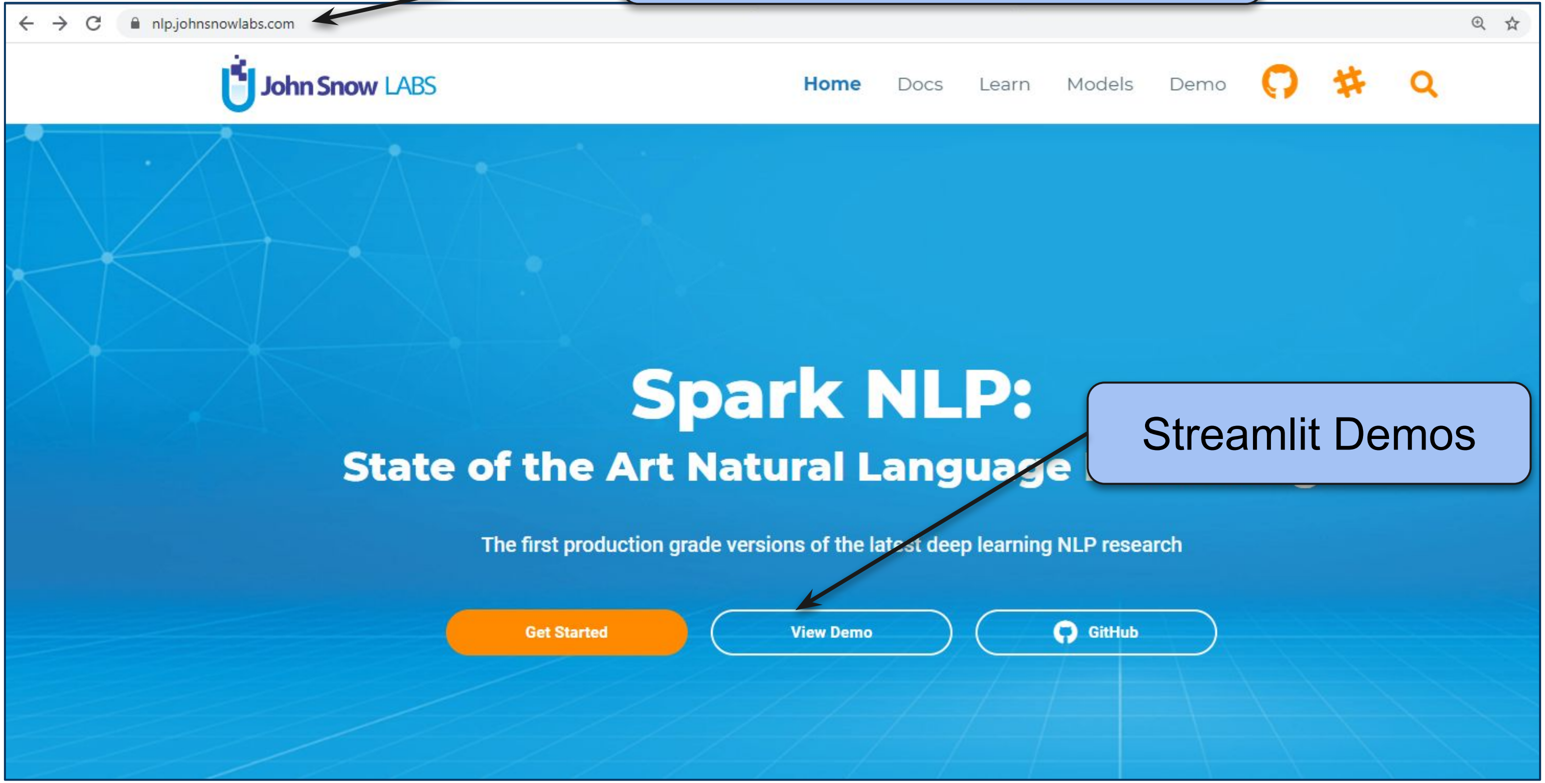
```
label       tp      fp      fn      prec            rec             f1
B-LOC       2081    157     142     0.9298481       0.93612236      0.9329747
I-ORG       1292    220     152     0.8544974       0.8947368       0.87415427
I-LOC       293     81      66      0.78342247      0.81615597      0.79945433
I-PER       1578    127     99      0.9255132       0.940966        0.9331757
B-ORG       1846    185     145     0.9089119       0.9271723       0.91795135
B-PER       3043    186     206     0.942397        0.93659586      0.9394875
tp: 10133 fp: 956 fn: 810 labels: 6
Macro-average    prec: 0.890765, rec: 0.9086249, f1: 0.8996063
Micro-average    prec: 0.91378844, rec: 0.9259801, f1: 0.9198439
```

Streamlit Demos

- Open Source
- Languages
- Healthcare
- Spark OCR
- De-Identification

Main Page

http://nlp.johnsnowlabs.com/



Streamlit Demos

Streamlit Demos

Open Source

FREE **Open Source** | FREE Languages | Healthcare | Spark OCR | De-identification

