



Spark NLP

VIRTUAL TRAINING

Day 1: January 25, 2022 – 9:00 to 13:00 PST

Day 2: January 26, 2022 – 9:00 to 13:00 PST



Spark NLP

for Healthcare Data Scientists

Day 1: January 27, 2022 – 9:00 to 13:00 PST

Day 2: January 28, 2022 – 9:00 to 13:00 PST



Christian Kasim Loan
Senior Data Scientist
christian@johnsnowlabs.com

4 days, 16 hrs live training

Certification exams in 2 weeks

Veysel Kocaman
Lead Data Scientist
veysel@johnsnowlabs.com



Experts



Spark NLP for Data Scientists

January 25 -26, 2022

Christian Kasim Loan
Senior Data Scientist
christian@johnsnowlabs.com



Welcome - We have a lot of things ahead of us

Day-1	60 min	<ul style="list-style-type: none">- Intro to John Snow Labs, Spark NLP and NLP Theory- Spark NLP Pretrained Pipelines basics and JVM/Spark concepts
	10 min	Break
	50 min	<ul style="list-style-type: none">- Text Preprocessing and composing custom pipeline in Spark NLP
	10 min	Break
	50 min	<ul style="list-style-type: none">- Showcase of the 4000+ pretrained Models for 200+ languages in Spark NLP
		Break
	50 min	<ul style="list-style-type: none">- 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	<ul style="list-style-type: none">- Unsupervised Keyword Extraction with YAKE- Sentence Detection- Spell Checking- Graph triplet extraction
	10 min	Break
	60 min	<ul style="list-style-type: none">- 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	<ul style="list-style-type: none">- 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	<ul style="list-style-type: none">- 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
Sr. Data Scientist
christian@johnsnowlabs.com

John Snow Labs NLP Documentation



Spark NLP



Healthcare NLP



Spark OCR



Annotation Lab



NLP Server



NLU

John Snow Labs NLP Documentation



Spark NLP



Healthcare NLP



Spark OCR



Annotation Lab



NLP Server



NLU

Setup

pip install spark-nlp

RUNNING CODE:

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

[How to set up Google Colab]

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

 Open in Colab

```
[ ] !wget -q https://raw.githubusercontent.com/JohnSnowLabs/spark-nlp-workshop/master/colab_setup.sh  
!bash colab_setup.sh -p 2.4.4 -s 2.7.5  
# -p is for pyspark  
# -s is for spark-nlp  
# by default they are set to the latest  
  
setup Colab for PySpark 2.4.4 and Spark NLP 2.7.5  
|██████████| 215.7MB 78kB/s  
|██████████| 143KB 53.1MB/s  
|██████████| 204KB 42.7MB/s  
Building wheel for pyspark (setup.py) ... done
```

master		spark-nlp-workshop / tutorials / Certification_Trainings / Public /	Go to file	Add file	...
	vkocaman	Merge pull request #147 from JohnSnowLabs/summit_notebooks2021	✓	f6858ae	22 hours ago
		...			
	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-LangDete 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

Introducing Spark NLP

Total downloads

14,558,265

Total downloads - 30 days

1,371,781

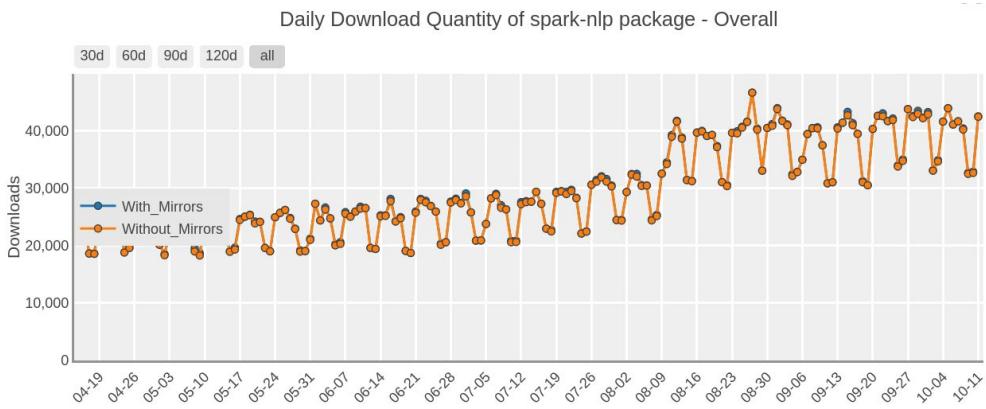
Total downloads - 7 days

318,804

downloads 14M

downloads/month 1M

downloads/week 318k

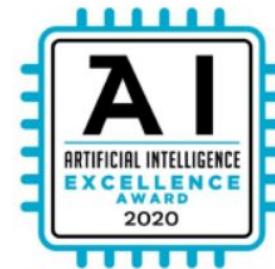


<https://medium.com/spark-nlp/introduction-to-spark-nlp-foundations-and-basic-components-part-i-c83b7629ed59>



- 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.)

Recognized by the Technology Experts



TRUSTED BY





NLP SUMMIT

HEALTHCARE

FREE & ONLINE

April 6th - 7th

Register



OFFICIALLY SUPPORTED RUNTIMES

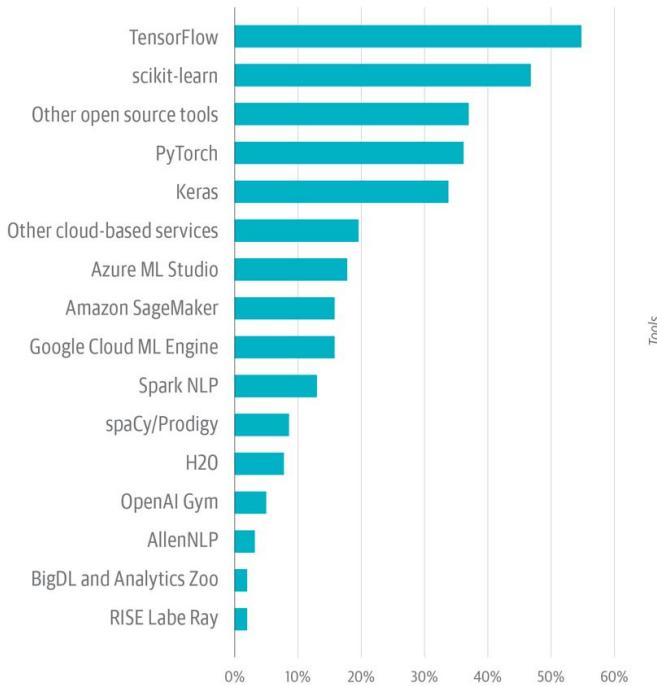


Azure

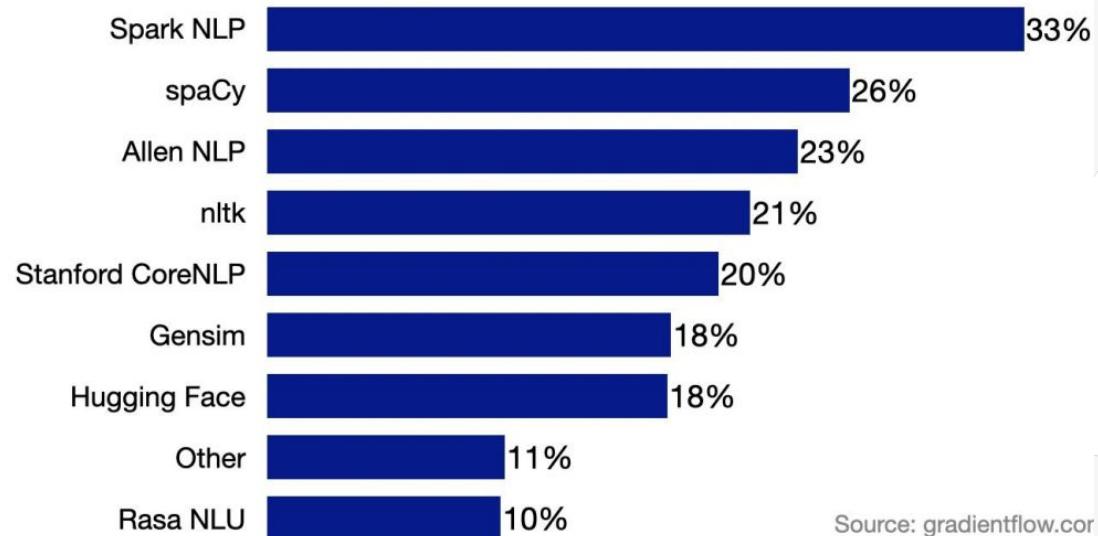


Spark NLP in Industry

Which of the following AI tools do you use?



Which NLP libraries does your organization use?



Source: gradientflow.co

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

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

Abstract

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

Anonymous NAACL-HLT 2021 submission

Spark NLP: Natural Language Understanding at Scale

Veysel Kocaman, David Talby

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

Accurate Clinical and Biomedical Named Entity Recognition at Scale

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

Veysel Kocaman, David Talby

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

Abstract

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

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

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

Spark NLP Modules

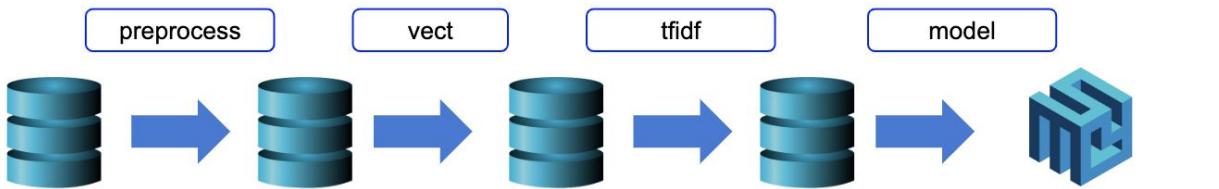
Clinical Entity Recognition	Clinical Entity Linking	Assertion Status	Relation Extraction	
40 units DOSAGE of insulin glargine DRUG at night FREQUENCY	Suspect diabetes SNOMED-CT: 473127005 Lisinopril 10 MG RxNorm: 316151 Hyponatremia ICD-10: E873	Fever and sore throat → PRESENT No stomach pain → ABSENT Father with Alzheimer → FAMILY	AFTER Admitted for nausea Symptom due to chemo Occurrence Treatment CAUSED BY	
Algorithms				
Extract Knowledge <ul style="list-style-type: none"> Entity Linker Entity Disambiguator Document Classifier Contextual Parser 	De-Identity Text <ul style="list-style-type: none"> Structured Data Unstructured Text Obfuscator Generalizer 	Medical Transformers <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>JSL-BERT-Clinical</p> </div> <div style="text-align: center;"> <p>BioBERT</p> </div> <div style="text-align: center;"> <p>ClinicalBERT</p> </div> <div style="text-align: center;"> <p>GloVe-Med</p> </div> <div style="text-align: center;"> <p>GloVe-ICD-O</p> </div> <div style="text-align: center;"> <p>BlueBert</p> </div> </div>	Linked Medical Terminologies <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>SNOMED-CT</p> </div> <div style="text-align: center;"> <p>CPT</p> </div> <div style="text-align: center;"> <p>UMLS</p> </div> <div style="text-align: center;"> <p>ICD-10-CM</p> </div> <div style="text-align: center;"> <p>RxNorm</p> </div> <div style="text-align: center;"> <p>HPO</p> </div> <div style="text-align: center;"> <p>ICD-10-PCS</p> </div> <div style="text-align: center;"> <p>ICD-O</p> </div> <div style="text-align: center;"> <p>LOINC</p> </div> </div>	
Split Text <ul style="list-style-type: none"> Sentence Detector Deep Sentence Detector Tokenizer nGram Generator 	Clean Medical Text <ul style="list-style-type: none"> Spell Checking Spell Correction Normalizer Stopword Cleaner 	200+ Pretrained Models <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> Clinical: Signs, Symptoms, Treatments, Procedures, Tests, Labs, Sections </div> <div style="width: 45%;"> Anatomy: Organ, Subdivision, Cell, Structure, Organism, Tissue, Gene, Chemical </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> Drugs: Name, Dosage, Strength, Route, Duration, Frequency, Poisons, Adverse Effects </div> <div style="width: 45%;"> Demographics: Age, Gender, Height, Weight, Race, Ethnicity, Marital Status, Vital Signs </div> </div> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> Risk Factors: Smoking, Obesity, Diabetes, Hypertension, Substance Abuse </div> <div style="width: 45%;"> Sensitive Data: Patient Name, Address, Phone, Email, Dates, Providers, Identifiers </div> </div>		
Clinical Grammar <ul style="list-style-type: none"> Stemmer Lemmatizer Part of Speech Tagger Dependency Parser 	Find in Text <ul style="list-style-type: none"> Text Matcher Regex Matcher Date Matcher Chunker 			
Trainable & Tunable	Scalable to a Cluster	Fast Inference	Hardware Optimized	Community