Open Source

Recognize entities in text

Live Demo

Colab Netbook

Recognize more entities in text

Live Demo

Colab Netbook

Classify documents

Live Demo

Colab Netbook

Analyze sentiment in movie reviews and tweets

Live Demo

Colab Netbook

Languages

FREE Open Source | FREE **Languages** | Healthcare | Spark OCR | De-identification

Languages

Detect language

Live Demo

Colab Netbook

Recognize entities in English text

Live Demo

Colab Netbook

Recognize entities in French text

Live Demo

Colab Netbook

Recognize entities in German text

Live Demo

Colab Netbook

Healthcare

FREE Open Source | FREE Languages | **Healthcare** | Spark OCR | De-identification

Healthcare

Detect signs and symptoms

Live Demo

Colab Netbook

Detect diagnosis and procedures

Live Demo

Colab Netbook

Detect drugs and prescriptions

Live Demo

Colab Netbook

Detect risk factors

Live Demo

Colab Netbook

Spark OCR

FREE Open Source | FREE Languages | Healthcare | **Spark OCR** | De-identification

Spark OCR

PDF to Text

Live Demo

Colab Netbook

DICOM to Text

Live Demo

Colab Netbook

Image to Text

Live Demo

Colab Netbook

Remove background noise from scanned documents

Live Demo

Colab Netbook

De-Identification

FREE
Open Source

FREE
Languages

Healthcare

Spark OCR

De-identification

De-identification

Deidentify structured data

Live Demo

Colab Netbook

Deidentify free text documents

Live Demo

Colab Netbook

Deidentify DICOM documents

Live Demo

Colab Netbook

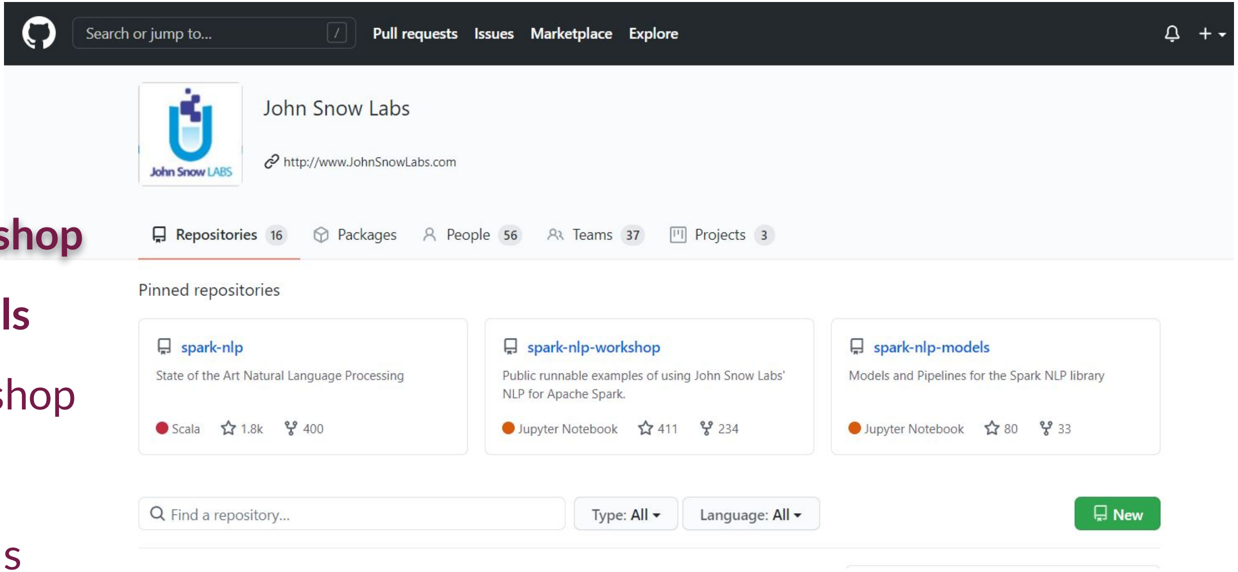
De-identify PDF documents - HIPAA Compliance

Live Demo

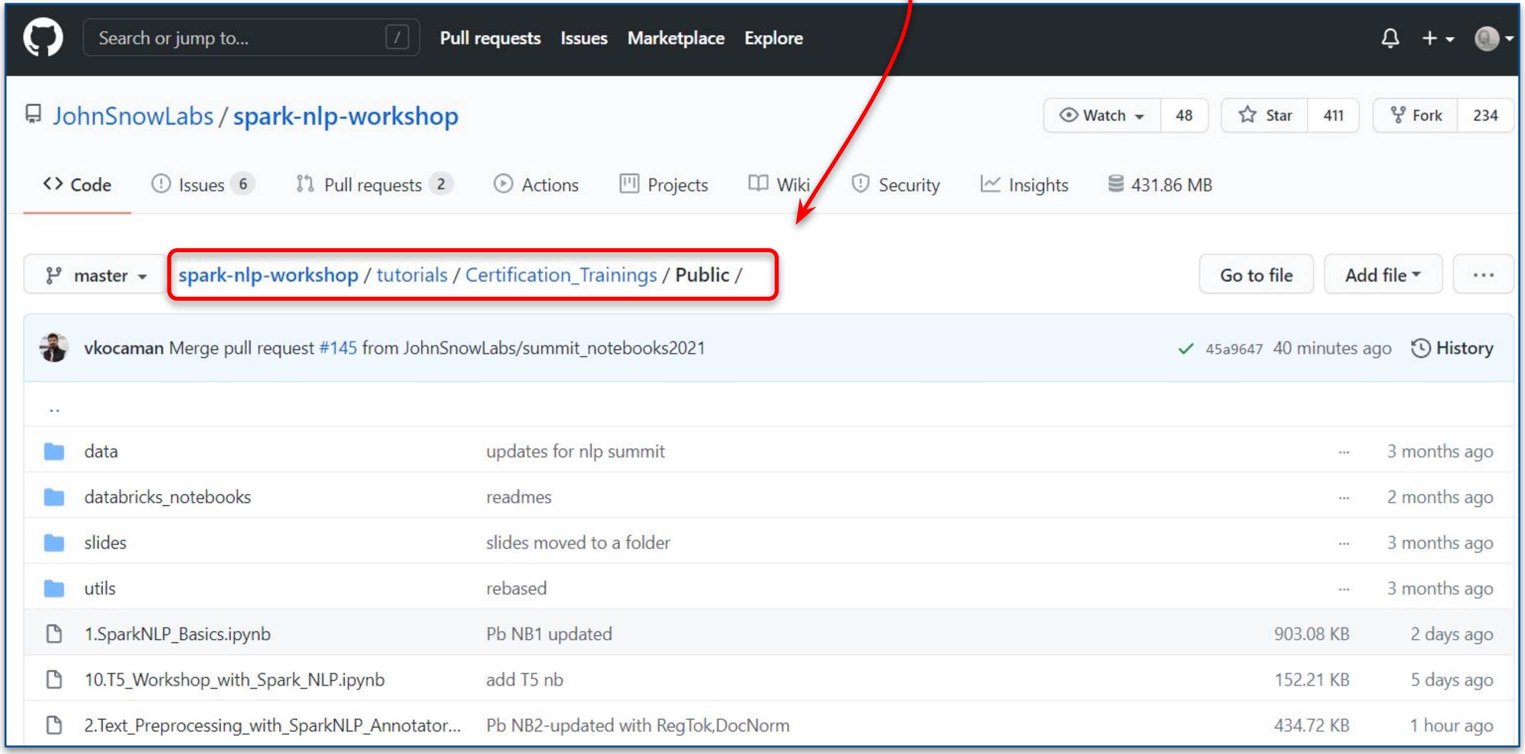
Colab Netbook

GitHub Repositories

- spark-nlp
- spark-nlp-workshop
- spark-nlp-models
- spark-ocr-workshop
- nlu
- spark-nlp-models
- spark-nlp-display