Entity Recognition	Information Extraction	Spelling & Grammar	Text Classification
I love Lucy PERSON	They met Last week DATE → 29-04-2020	abc She became the first... → She became the first	
Translation	Summarization	Question Answering	Emotion Detection
Split Text	Clean Text	4000+	200+
<ul style="list-style-type: none"> Sentence Detector Deep Sentence Detector Tokenizer nGram Generator Word Segmentation 	<ul style="list-style-type: none"> Spell Checking Spell Correction Normalizer Stopword Cleaner Summarization 	Pre-trained Pipelines, Models & Transformers	Languages
		<div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>BERT</p> </div> <div style="text-align: center;"> <p>ELMO</p> </div> <div style="text-align: center;"> <p>GloVe</p> </div> </div> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>ALBERT</p> </div> <div style="text-align: center;"> <p>XLNet</p> </div> <div style="text-align: center;"> <p>USE</p> </div> </div> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Small BERT</p> </div> <div style="text-align: center;"> <p>ELECTRA</p> </div> </div> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>T5</p> </div> <div style="text-align: center;"> <p>NMT</p> </div> <div style="text-align: center;"> <p>LaBSE</p> </div> </div> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>DistilBERT</p> </div> <div style="text-align: center;"> <p>RoBERTa</p> </div> </div> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>XLM-RoBERTa</p> </div> </div> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>S-BERT</p> </div> <div style="text-align: center;"> <p>XLING</p> </div> </div>	
Understand Grammar	Find in Text	Trainable & Tunable	Scalable to a Cluster
<ul style="list-style-type: none"> Stemmer Lemmatizer Part of Speech Tagger Dependency Parser Translation 	<ul style="list-style-type: none"> Text Matcher Regex Matcher Date Matcher Chunker Question Answering 		
Scalable to a Cluster	Fast Inference	Hardware Optimized	Community

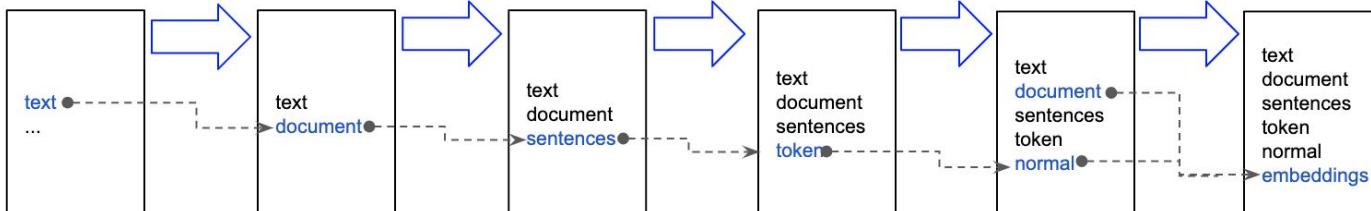
Spark NLP Modules (Enterprise and Public)

Introducing Spark NLP

Pipeline of annotators



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



DataFrame

```
from pyspark.ml import Pipeline
documentAssembler = 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=[documentAssembler,
sentenceDetector,
tokenizer,
normalizer,
word_embeddings,
])
nlpPipeline.fit(df).transform(df)
```

Introducing Spark NLP



Faster inference

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

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

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

Coding ...

1. Spark NLP Basics

2. Text Preprocessing with Spark NLP

3. Spark NLP Pretrained Models

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

https://github.com/JohnSnowLabs/spark-nlp-workshop/blob/master/tutorials/Certification_Trainings/Public/1.SparkNLP_Basics.ipynb

[https://github.com/JohnSnowLabs/spark-nlp-workshop/blob/master/tutorials/Certification_Trainings/Public/2.Text_Preprocessing_with_SparkNLP\(Annotators_Transformers\).ipynb](https://github.com/JohnSnowLabs/spark-nlp-workshop/blob/master/tutorials/Certification_Trainings/Public/2.Text_Preprocessing_with_SparkNLP(Annotators_Transformers).ipynb)

https://github.com/JohnSnowLabs/spark-nlp-workshop/blob/master/tutorials/Certification_Trainings/Public/3.SparkNLP_Pretrained_Models.ipynb

Spark NLP Basics and Pretrained Pipelines

- Colab Setup
- How to prevent Google Colab from disconnecting?
- Start Spark Session
- Using Pretrained Pipelines
 - Explain Document ML
 - Explain Document DL
 - Recognize Entities DL
 - Clean Stop Words
 - Clean Slang
 - Spell Checker
 - Spell Checker DL
 - Parsing a list of texts
 - Using fullAnnotate to get more details
 - Use pretrained match_chunk Pipeline for Individual Noun Phrase
 - Extract exact dates from referential date phrases
 - Sentiment Analysis
 - Vivek algo
 - DL version (trained on imdb)
 - DL version (trained on twitter dataset)

Text Preprocessing with Spark NLP

- Colab Setup
- Annotators and Transformer Concepts
- Create Spark Dataframe
- Transformers
- Document Assembler
- Sentence Detector
- Tokenizer
- Stacking Spark NLP Annotators in Spark ML Pipeline
- Normalizer
- Stopwords Cleaner
- Token Assembler
- Stemmer
- Lemmatizer
- NGram Generator
- TextMatcher
- RegexMatcher
- Text Cleaning with UDF
- Finisher
- LightPipeline

Spark NLP Pretrained Models

- Colab Setup
- LemmatizerModel
- PerceptronModel (POS - Part of speech tags)
- Chunker
- Dependency Parser
- SpellChecker
 - Norvig Spell Checker
 - Context SpellChecker
- Language Detector
- Embeddings
 - Word Embeddings (Glove)
 - Using your own Word embeddings in Spark NLP
- Elmo Embeddings
- Bert Embeddings
- Albert Embeddings
- XlnetEmbeddings
- Chunk Embeddings
- UniversalSentenceEncoder
- Loading Models from local
- Getting Sentence Embeddings from word embeddings
 - Cosine similarity between two embeddings (sentence similarity)
- NERDL Model
 - Public NER (CoNLL 2003)
 - NerDL OntoNotes 100D
 - NER with Bert (Onto)
 - Getting the NER chunks with NER Converter
- Highlight the entities

Natural Language Processing

Information Retrieval

Doc A



Doc 1

Doc 2

Doc 3

Sentiment Analysis



Information Extraction



Machine Translation



Question Answering



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
study ing	- ing	study
niñ as	- as	niñ
niñ ez	- 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

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

Stopwords

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

(520 stopwords)

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!

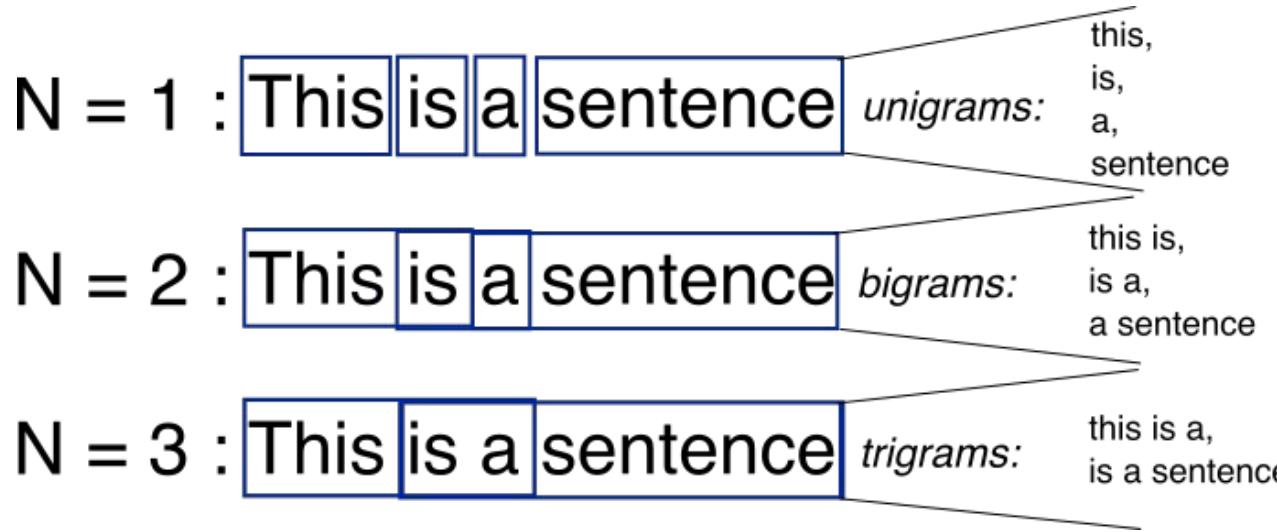
Spark NLP also comes with a Slang normalizer:

Original tweet
@USER, r u cuming 2 MidCorner dis Sunday?

Normalized tweet

@USER, are you coming to MidCorner this Sunday?

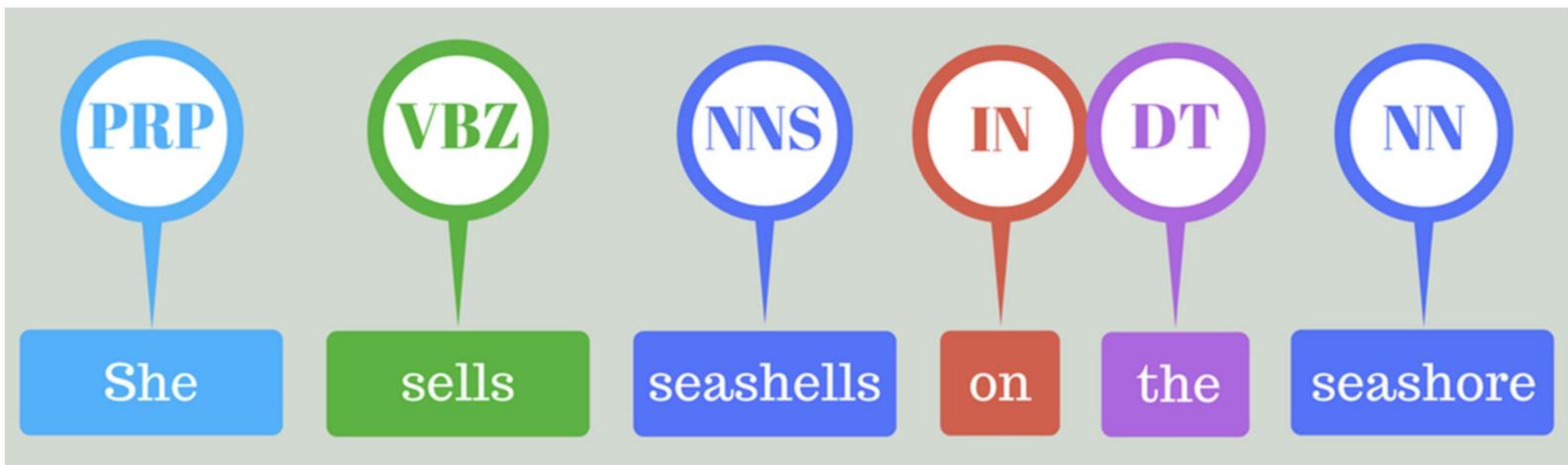
N-gram Tokenization



- Kind of tokenizers which split words or sentences into several tokens
- Each token has certain number of 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.**

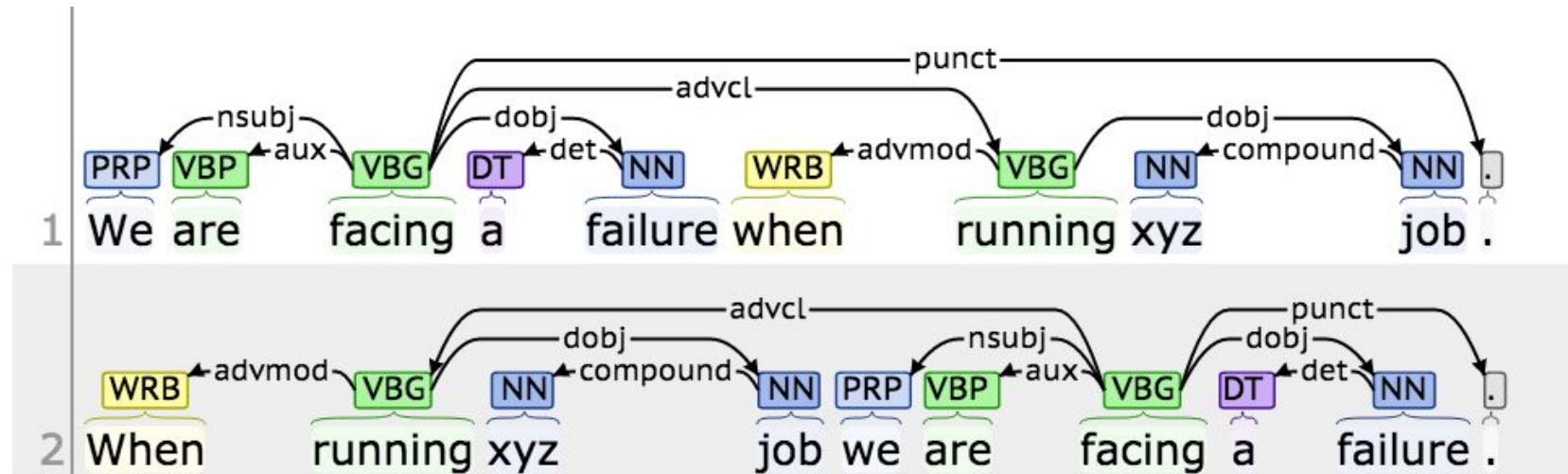


People	
Companies	
Places	
Money	
Links	foo.com , acme.org , bar.com
Phone #s	1-800-12345, +353-44-34-254
	...