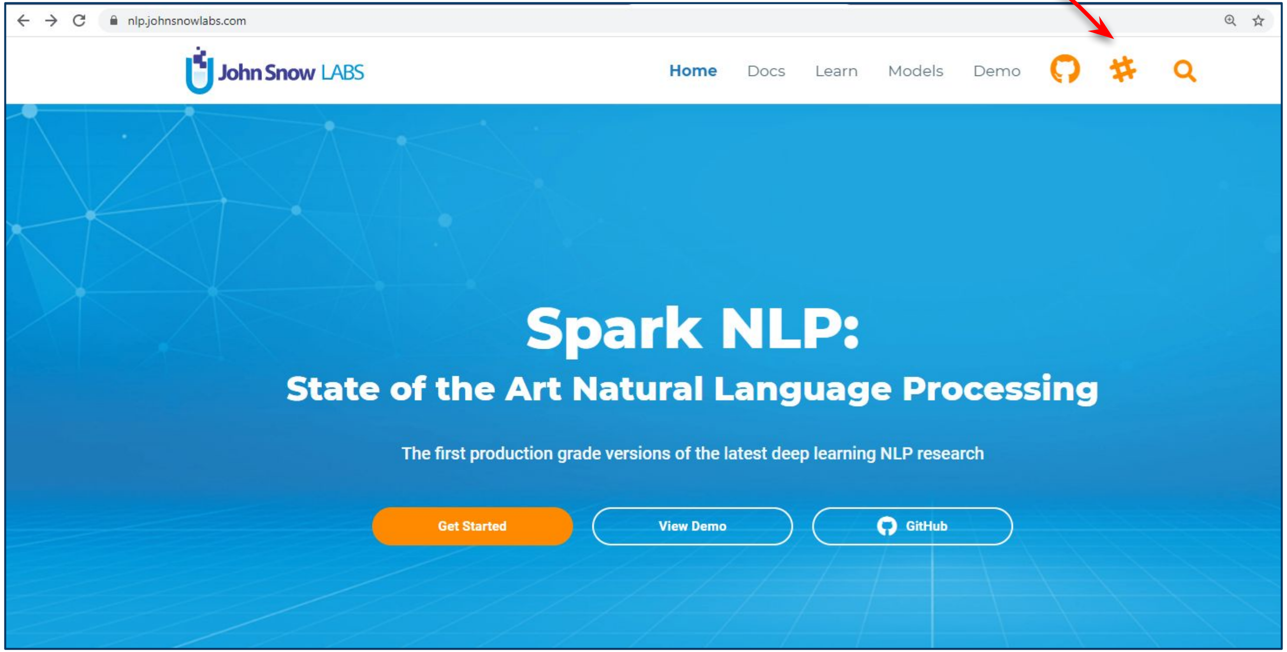


Certification Training



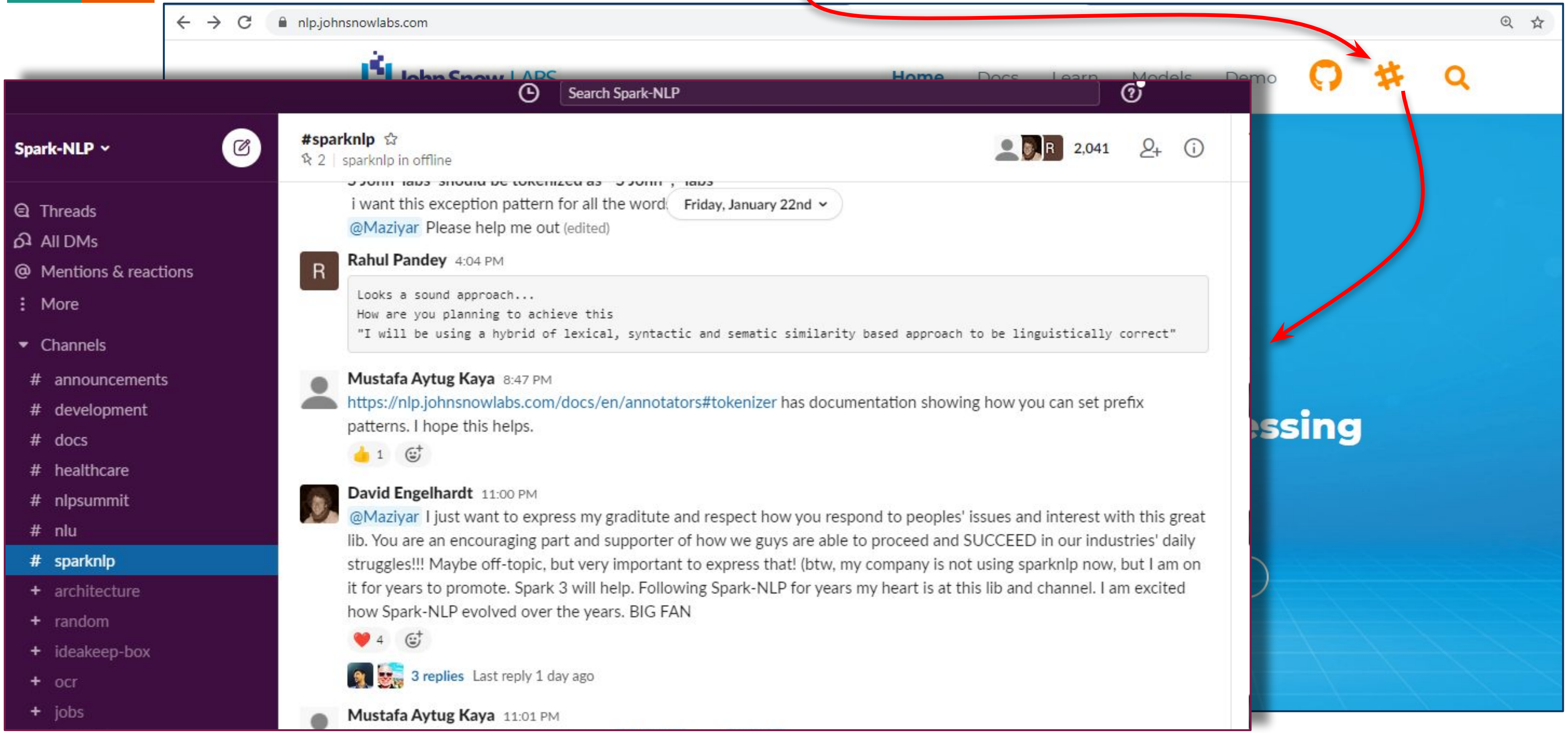
Spark-NLP Slack Channels

- #sparknlp
- #healthcare
- #nlu
- #ocr
- #jobs

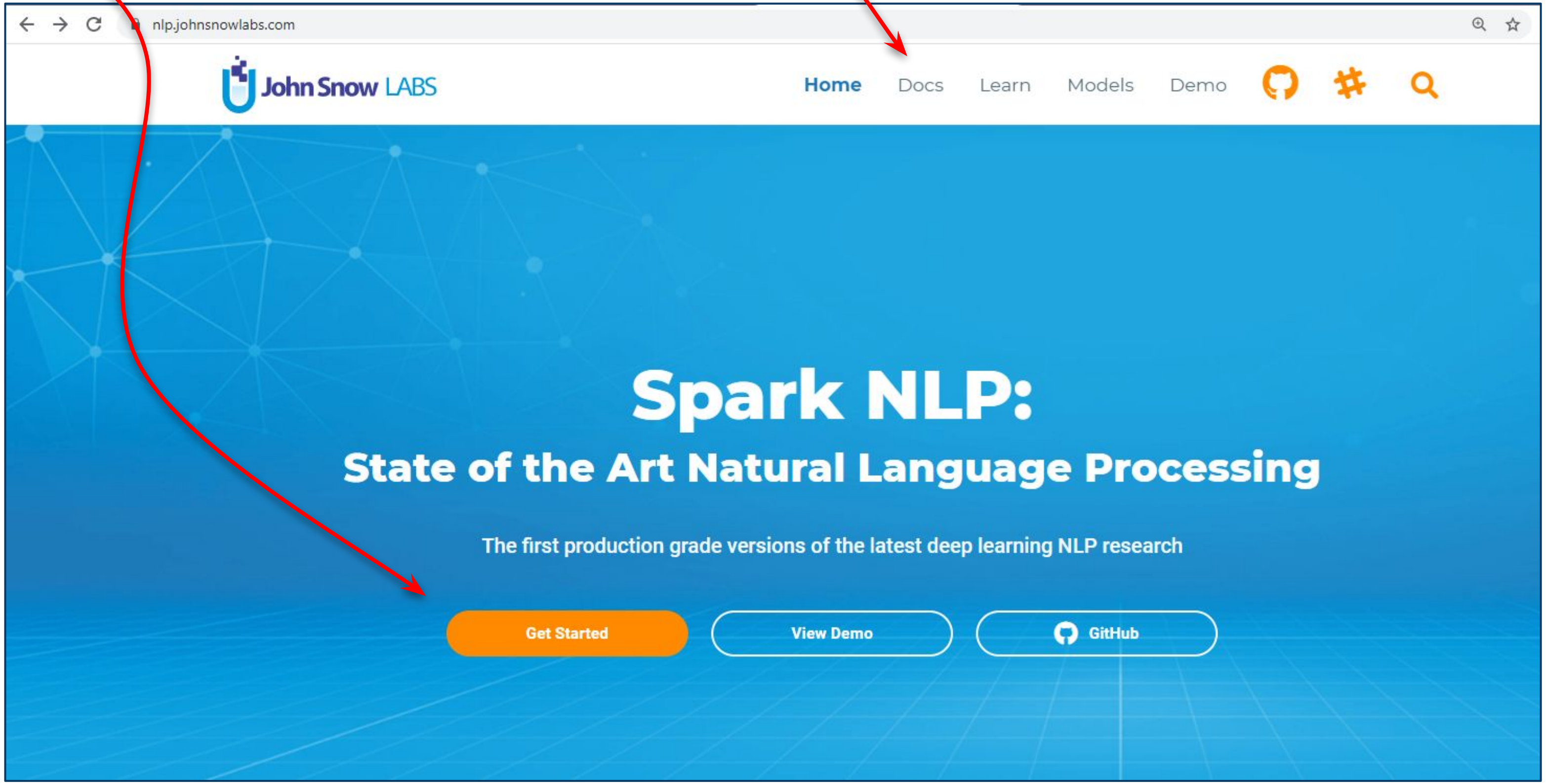


Spark-NLP Slack Channels

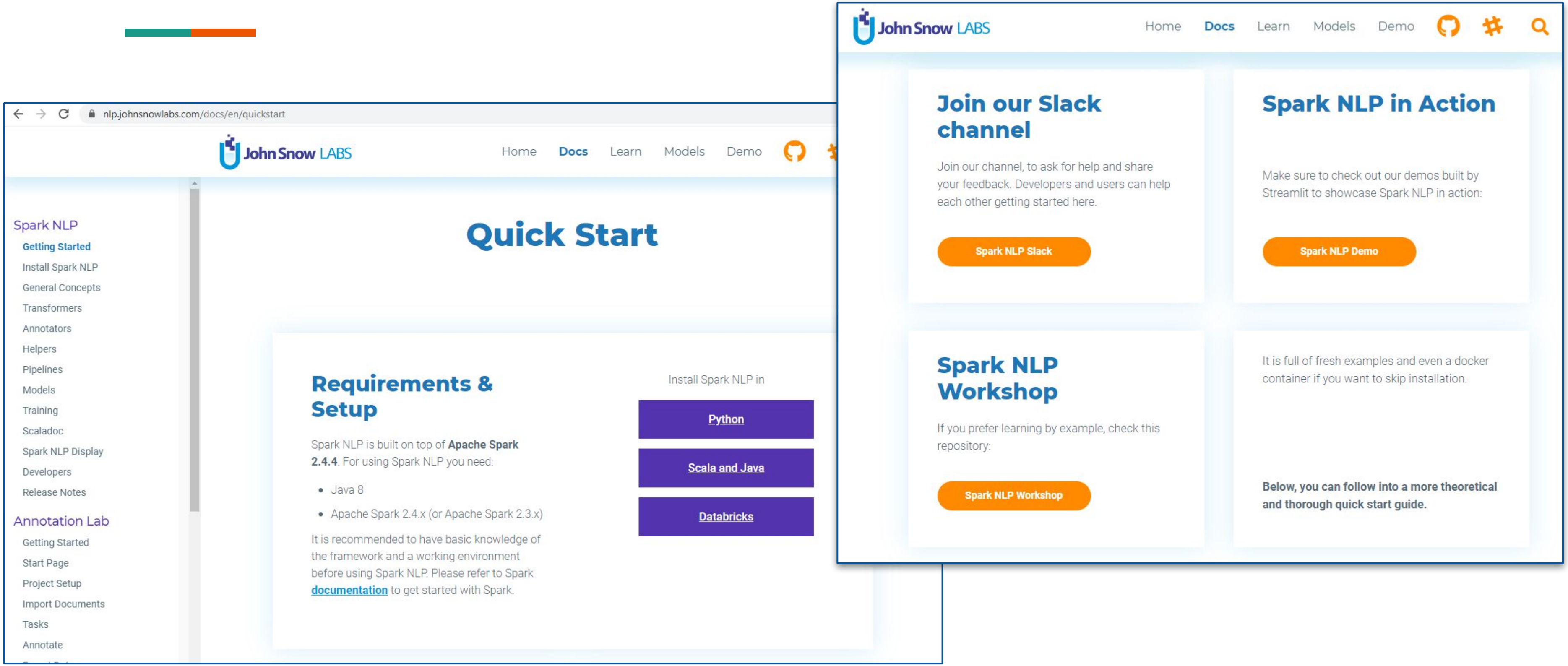
spark-nlp.slack.com



Docs


nlp.johnsnowlabs.com
John Snow LABS
Home
Docs
Learn
Models
Demo
Spark NLP:
State of the Art Natural Language Processing
The first production grade versions of the latest deep learning NLP research
Get Started
View Demo
GitHub

Docs

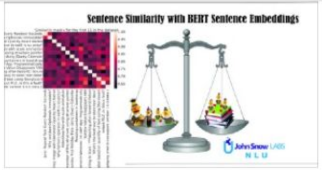


nlp.johnsnowlabs.com/docs/en/quickstart
John Snow LABS
Home
Docs
Learn
Models
Demo
Spark NLP
Getting Started
Install Spark NLP
General Concepts
Transformers
Annotators
Helpers
Pipelines
Models
Training
Scaladoc
Spark NLP Display