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

DEPENDENCY PARSING

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



Part - I (Day 1) - Coding Time

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

Spark NLP
for Data Scientists



Christian Kasim Loan
Sr. Data Scientist
christian@johnsnowlabs.com

Part - II (Day 1)

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

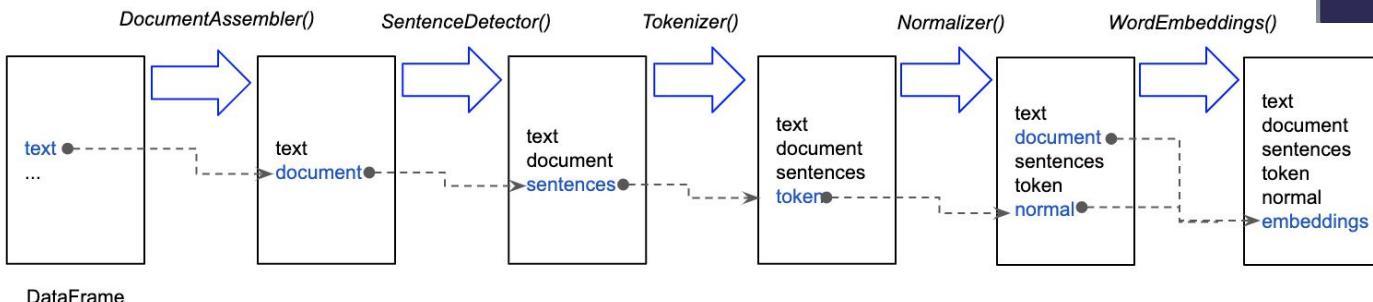
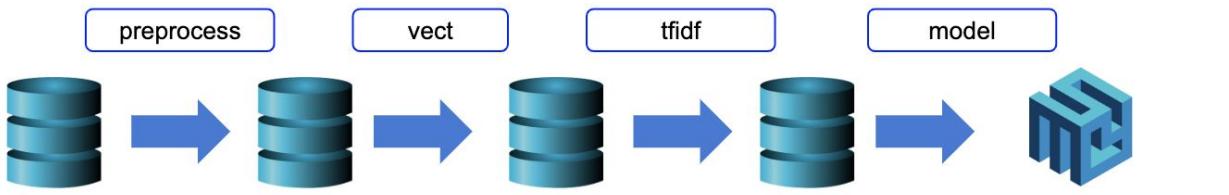
Spark NLP
for Data Scientists



Christian Kasim Loan
Sr. 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
Sr. 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
Sr. Data Scientist
christian@johnsnowlabs.com

Spark NLP Modules

Clinical Entity Recognition	Clinical Entity Linking	Assertion Status	Relation Extraction	
40 units DOSAGE of insulin glargine DRUG at night FREQUENCY	Suspect diabetes SNOMED-CT: 473127005 Lisinopril 10 MG RxNorm: 316151 Hyponatremia ICD-10: E87.3	Fever and sore throat → PRESENT No stomach pain → ABSENT Father with Alzheimer → FAMILY	AFTER Admitted for nausea due to chemo Occurrence Symptom Treatment CAUSED BY	
Algorithms				
Extract Knowledge <ul style="list-style-type: none"> Entity Linker Entity Disambiguator Document Classifier Contextual Parser 		De-Identity Text <ul style="list-style-type: none"> Structured Data Unstructured Text Obfuscator Generalizer 		
Split Text <ul style="list-style-type: none"> Sentence Detector Deep Sentence Detector Tokenizer nGram Generator 		Clean Medical Text <ul style="list-style-type: none"> Spell Checking Spell Correction Normalizer Stopword Cleaner 		
Clinical Grammar <ul style="list-style-type: none"> Stemmer Lemmatizer Part of Speech Tagger Dependency Parser 		Find in Text <ul style="list-style-type: none"> Text Matcher Regex Matcher Date Matcher Chunker 		
Trainable & Tunable	Scalable to a Cluster	Fast Inference	Hardware Optimized	Community

Entity Recognition	Information Extraction	Spelling & Grammar	Text Classification		
I love Lucy PERSON	They met Last week DATE → 29-04-2020	abc She became the first... → She became the first			
Translation		Summarization			
[je t'aime → I love you]					
Split Text		Clean Text			
<ul style="list-style-type: none"> Sentence Detector Deep Sentence Detector Tokenizer nGram Generator Word Segmentation 		<ul style="list-style-type: none"> Spell Checking Spell Correction Normalizer Stopword Cleaner Summarization 			
Understand Grammar		Find in Text			
<ul style="list-style-type: none"> Stemmer Lemmatizer Part of Speech Tagger Dependency Parser 		<ul style="list-style-type: none"> Text Matcher Regex Matcher Date Matcher Chunker Question Answering 			
Trainable & Tunable		Scalable to a Cluster	Fast Inference	Hardware Optimized	Community

Spark NLP Modules (Enterprise and Public)



200+

Languages

Word & Sentence Embeddings

Vocabulary

index: Word:

0 aardvark
1 able
...

2409 black
2410 bling
...

3202 candid

3203 cast

3204 cat

...

5281 is

5282 island

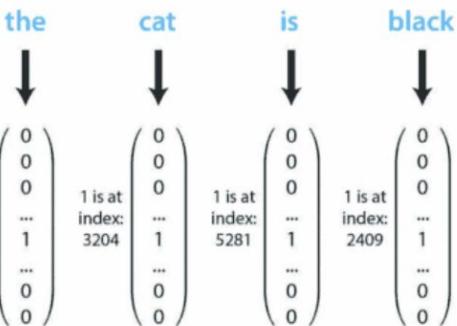
...

8676 the

8677 thing

...

9999 zombie



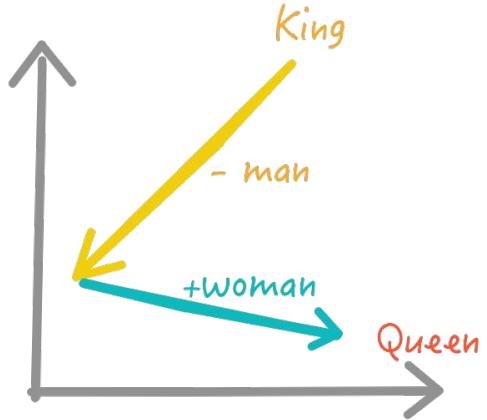
One-hot vector encoding for words in input sentence complete.

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 ,
 0.00211152,  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



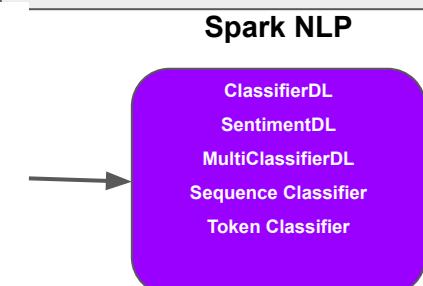
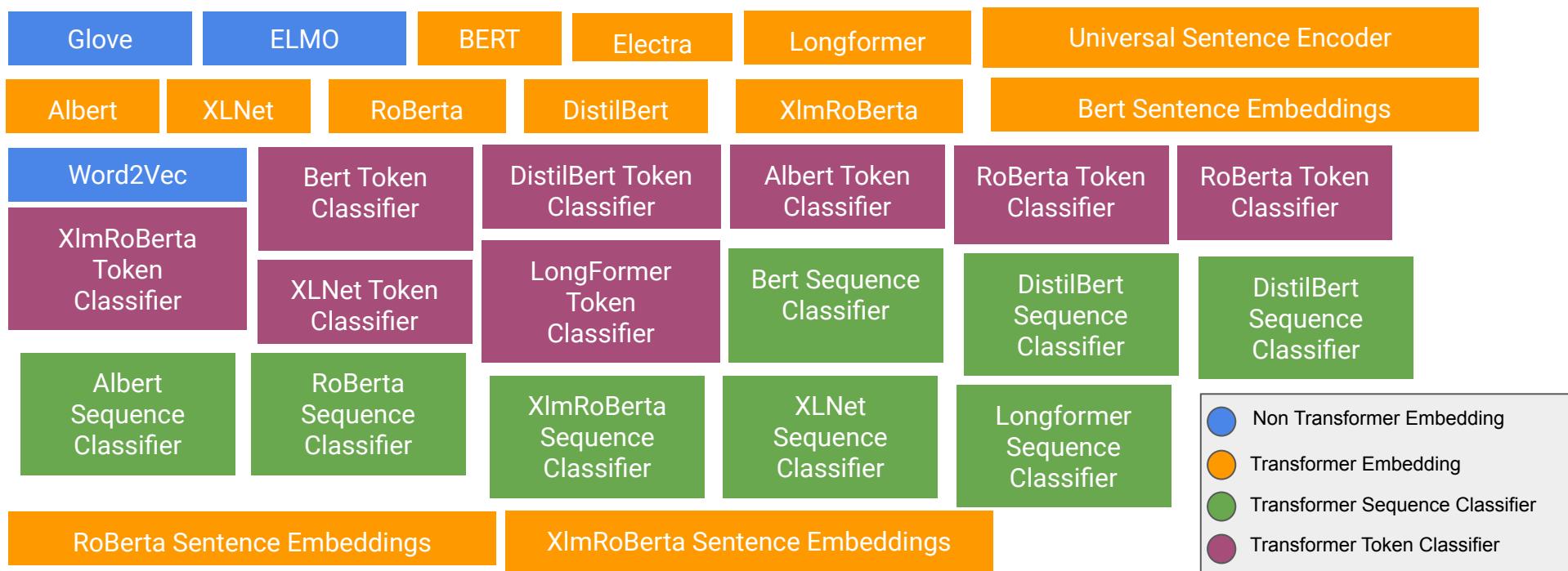
b) Similar Words have same angles

```
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



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

Embeddings

in

Spark NLP

Part 1

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

Language	Spark NLP Model Name	Model
Number		
51	en	sent_small_bert_L12_128
52	en	BertSentenceEmbeddings
53	en	sent_small_bert_L12_256
54	en	BertSentenceEmbeddings
55	en	sent_small_bert_L12_512
56	en	BertSentenceEmbeddings
57	en	sent_small_bert_L2_128
58	en	BertSentenceEmbeddings
59	en	sent_small_bert_L2_256
60	en	BertSentenceEmbeddings
61	en	sent_small_bert_L2_512
62	en	BertSentenceEmbeddings
63	en	sent_small_bert_L2_768
64	en	BertSentenceEmbeddings
65	en	sent_small_bert_L4_128
66	en	BertSentenceEmbeddings
67	en	sent_small_bert_L4_256
68	en	BertSentenceEmbeddings
69	en	sent_small_bert_L4_512
70	en	BertSentenceEmbeddings
71	en	sent_small_bert_L4_768
72	en	BertSentenceEmbeddings
73	en	sent_small_bert_L6_128
74	en	BertSentenceEmbeddings
75	en	sent_small_bert_L6_256
76	en	BertSentenceEmbeddings
77	en	sent_small_bert_L6_512
78	en	BertSentenceEmbeddings
79	en	sent_small_bert_L6_768
80	en	BertSentenceEmbeddings
81	en	sent_small_bert_L8_128
82	en	BertSentenceEmbeddings
83	en	sent_small_bert_L8_256
84	en	BertSentenceEmbeddings
85	en	sent_small_bert_L8_512
86	en	BertSentenceEmbeddings
87	en	sent_small_bert_L10_128
88	en	BertSentenceEmbeddings
89	en	sent_small_bert_L10_256
90	en	BertSentenceEmbeddings
91	en	sent_small_bert_L10_512
92	en	BertSentenceEmbeddings
93	en	sent_small_bert_L10_768
94	en	BertSentenceEmbeddings
95	en	covidbert_large_uncased
96	en	BertSentenceEmbeddings
97	en	electra_base_uncased
98	en	BertSentenceEmbeddings
99	en	electra_large_uncased
100	en	BertSentenceEmbeddings

Embeddings

in

Spark NLP

Part 2

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

Embeddings

in

Spark NLP

Part 3

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

Embeddings in Spark NLP Part 4

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

Language	Spark NLP Model Name	Model
Number		
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		
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

Embeddings

in

Spark NLP

Part 5

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_fibert	BertForSequenceClassification
9	ja	bert_sequence_classifier_japanese_sentiment	BertForSequenceClassification
10	xx	bert_sequence_classifier_multilingual_sentiment	BertForSequenceClassification
11	ru	bert_sequence_classifier_rubert_sentiment	BertForSequenceClassification
12	de	bert_sequence_classifier_sentiment	BertForSequenceClassification
13	tr	bert_sequence_classifier_turkish_sentiment	BertForSequenceClassification
14	en	bert_sequence_classifier_question_statement	BertForSequenceClassification
15	en	bert_sequence_classifier_question_statement_cl...	BertForSequenceClassification
16	en	bert_sequence_classifier_antisemitism	BertForSequenceClassification
17	en	bert_sequence_classifier_hatexplain	BertForSequenceClassification
18	en	bert_sequence_classifier_trec_coarse	BertForSequenceClassification
19	en	bert_sequence_classifier_age_news	BertForSequenceClassification
20	en	bert_sequence_classifier_banking77	BertForSequenceClassification
21	en	bert_sequence_classifier_sms_spam	BertForSequenceClassification
22	en	bert_sequence_classifier_song_lyrics	BertForSequenceClassification
23	en	distilbert_base_sequence_classifier_ag_news	DistilBertForSequenceClassification
24	en	distilbert_base_sequence_classifier_amazon_pol...	DistilBertForSequenceClassification

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

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

Transformer Token Classifiers in Spark NLP Part 1

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

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

Word & Sentence 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.

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.

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 pretraining method BERT. Researchers introduced two parameter-reduction techniques in ALBERT to lower memory consumption and increase training speed.

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

Word & Sentence 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.

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

Train in 1 Language, predict in 100+ different languages



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

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

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

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

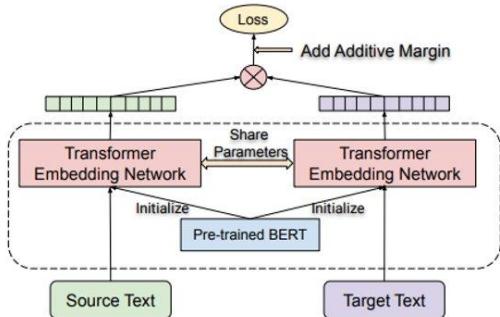


Figure 1: Dual encoder model with BERT based encoding modules.



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

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

... 94 more!

MARIAN NMT

Fast Neural Machine Translation in C++



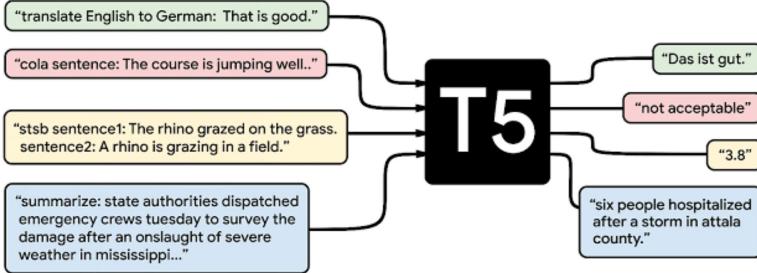
```
# Use ISO standards for the languages
nlu.load('<start_language>.translate_to.<target_language>')

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

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

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

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



```

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

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

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

```

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

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



Every T5 Task with explanation:

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

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



TF Hub to Spark NLP

Spark NLP	TF Hub Notebooks	Colab
BertEmbeddings	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

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



Part - III (Day 1) - Coding Time

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

Spark NLP
for Data Scientists



Christian Kasim Loan
Sr. 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



Christian Kasim Loan
Sr. Data Scientist
christian@johnsnowlabs.com

CoNLL 2003 (English)

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

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

NER-DL in Spark NLP

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

The best NER score in production

93.3 %
Test Set

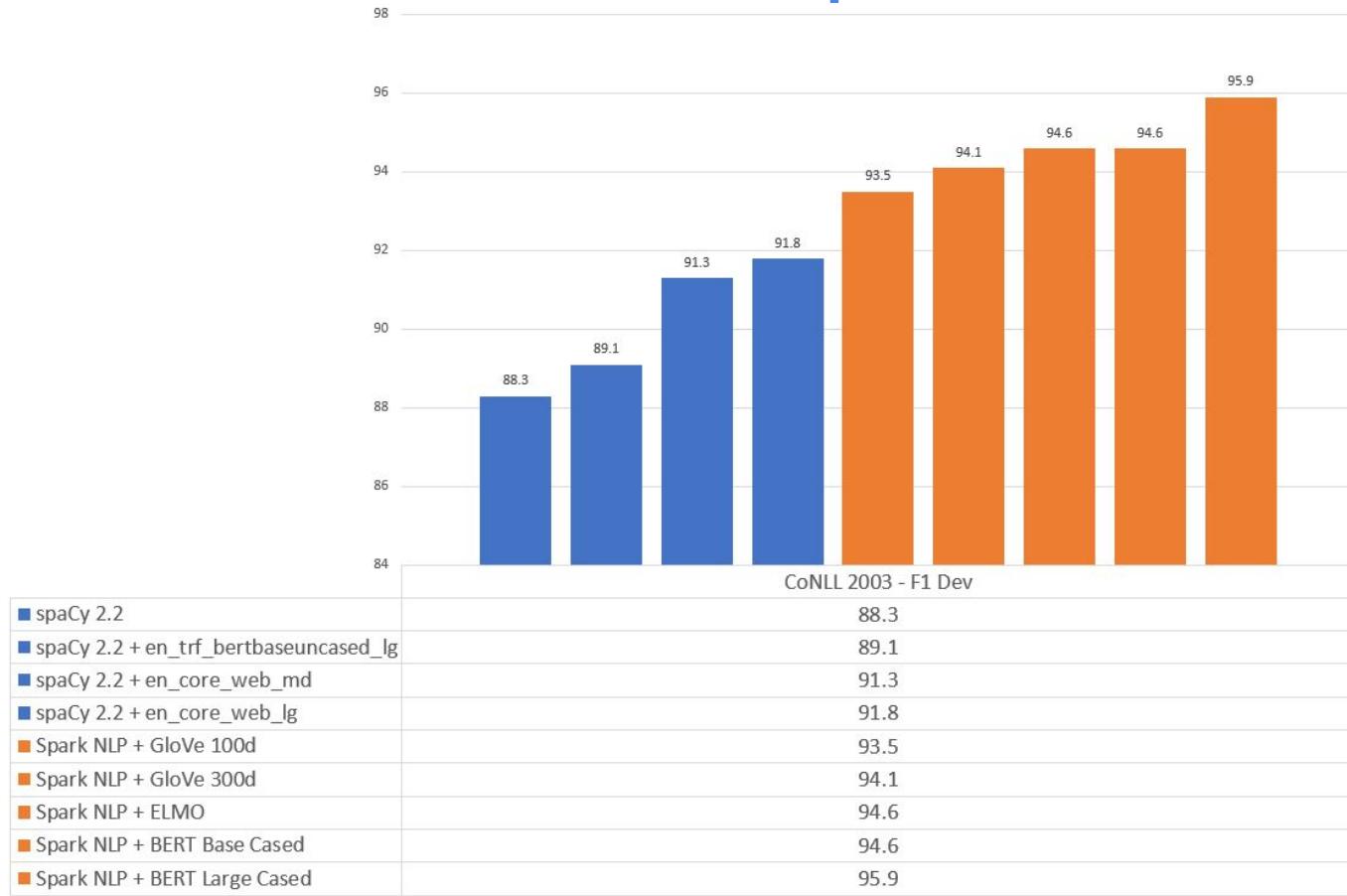


Bert



NerDLApproach

NER-DL in Spark NLP



NER Systems

Feature-engineered machine learning systems	Dict	SP	DU	EN	GE
Carreras et al. (2002) binary AdaBoost classifiers	Yes	81.39	77.05	-	-
Malouf (2002) - Maximum Entropy (ME) + features	Yes	73.66	68.08	-	-
Li et al. (2005) SVM with class weights	Yes	-	-	88.3	-
Passos et al. (2014) CRF	Yes	-	-	90.90	-
Ando and Zhang (2005a) Semi-supervised state of the art	No	-	-	89.31	75.27
Agerri and Rigau (2016)	Yes	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 (SENNNA+Gazetteer)	Yes	-	-	89.59	-
Huang et al. (2015) Bi-LSTM+CRF+ (SENNNA+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)

NER-DL in Spark NLP

Char-CNN-BiLSTM

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

NER-DL in Spark NLP

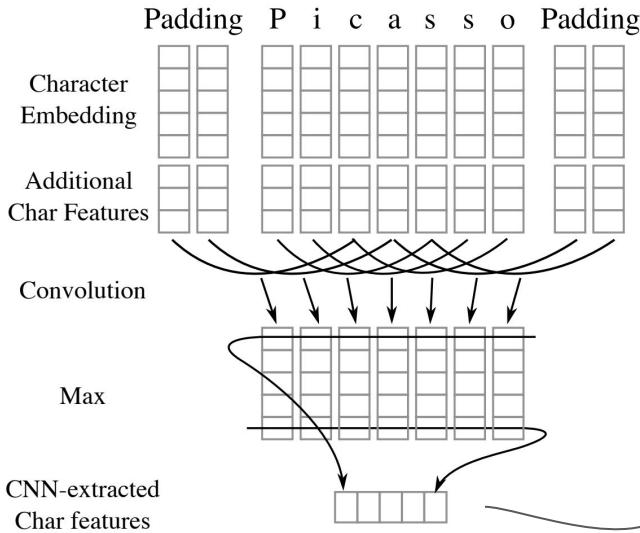


Figure 2: The convolutional neural network extracts character features from each word. The character embedding and (optionally) the character type feature vector are computed through lookup tables. Then, they are concatenated and passed into the CNN.

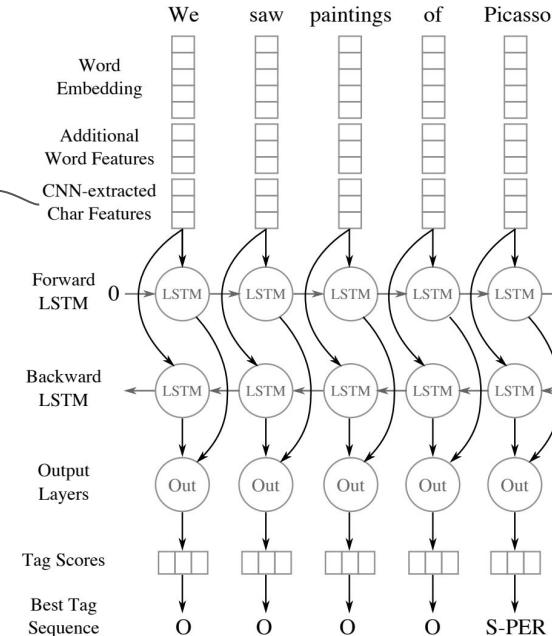


Figure 1: The (unrolled) BLSTM for tagging named entities. Multiple tables look up word-level feature vectors. The CNN (Figure 2) extracts a fixed length feature vector from character-level features. For each word, these vectors are concatenated and fed to the BLSTM network and then to the output layers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

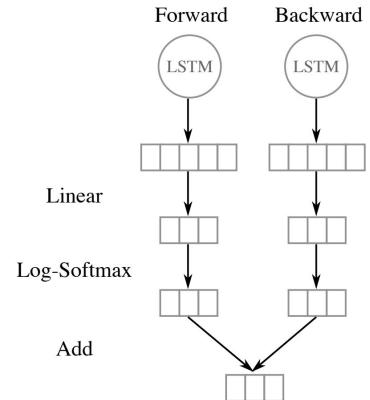


Figure 3: The output layers ("Out" in Figure 1) decode output into a score for each tag category.