Developers
Release Notes
Annotation Lab
Getting Started
Start Page
Project Setup
Import Documents
Tasks
Annotate
Quick Start
Requirements & Setup
Spark NLP is built on top of Apache Spark 2.4.4. For using Spark NLP you need:
Java 8
Apache Spark 2.4.x (or Apache Spark 2.3.x)
It is recommended to have basic knowledge of the framework and a working environment before using Spark NLP. Please refer to Spark documentation to get started with Spark.
Install Spark NLP in
Python
Scala and Java
Databricks
John Snow LABS
Home
Docs
Learn
Models
Demo
Join our Slack channel
Join our channel, to ask for help and share your feedback. Developers and users can help each other getting started here.
Spark NLP Slack
Spark NLP in Action
Make sure to check out our demos built by Streamlit to showcase Spark NLP in action:
Spark NLP Demo
Spark NLP Workshop
If you prefer learning by example, check this repository:
Spark NLP Workshop
It is full of fresh examples and even a docker container if you want to skip installation.
Below, you can follow into a more theoretical and thorough quick start guide.

Articles-Blog Posts

- Medium-Spark NLP (`https://medium.com/spark-nlp`)
- Blog Posts

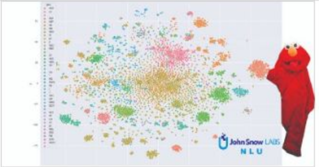
Medium-Spark NLP



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

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

Christian Kasim Loan
Nov 20, 2020 · 8 min read



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

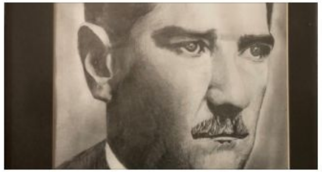
Including Part of Speech, Named Entity Recognition, Emotion, and Sentiment Classification in the same line! With Bonus t-SNE plots and...