Char-CNN-BiLSTM

NER-DL in Spark NLP

CoNLL2003 format

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

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

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

* CoNLL: Conference on Computational Natural Language Learning

BIO schema

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

John Smith ⇒ PERSON
New York ⇒ LOCATION

Coding ...

Open 4. NERDL Training notebook in Colab

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

https://github.com/JohnSnowLabs/spark-nlp-workshop/blob/master/tutorials/Certification_Trainings/Public/4.NERDL_Training.ipynb

Test set evaluation

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

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

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

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

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

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

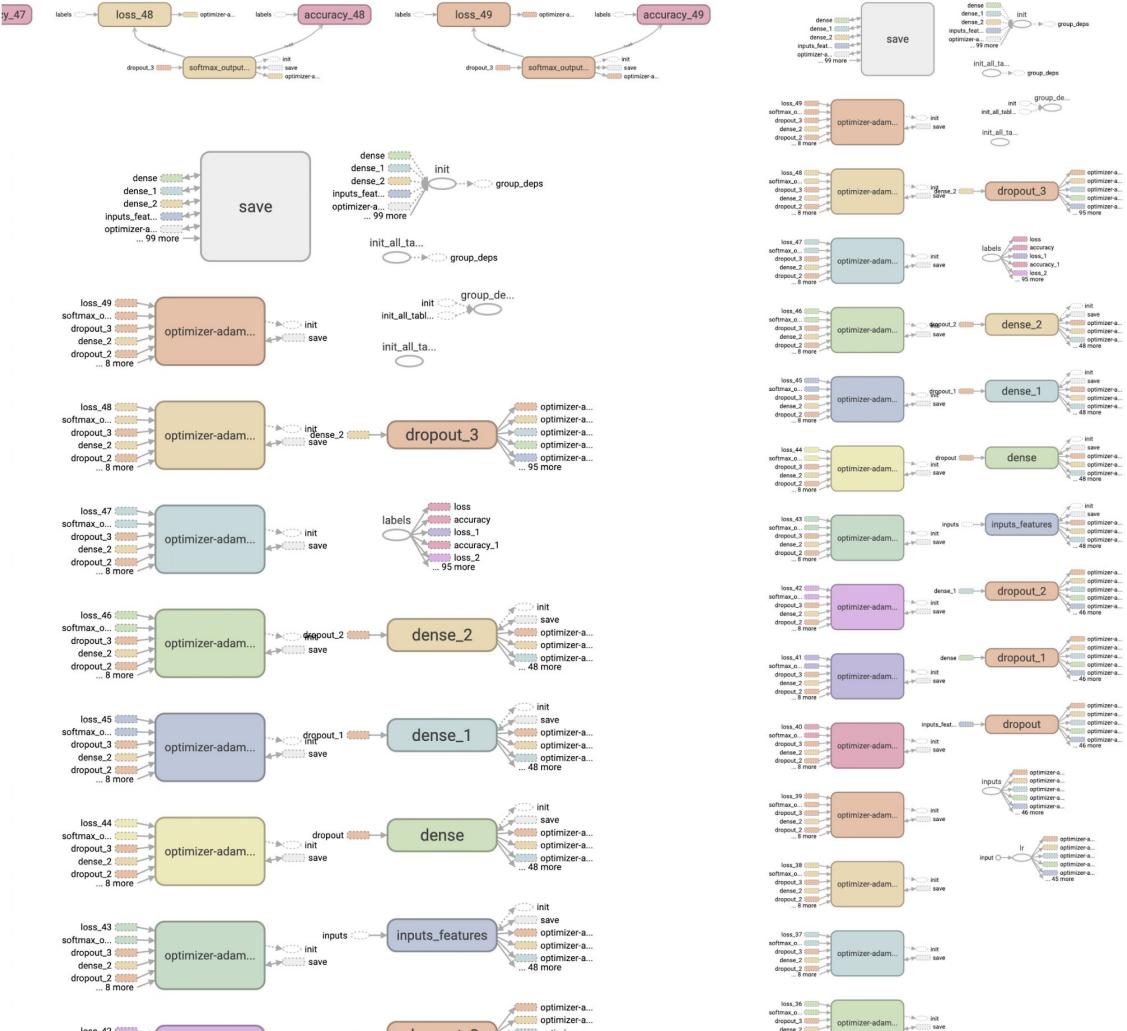
precision    recall   f1-score   support
B-LOC       0.88      0.93      0.90      1837
B-MISC      0.80      0.82      0.81       922
B-ORG       0.92      0.73      0.81      1341
B-PER       0.94      0.95      0.95      1842
```

SentimentDL, ClassifierDL, and MultiClassifierDL

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

- 100 dimensions
 - 200 dimensions
 - 128 dimensions
 - 256 dimensions
 - 300 dimensions
 - 512 dimensions
 - 768 dimensions
 - 1024 dimensions
- tfhub_ues
 - tfhub_use_lg
 - glove_6B_100
 - glove_6B_300
 - glove_840B_300
 - bert_base_cased
 - bert_base_uncased
 - bert_large_cased
 - bert_large_uncased
 - bert_multi_uncased
 - electra_small_uncased
 - elmo
 - ... 100+ Word & Sentence models

Classifier DL Tensorflow Architecture



Upload trained models to Models Hub

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

The screenshot shows a Jupyter Notebook interface with the following details:

- Files:** A sidebar on the left lists files and directories:
 - my_nlp_pipeline (highlighted with a red box)
 - sample_data
 - my_nlp_pipeline.zip
 - my_nlp_pipeline2.zip
 - news_category_test.csv
 - news_category_train.csv
- Code Cell:** The main area contains two code snippets:

```
[ ] # Fit a Spark NLP pipeline
nlp_pipeline = unfitted_pipeline.fit(dataset)
nlp_pipeline.save('my_nlp_pipeline')

[35] # cd into saved dir and zip
! cd /content/my_nlp_pipeline ; zip -r my_nlp_pipeline.zip *
```
- Context Menu:** A context menu is open over the "my_nlp_pipeline.zip" file in the file list, with "Download" highlighted (also with a red box). Other options include "Rename file", "Delete file", "Copy path", and "Refresh".
- Output:** Below the code cell, the terminal output shows the process of creating the zip file, listing various embeddings and their storage paths.

Part - IV (Day 1) - Coding Time

- ❖ [Notebook 4 NERDL Training](#)
- ❖ [Notebook 5 ClassifierDL, SentimentDL, MultiClassifierDL Training](#)

Spark NLP
for Data Scientists



Christian Kasim Loan
Sr. Data Scientist
christian@johnsnowlabs.com

End of Day 1

See you tomorrow same time same place :)

Spark NLP
for Data Scientists



Christian Kasim Loan
Sr. Data Scientist
christian@johnsnowlabs.com

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



Christian Kasim Loan
Sr. 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,

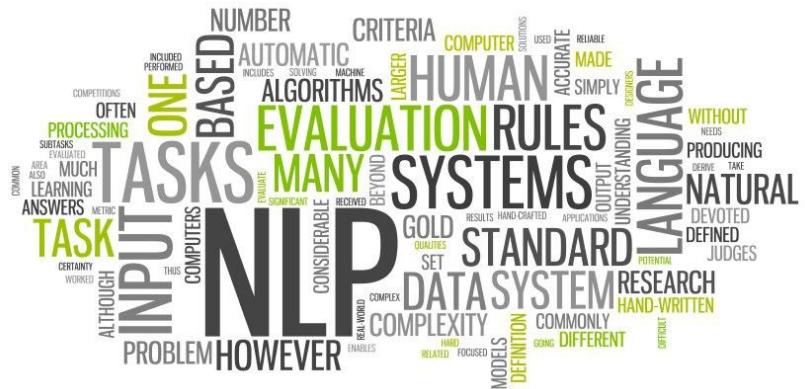
<https://www.merriam-webster.com/dictionary/siter>

[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 implementation a little bit to cover broken sentences and some impossible end of line chars.

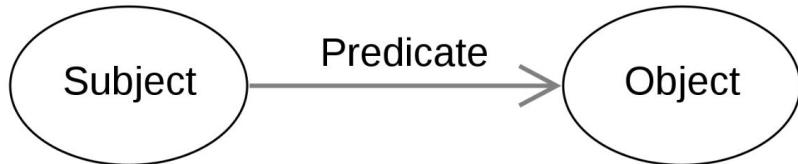
We are releasing two pretrained SDDL models: english and multilanguage that are trained on SETimes corpus (Tyers and Alperen, 2010) and Europarl. Wong et al. (2014) datasets.

Here are the test metrics on various languages for multilang model

Notebook 9 Sentence Detector DL

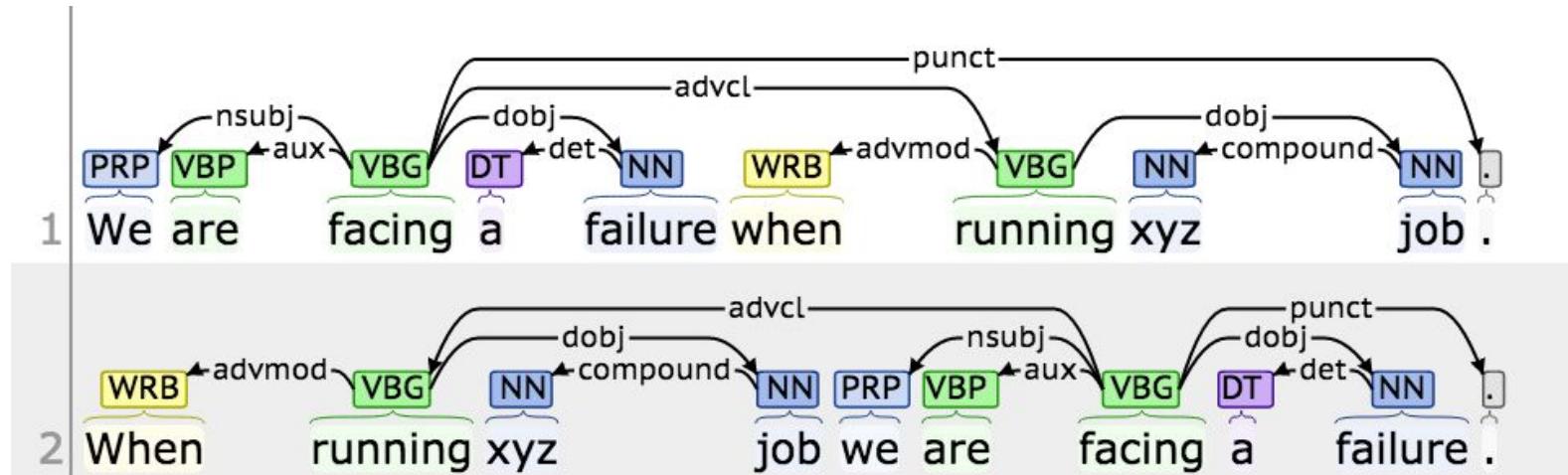
Language	Accuracy	Recall	Precision	F1
bg	0.96	1.00	0.93	0.96
bs	0.98	1.00	0.96	0.98
de	0.97	0.99	0.94	0.97
el	0.97	1.00	0.94	0.97
en	0.98	1.00	0.96	0.98
hr	0.98	1.00	0.96	0.98
mk	0.96	0.99	0.93	0.96
ro	0.97	1.00	0.95	0.97
sq	0.98	1.00	0.96	0.98
sr	0.98	1.00	0.95	0.97
tr	0.98	0.99	0.96	0.98

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
Sr. 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**



Christian Kasim Loan
Sr. 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 sofar.
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 Ngram may be repeated. Setting it to N, all Ngrams of size N can only occur once.

[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")
```

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

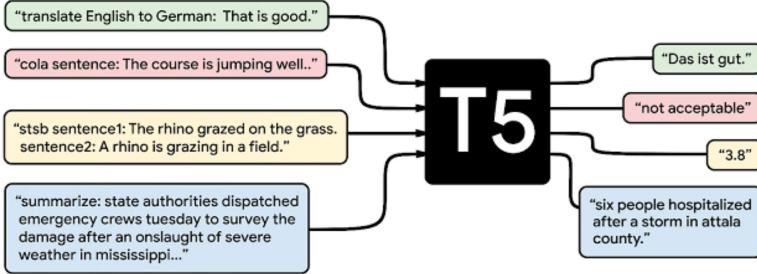
```
gpt2_pipe.predict("""Generate a top 10 movie list: \n  
1.The Matrix \n  
2.Terminator \n  
3. """)
```



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.



```

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

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

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

```

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

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



Every T5 Task with explanation:

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

Transformer Based Token and Sequence Classification

```
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")
```

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.