Christian Kasim Loan
Oct 24, 2020 · 7 min read

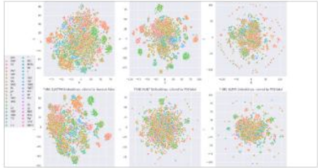
Turkish NER Model Training using Spark NLP, with the help of NLU.

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

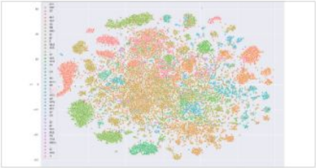
Murat Gunay
Oct 24, 2020 · 10 min read



Classification of Texts Written in Turkish Language Using Spark NLP



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



1 line to BERT Word Embeddings with NLU

Including Part of Speech, Named Entity

Medium

Blog Posts

an intro article for Spark NLP:

https://towardsdatascience.com/introduction-to-spark-nlp-foundations-and-basic-components-part-i-c83b7629ed59

How to start Spark NLP in 2 weeks:

https://towardsdatascience.com/how-to-get-started-with-sparknlp-in-2-weeks-cb47b2ba994d

https://towardsdatascience.com/how-to-wrap-your-head-around-spark-nlp-a6f6a968b7e8

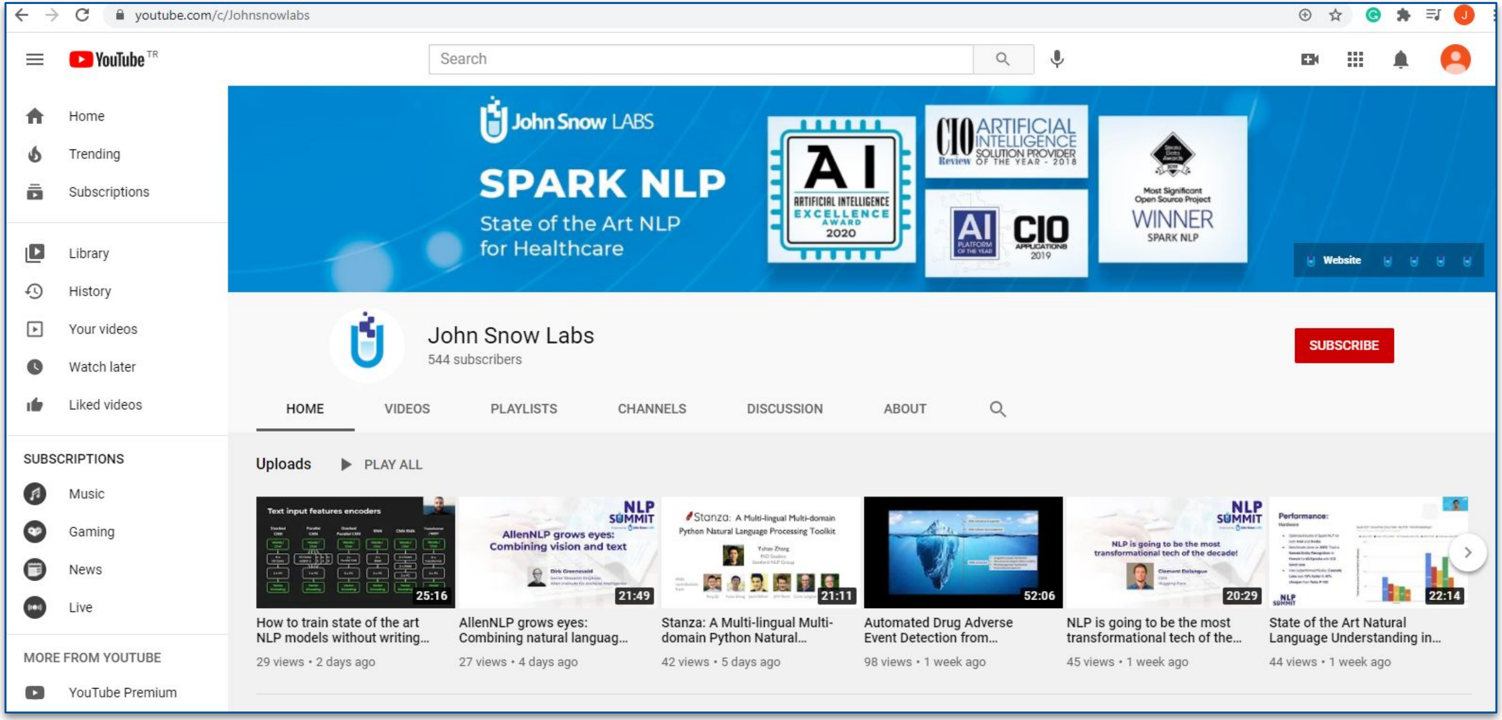
Article for NER and text classification in Spark NLP

https://towardsdatascience.com/named-entity-recognition-ner-with-bert-in-spark-nlp-874df20d1d77

https://medium.com/spark-nlp/named-entity-recognition-for-healthcare-with-sparknlp-nerdl-and-nercrf-a7751b6ad571

https://towardsdatascience.com/text-classification-in-spark-nlp-with-bert-and-universal-sentence-encoders-e644d618ca32