1 entry ?

38] # 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. """)
```



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.

1 entry ?

[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
Sr. 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**



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

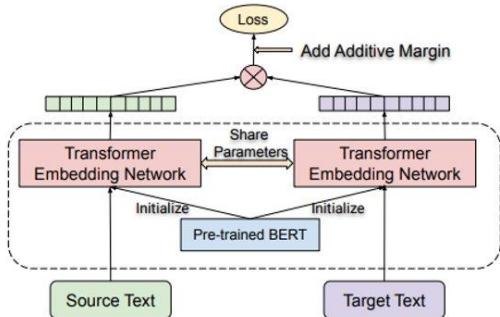
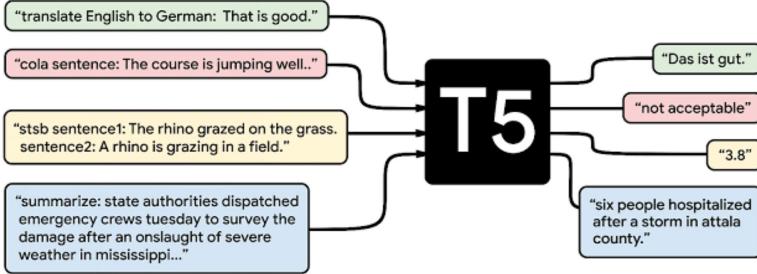


Figure 1: Dual encoder model with BERT based encoding modules.



```

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

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

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

```

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

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



Every T5 Task with explanation:

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

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



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

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



Part - III (Day 2) - Coding Time

- ❖ Notebook 5.2 training Multilingual Classifier
- ❖ Notebook 10 Question Answering and Summarization with T5



Spark NLP
for Data Scientists



Christian Kasim Loan
Sr. 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/why-spark-nlp-is-the-most-widely-used-nlp-library-enterprise/>

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

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

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

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

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

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
Sr. 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, 0.70...]	[-0.003953204490244389, -1.5821468830108643, ...]	[-0.7420049905776978, -0.8647691011428833, 0.1...]	[0.04002974182367325, -0.43536433577537537, -0...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
all	[-0.2182299941778183, 0.691990178909302, 0.70...]	[1.1014549732208252, -0.43204769492149353, -0...]	[0.31148090958595276, -1.098618268966748, 0.3...]	[-0.8933112025260925, 0.44822725653648376, -0...]	[0.17885173857212067, 0.045830272138118744, -0...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
of	[-0.15289999544620514, -0.24278999865055084, 0...]	[1.1535910367965698, 0.28440719842910767, 0.60...]	[-1.403516411781311, 0.3108177185058594, -0.32...]	[-0.5550722479820251, 0.2702311873435974, 0.04...]	[0.24783466756343842, -0.248960942029953, 0.02...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
them	[-0.10130999982357025, 0.10941000282764435, 0...]	[0.5475010871887207, 0.8660883903503418, 2.817...]	[-0.7559828758239746, -0.4712887704372406, -1...]	[-0.2922026813030243, -0.1301671266555786, -0...]	[-0.24157099425792694, -0.8055092692375183, -0...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
at	[0.17659999430179596, 0.0938510000705719, 0.24...]	[-0.5005946159362793, -0.4600788354873657, 0.5...]	[0.04092511534690857, -1.0951932668685913, -1...]	[-0.5613634586334229, -0.00903533399105072, ...]	[-0.11999595910310745, 0.012994140386581421, ...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
once	[-0.2383799999523163, 0.22167001745224, 0.35...]	[-0.39100387692451477, -0.8297092914581299, 2...]	[-0.46001458168029785, -1.2062749862670898, 0...]	[0.29886400609961548, 0.3360409140586853, -0.37...]	[0.6701997518539429, 1.1368376016616821, 0.244...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
!	[0.38471999764442444, 0.49351000785827637, 0.4...]	[0.007945209741592407, -0.27733859419822693, 0...]	[-1.5816600322723389, -0.992130696773529, -0.1...]	[0.7550013065338135, -0.525778167724609, -0.4...]	[-1.335283073425293, 0.6296550035476685, -1.4...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
Watch	[-0.38264000415802, -0.08968199785924078, 0.02...]	[-0.10218311846256256, -0.433427620526886, 0...]	[-1.3921688795089722, 0.6997514963150024, -0.8...]	[-0.24852752685546875, 1.222611427307129, -0.1...]	[0.04002974182367325, -0.43536433577537537, -0...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
your	[-0.5718399882316589, 0.046348001807928085, 0...]	[-0.4086211323738098, 1.0755341053009033, 1.78...]	[-0.8588163256645203, -2.3702170848846436, 0.0...]	[-0.035358428955078125, 0.7711482048034668, 0...]	[0.17885173857212067, 0.045830272138118744, -0...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
RAM	[-1.87559980534309, -0.40814998745918724, 0...]	[-0.09772858023643494, 0.3632940351963043, -0...]	[1.1277621984481812, -1.689896583557129, -0.19...]	[0.4528151750564575, -0.36768051981925964, -0...]	[0.24783466756343842, -0.248960942029953, 0.02...]	[[-0.0019260947592556477, 0.009215019643306732...]	1
tough	[-0.5099300146102905, -0.142800032901764, 0.5...]	[-0.22261293232440948, 0.21325691044330597, 0...]	[-1.3547197580337524, 0.43423181772232056, -1...]	[0.46073707938194275, 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 array', ' Of strings!'])
```

Pandas data frame

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

Pandas series

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

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

Spark
Data Frame

Ray
Data Frame

Dask
Data Frame

NLU : Apache License 2.0

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

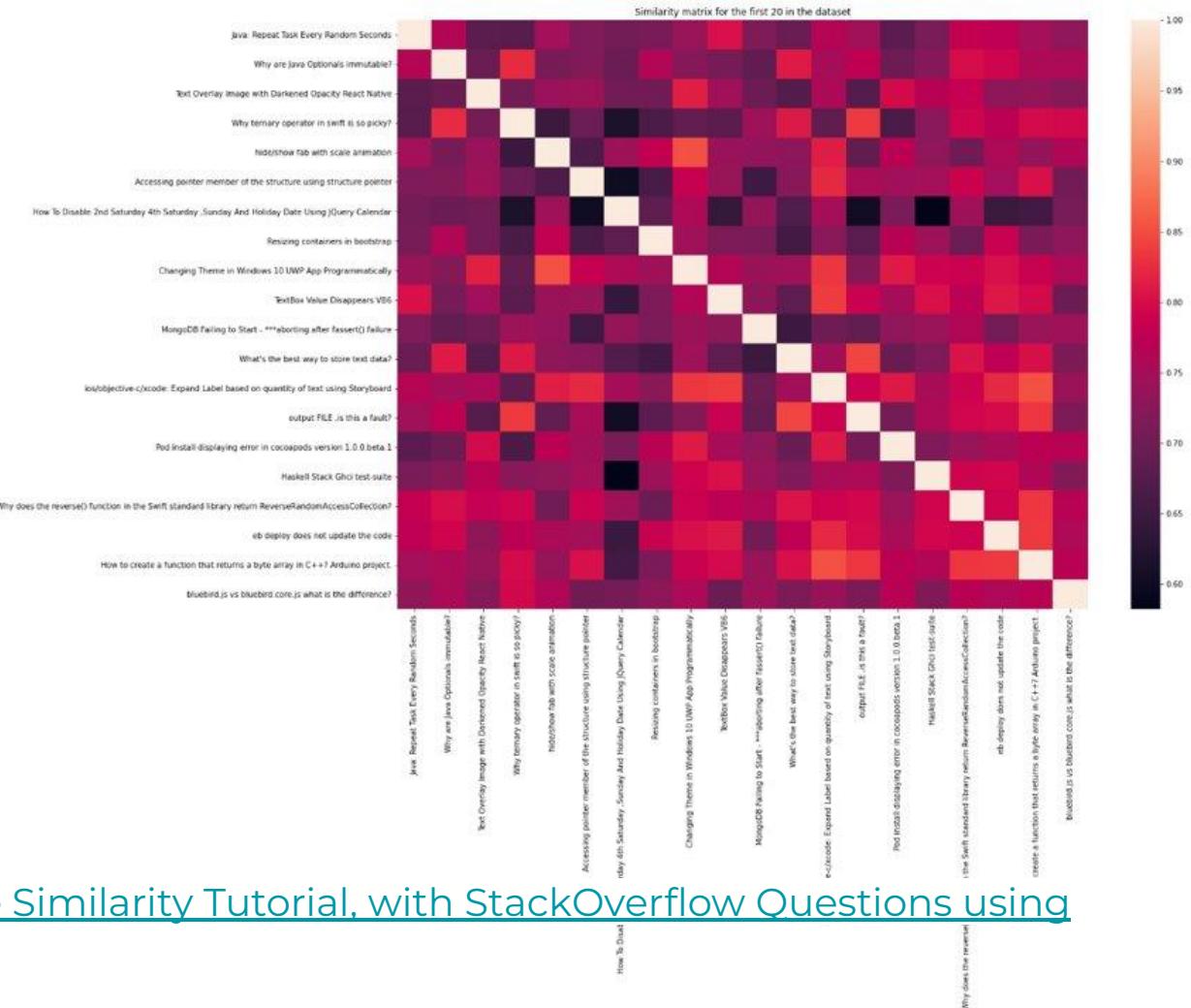
# Get BERTology and Transformer Embeddings for Sentences and Words
nlu.load('bert').predict('BERTology Word embeddings!')
nlu.load('bert elmo albert glove').predict('Multiple BERTology Word embeddings!')
nlu.load('embed_sentence.bert').predict('BERTology Sentence embeddings!')

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

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

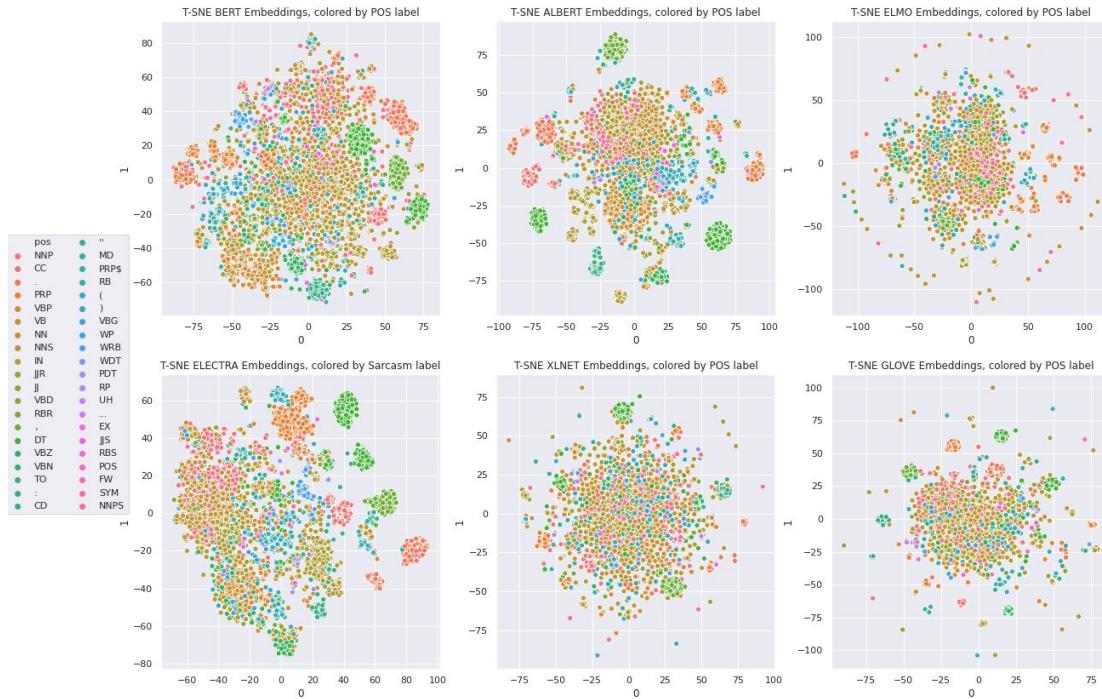
Sentence Similarity With BERTology Embeds or T5

Document



[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
```

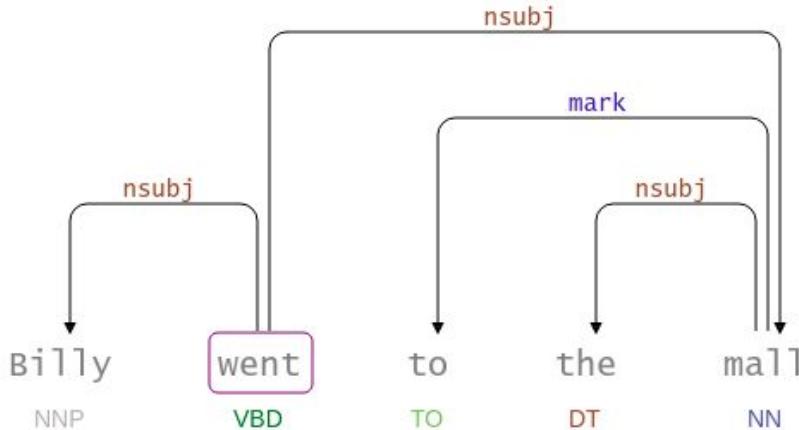
```
[OK!]
```

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

Visualize Dependency Trees

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

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



Visualize Assertion results

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

ner_clinical download started this may take some time.

Approximate size to download 13.9 MB

[OK!]

assertion_dl download started this may take some time.

Approximate size to download 1.3 MB

[OK!]

embeddings_clinical download started this may take some time.

Approximate size to download 1.6 GB

[OK!]

sentence_detector_dl download started this may take some time.

Approximate size to download 354.6 KB

[OK!]

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

TEST

PRESENT

PROBLEM

ABSENT

Visualize Resolution

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

jsl_ner_wip_clinical download started this may take some time.

Approximate size to download 14.5 MB

[OK!]

chunkresolve_rxnorm_in_clinical download started this may take some time.

Approximate size to download 26.9 MB

[OK!]

embeddings_clinical download started this may take some time.

Approximate size to download 1.6 GB

[OK!]

sentence_detector_dl download started this may take some time.

Approximate size to download 354.6 KB

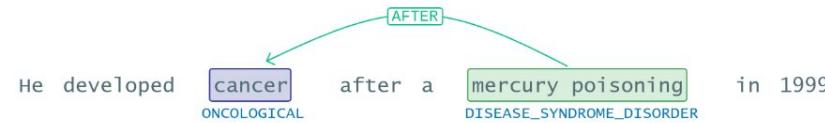
[OK!]

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

Visualize Entity Relationships

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

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





nlu_viz_cheatsheet.py

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

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

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

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

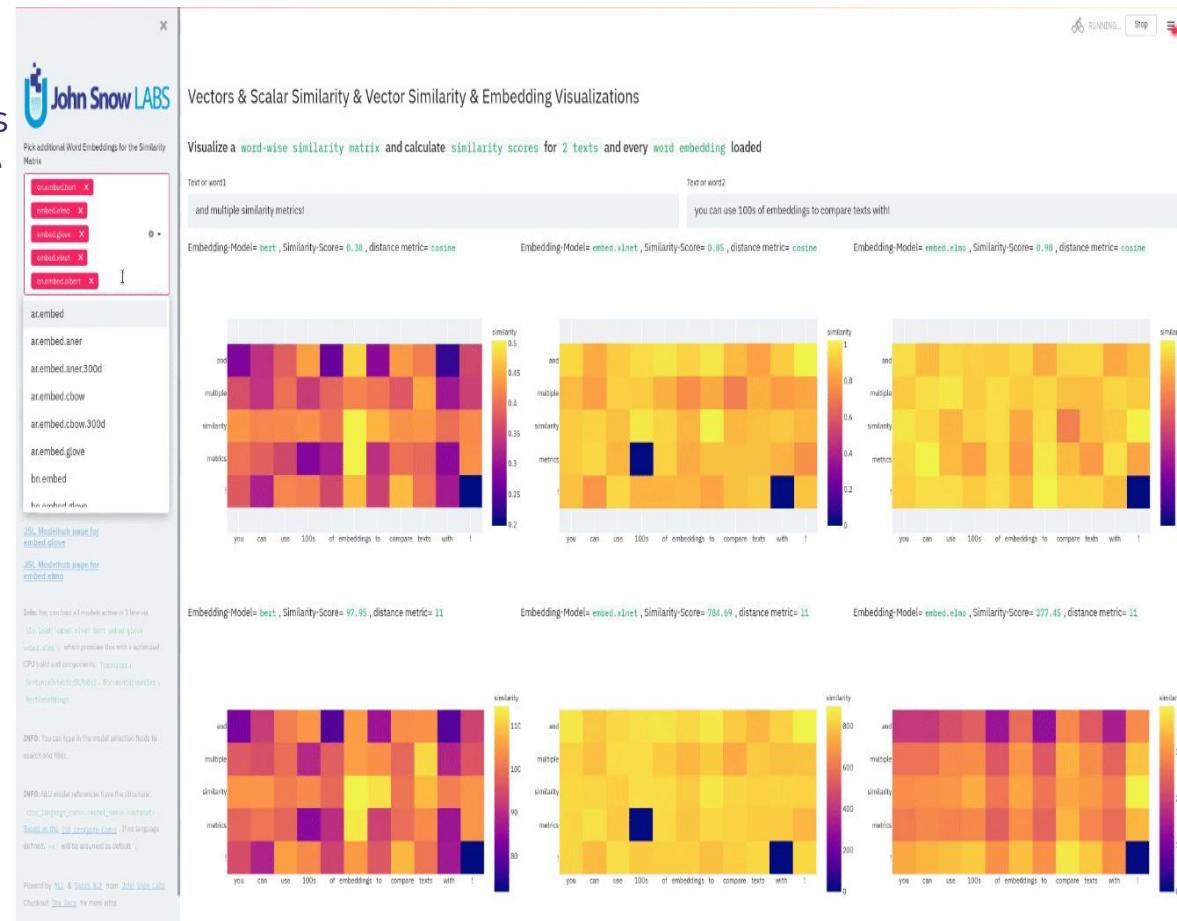
# Viz resolved sentences
nlu.load('med_ner.jsl.wip.clinical resolve.icd10cm').viz(data)

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

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

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

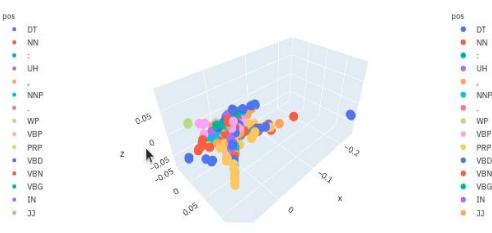
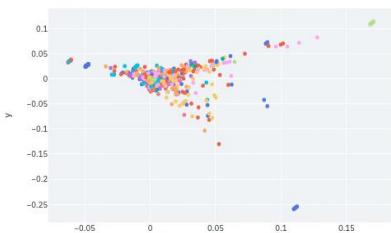


Explore 100+ Embeddings via 10+ Manifold and Matrix Decomposition with 0 lines of code with NLU & Streamlit

Compare multiple dimension reduction Techniques and Embeddings at the same time

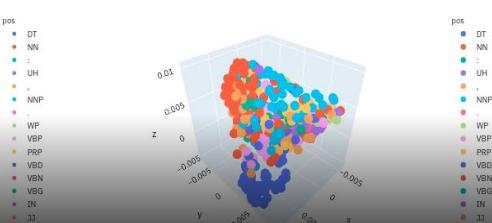
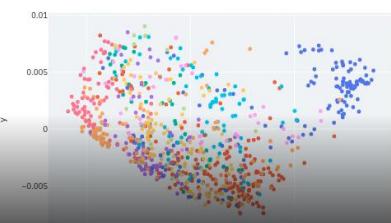
Word-Embeddings = small_bert_L2_128 , Manifold-Algo = LLE for D=2

Word-Embeddings = small_bert_L2_128 , Manifold-Algo = LLE for D=3



Word-Embeddings = small_bert_L2_128 , Manifold-Algo = Spectral Embedding for D=2

Word-Embeddings = small_bert_L2_128 , Manifold-Algo = Spectral Embedding for D=3



The screenshot shows a Streamlit application interface for visualizing sentence embeddings. At the top, it says "Lower dimensional Manifold visualization for sentence embeddings". Below that, it asks "Apply any of the 11 Manifold OR Matrix Decomposition algorithms to reduce the dimensionality of Word Embeddings to 1-D, 2-D and 3-D". It lists several options: T-SNE, PCA, SVD, Autoencoder, tSNE, tPCA, tSVD, SentenceBERT, and SentenceT-SNE. The main area contains three 2D scatter plots and one 3D scatter plot, each showing sentence embeddings colored by sentiment (positive, negative, neutral). The 3D plot is labeled "Harry Potter and the Philosopher's Stone". On the left, there's a sidebar titled "NLU pipeline components info" with links to "Search over 100s available SOTA models in John Snow Labs ModelHub", "2D Manifold page for sentiment.mhd", and "3D Manifold page for sentence.bert". There's also a "Select additional manifold and dimension reduction techniques to apply" section with checkboxes for T-SNE, PCA, SVD, Autoencoder, tSNE, tPCA, tSVD, SentenceBERT, and SentenceT-SNE. At the bottom, there's a "Powered by" link to "John Snow Labs" and a "Check out our live site".



nlu_streamlit_cheatsheet.py

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

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

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

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

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

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

# Calculate similarity between two texts based on Word Embeddings
nlu.load('bert').viz_streamlit_word_similarity(['I love NLU! <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	ned_ner.adde.biobert	Aspirin, vomiting	DRUG, ADE	CAECD Twineid
Anatomy	Find body parts, anatomical sites & related entities	ned_ner.anatomy	tubules, nasopharyngeal aspirates, embryoid bodies, NK cells, epithelial-mesenchymal transition, fistulas, heart, colic, 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	AniEM
Cellular/Molecular Biology	Find Genes, Molecules, Cell or general Biology related entities	ned_ner.cellular.biobert	human T-cell leukemia virus type 1 Tax-responsive, primary T-lymphocytes, TLR3, Spi-B mRNA, zeta-globin	DNA, Cell type, Cell_line, DNA, Protein	JNLPBA
Chemical/Genes/Proteins	Find Chemical, Gene and Protein related entities	ned_ner.chemprot.clinical	nitrogen, β -amyloid, NF-kappaB	CHEMICAL, GENE_Y, GENE_N	ChemProt
Chemical Compounds	Find general chemical compound related entities	ned_ner.chemicals	resveratrol, β -phenylethanol	CHEM	Dataset by John Snow Labs
Drug/Chemicals	Find chemical and drug related entities	ned_ner.drugs	potassium, anthracyclines, taxanes	DrugChem, DrugChem, DrugChem	IDb2 + FDA
Physiology/Drugs	Find physiology and drug related entities	ned_ner.physology.biobert	5000 units, Aspirin, 14 days, tablets, daily, topically, 30 mg	DOSE, DRUG, DURATION, FORM, FREQUENCY, ROUTE, STRENGTH	IDb2 + FDA
Risk Factors	Find risk factor of patient related entities	ned_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, OBESITY, PANTHER, MORBID_OBESITY, HYPERTENSION_PHI, HYPERLIPIDEMIA, DIABETES	De-identification and Heart Disease Risk Factors Challenge datasets
Cancer Genetics	Find cancer and genetics related entities	ned_ner.cancer	Human, Kir 3.3, GIRQ3, potassium, Human chromosome 23, pancreas, tissues, fat antineoplastic agent, KIR3, type II, breast cancer, patients, anthracyclines, taxanes, vinorelbine, patients, drug, vinorelbine, imatinib, anthracyclines	Amino acid, Anatomical_system, Cancer, Cell, Cellular component, Developing anatomical Structure, Gene or gene product, Immaterial_anatomical_entity, Multi-tissue structure, Organ, Organelle, Organism subdivision, Simple_chemical, Tissue	CG TASK of BioNLP 2012
Diseases	Find disease related entities	ned_ner.diseases.biobert	the cyst, a large Prolene suture, a very small incisional hernia, the hernia cavity, inguinal hernia, the wound lesion, The lesion, the existing scar, the cyst, the hernia, this cyst down to its base, it is located in the hernia, the cyst	Disease	CG TASK of BioNLP 2012
Bacterial Species	Find bacterial species related entities	ned_ner.bacterial_species	Neisseria wadsworthii, N. bacilliformis, N. spirochetal litralis	SPECIES	Dataset by John Snow Labs
Medical Problem/Test/Treatment	Find medical problem,test and treatment related entities	ned_ner.healthcare	respiratory tract infection, outpatient studies, stavastatin	PROBLEM, TEST, TREATMENT	IDb2
Clinical Admissions/Events	Find clinical admission event related entities	ned_ner.admission_events	2007, 12 AM, Headache, blood sample, presented, emergency room, daily	DATE, TIME, PROBLEM, TEST, TREATMENT, CLINICAL_DEPT, EPISITENTIAL_DURATION, FREQUENCY, ADMISSION, DISCHARGE	Custom IDb2, enriched with Events
Genetic Variants	Find genetic variant related entities	en_ner.genetic_variants	rs1061178, p.S45P, T1304C	DNAmutation, ProteinMutation, SNP	TMVAR
PHI (Protected Healthcare Information)	Find PHI(Protected Healthcare Information)	en.ned_ner.deid	2003-01-13, David Hale, Hendrickson, Ora, T194334, 01/13/03, Oliveira, 25-13001-11-2888, Cestka County Baptist Hospital, 9200 Health Street, (302) 780-5227, Brothers Coal Mine		n2c2 (Db2-PHI)
Social Determinants / Demographic Data	Find Social Determinants and Demographic Data Related Entities	ned_ner.jsl.enriched	21-day-old, male, congestion non, suctioning yellow discharge, the child has been having diarrhea, perioral cyanosis, retractions, non, Tylenol, His, his, respiratory congestion, He tired, fussy, abdominal	Age, Diagnosis, Dosage, Drug Name, Frequency, Gender, Lab_Name, Lab_Result, Symptom_Name	Dataset by John Snow Labs
General Clinical	Find General Clinical Entities	ned_ner.jsl.wip.clinical.modifier	28-year-old, female, gestational diabetes, mellitus, eight years, prior_type, type, I, age, 28, gestational, mellitus, T2DM, HbG-induced, pancreatitis, three, years, prior, type, I, gestational, diabetes, hepatitis, obesity, body, mass, index, BMI, kg/m ² , polycythaemia, polydipsia, polyuria, post, asymptote, vomiting, the, last, weeks, prior, she, five-day, course	Injury or Poisoning, Direction, Test, Admission_Discharge, Death Entity, Relationship_Status, Duration, Response, Hospitalization, Procedure, Labour_Delivery, Family_History, Header, BMI, Temperature, Alcohol, Nicotine, Kidney Disease, Oncological, Medical History Header, Cerebrovascular_Disease, Oxygen_Therapy, Oral_Supplements, Respiratory_Disease, Condition, Heart_Disease, Employment, Obesity, Disease_Syndrome, disorder, Pregnancy, Immunotherapy, Prescription, Medical_Device, Race_Ethnicity, Sex_Gender, Age, Allergy, Treatment, Substance, Recipe, Drug_Ingredient, Blood, External_body_part or region, Lip, Ear, Eye, Mouth, Nose, EKG_Findings, Imaging_Technique, Glycylcerides, RelativeLine, Gender, Pulse, Socioeconomic_Status, Insurance_Quantity, Diabetes, Modifier, Immune_System, Component, Clinical_Dept, Form, Drug_BrandName, Strength, Fetus_Newborn, Result, Vital_Signs, Response, Sex, Sexual_Orientation, Frequency, Time, Vital_Signs_Header, Communicable_Disease, Dosage, Overweight_Hypertension, Hba1c, Total_Cholesterol, Smoking,	Dataset by John Snow Labs