Youtube Channel



```
annotated_d[20]['ner_chunk']
```

```
[Annotation(chunk, 13, 56, 2080-06-14       Johnsonville Family Clinic, {'entity': 'DATE', 'sentence': '0', 'chunk': '0'}),
 Annotation(chunk, 64, 115, Main Street                OSBORNE, MATTISON  Oroville, {'entity': 'LOCATION', 'sentence': '0', 'chu
nk': '1'}),
 Annotation(chunk, 118, 201, AL  89389                    48423663  (468) 429-7459                    06/14/2080, {'entity':
'LOCATION', 'sentence': '0', 'chunk': '2'}),
 Annotation(chunk, 254, 264, 71-year-old, {'entity': 'AGE', 'sentence': '0', 'chunk': '3'}),
 Annotation(chunk, 1109, 1116, February, {'entity': 'DATE', 'sentence': '0', 'chunk': '4'}),
 Annotation(chunk, 1761, 1772, Willie Knapp, {'entity': 'NAME', 'sentence': '0', 'chunk': '5'}),
 Annotation(chunk, 1856, 1863, 06/17/80, {'entity': 'DATE', 'sentence': '0', 'chunk': '6'}),
 Annotation(chunk, 1870, 1877, 06/17/80, {'entity': 'DATE', 'sentence': '0', 'chunk': '7'}),
 Annotation(chunk, 1884, 1891, 06/14/80, {'entity': 'DATE', 'sentence': '0', 'chunk': '8'})]
```

```
annotated_d[20]['clinical_ner']
```

```
[Annotation(named_entity, 0, 5, O, {'word': 'Record', 'confidence': '1.0'}),
 Annotation(named_entity, 7, 10, O, {'word': 'date', 'confidence': '1.0'}),
 Annotation(named_entity, 11, 11, O, {'word': ':', 'confidence': '1.0'}),
 Annotation(named_entity, 13, 22, B-DATE, {'word': '2080-06-14', 'confidence': '0.998'}),
 Annotation(named_entity, 31, 42, I-LOCATION, {'word': 'Johnsonville', 'confidence': '0.824'}),
 Annotation(named_entity, 44, 49, I-LOCATION, {'word': 'Family', 'confidence': '0.9086'}),
 Annotation(named_entity, 51, 56, I-LOCATION, {'word': 'Clinic', 'confidence': '0.9936'}),
 Annotation(named_entity, 59, 62, O, {'word': '5388', 'confidence': '0.5029'}),
 Annotation(named_entity, 64, 67, B-LOCATION, {'word': 'Main', 'confidence': '0.8511'}),
 Annotation(named_entity, 69, 74, I-LOCATION, {'word': 'Street', 'confidence': '0.9542'}),
 Annotation(named_entity, 89, 95, I-NAME, {'word': 'OSBORNE', 'confidence': '0.6801'}),
 Annotation(named_entity, 96, 96, I-NAME, {'word': ',', 'confidence': '0.9632'}),
 Annotation(named_entity, 98, 105, I-NAME, {'word': 'MATTISON', 'confidence': '0.8746'}),
 Annotation(named_entity, 108, 115, I-NAME, {'word': 'Oroville', 'confidence': '0.6039'}),
 Annotation(named_entity, 116, 116, O, {'word': ',', 'confidence': '0.9902'}),
 Annotation(named_entity, 118, 119, B-LOCATION, {'word': 'AL', 'confidence': '0.9959'}),
 Annotation(named_entity, 122, 126, I-LOCATION, {'word': '89389', 'confidence': '0.6723'}),
 Annotation(named_entity, 147, 154, I-ID, {'word': '48423663', 'confidence': '0.965'}),
 Annotation(named_entity, 157, 157, I-CONTACT, {'word': '(', 'confidence': '0.939'}),
 Annotation(named_entity, 158, 160, I-CONTACT, {'word': '468', 'confidence': '0.9925'}),
 Annotation(named_entity, 161, 161, I-CONTACT, {'word': ')', 'confidence': '0.8479'}),
 Annotation(named_entity, 163, 170, I-CONTACT, {'word': '429-7459', 'confidence': '0.7224'}),
 Annotation(named_entity, 192, 201, I-DATE, {'word': '06/14/2080', 'confidence': '0.967'}),
 Annotation(named_entity, 208, 214, O, {'word': 'HISTORY', 'confidence': '1.0'}),
```