NER Domains and Models Overview

Radiology	Find Radiology related entities	ned_ner.radiology.wip_clinical	Bilateral, breast, ultrasound, ovoid mass, 0.5 x 0.5 x 0.4, cm; anteromedial aspect, left, shoulder, mass, isoechogenic echotexture, muscle, internal color flow, benign fibrous tissue, lipoma	ImagingTest, Imaging Technique, ImagingFindings, OtherFindings, BodyPart, Direction, Test, Symptom, Disease_Syndrome_Disorder, Medical_Device, Procedure, Measurements, Units	Dataset by John Snow Labs, MIMIC-CXR and MT Radiology texts
Radiology Clinical JSL-V1	Find radiology related entities in clinical setting	ned_ner.radiology.wip_greedy_biobert	Bilateral, breast, ultrasound, ovoid mass, 0.5 x 0.5 x 0.4, cm, anteromedial aspect, left, shoulder, mass, isoechogenic echotexture, muscle, internal color flow, benign fibrous tissue, lipoma	Test Result, OtherFindings, BodyPart, ImagingFindings, Disease_Syndrome_Disorder, ImagingTest, Measurements, Procedure, Score, Test, Medical_Device, Direction, Symptom, Imaging_Technique, ManualFix, Units	Dataset by John Snow Labs
Genes and Phenotypes	Find Genes and Phenotypes (the observable physical properties of an organism) related entities	ned_ner.human_phenotype.gene_biobert	AP0C4 . 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	ned_ner.human_phenotype.go_biobert	protein complex oligomerization , defective platelet aggregation	G0, HP	PGR_1, PGR_2
				Kidney Disease, HDL, Diet, Test, Imaging Technique, Triglycerides, Obesity, Duration, Weight, Sex, Height, Header, ImagingTest, Labour_Delivery, Disease_Syndrome_Disorder, Communicable_Disease, Overweight, Units, Smoking, Score, Substance_Quantity, Form, Frequency, Modifier, Hyperlipidemia, ImagingFindings, Psychological_Condition, OtherFindings, Cerebrovascular_Disease, Date, Test_Result, VS_Finding, Employment, Death_Entity, Gender, Oncological, Headache, Disease, Medical_Device, Total_Cholesterol, ManualFix, Time, Route, Pulse, Admission_Discharge, RelativeDate, O_Saturation, Frequency, Red_Actions, Hypertension, Alcohol, Allergen, Fever, Newborn, Birth_Entity, Age, Respiration, Medical_History_Header, Oxygen_Therapy, Section_Header, LDL, Treatment, Vital_Signs_Header, Direction, BMI, Pressure, Temperature, 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	
Radiology Clinical JSL-V2	Find radiology related entities in clinical setting	ned_ner.jsl.wip.clinical.rd		Qualitative_Concept, Organization, Manufactured_Object, Amino_Acid, Peptide or Protein, Pharmacologic_Substance, Professional or Occupational_Group, Cell_Component, Neoplastic_Process, Substance, Laboratory_Procedure, Nucleic_Acid_Nucleoside or Nucleotide, Research_Group, Genetic_Element, Indicator_Reagent_or_Diagnostic_Aid, Biologic_Function, Chemical_Manufacture, Molecular_Function, Quantitative_Concept, Prokaryote, Mental or Behavioral_Dysfunction, Injury or Poisoning, Body_Location or Region, Spatial_Relation, Molecular_Sequence, Tissue, Pathologic_Function, Body_Substance, Fungus, Mental_Process, Medical_Device, Plant, Health_Care_Activity, Clinical_Attribute, Genetic_Function, Food, Therapeutic or Preventive_Procedure, Bio, Biological, Organism, Organ, Organ_Component, Geographic_Area, Virus, Biomedical or Dental_Material, Diagnostic_Procedure, Eukaryote, Anatomical_Structure, Organism_Attribute, Molecular_Biology_Research_Technique, Organic_Chemical_Cell, Daily or Recreational_Activity, Person, Group, Disease or Syndrome, Group_Sign or Symptom, Body_System	Dataset by John Snow Labs
General Medical Terms	Find general medical terms and medical entities.	ned_ner.medmentions			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	<ul style="list-style-type: none"> - 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, Hypothetical	<ul style="list-style-type: none"> - 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 daughter. - Past : The patient has no more headaches since the operation - Family : The patients father has cancer - Hypothetical : 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	<ul style="list-style-type: none"> - 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 daughter - 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	<ul style="list-style-type: none"> - 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 maintained by Centers for Medicare & Medicaid Services	resolve.icd10pcs	hypertension , gastritis	DWY18ZZ, 0472326	ICD10-PCS CMS ICD-10-PCS
ICD-O (International Classification of Diseases, Oncology) Topography & Morphology codes	Get ICD-8 codes of Medical and Clinical Entities. The International Classification of Diseases for Oncology (ICD-O), is a domain-specific extension of the International Statistical Classification of Diseases and Related Health Problems for tumor diseases.	resolve.icdo.base	metastatic lung cancer	9858/3 + C38.3, 8801/3 + C39.8	ICD-O Histology Behaviour dataset
HCC (Hierarchical 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	C7880 + ['1', '1', '8']	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/diagnoses. The codes are used to derive the amount of payment a healthcare provider may receives from insurance companies for the provided service/receives	resolve.cpt.procedures_measurements	calcium score, heart surgery	82310, 33257	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, 0180753	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	148439082	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	50214004	SNOMED
RxNorm and RxCUI (Concept Unique Identifier)	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.human_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 occurred after Another Entity - BEFORE if Any Entity occurred 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 - TrMP: A patient's medical problem has deteriorated or worsened because of treatment - TrCP: A treatment caused a medical problem - TrAP: A treatment administered for a medical problem - TrAP: The administration of a treatment was avoided because of a medical problem - TeRP: A test has revealed some medical problem - TeCP: A test was performed to investigate a medical problem - PIP : Two problems are related to each other
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 upper 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 Lipitor - 0 for sever fatigue and voltaren - 0 for cramps and Lipitor - 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 aciAntology
Temporal events	She is diagnosed with cancer in 1991. Then she was admitted to Mayo Clinic in May 2000 and discharged in October 2001	- OVERLAP for cancer and 1991 - AFTER for admitted and Mayo Clinic - BEFORE for admitted and discharged	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 : Azotenia presumed secondary to sepsis	- TrIP for infection and antibiotic course - TrWP for tumor and drain - TrCP for penicillin and rash - TrAP for Dexamphetamine and narcolepsy - TrNAP for Ralafen and ulcers - TeRP for echocardiogram and pericardial effusion - TeCP for chest x-ray and pneumonia - PIP for Azotenia and sepsis	2010 I2b2 relation challenge
DDI Effects of using Multiple Drugs (Drug Drug Interaction)	- DDI-advice: UROXATRAL should not be used in combination with other alpha-blockers - DDI-effect: Chlorthalidone may potentiate the action of other antihypertensive drugs - DDI-int: The interaction of omeprazole and ketoconazole has been established - DDI-mechanism: Grepafloxacin may inhibit the metabolism of the theobromine - DDI-false: Aspirin does not interact with Chlorthalidone	- DDI-advice for UROXATRAL and alpha-blockers - DDI-effect for Chlorthalidone and antihypertensive drugs - DDI-int for omeprazole and ketoconazole - DDI-mechanism for Grepafloxacin and theobromine - DDI-false for Aspirin and Chlorthalidone	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	Magie, 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 agonists at distinct opioid receptors - CPR:6 (Antagonist) : GDC-0152 is a drug triggers tumor cell apoptosis by selectively antagonizing LAPs - CPR:7 (Modulator) : Hydrogen sulfide is a allosteric modulator of ATP-sensitive potassium channels - CPR:8 (Cofactor) : polyinosinic:polycytidylic acid and the IFNa/β demonstrate capability of endogenous IFN. - CPR:9 (Substrate) : ZIP9 plays an important role in the transport and toxicity of Cd(2+) cells - CPR:10 (Not Related) : Studies indicate that GSK-3β inhibition by palbutorin cannot be competed out by ATP	- CPR:1 (Part of) for amino acid and rabbit alpha(2A)-adrenoceptor - CPR:2 (Regulator) for Triacsin and ACS - CPR:3 (Upregulator) for Ibandoante and FAS gene - CPR:4 (Downregulator) for Vitamin C and C-Rel - CPR:5 (Agonist) for tricyclic antidepressants and opioid receptors - CPR:6 (Antagonist) (Antagonist) for GDC-0152 and LAPs - CPR:7 (Modulator) for Hydrogen sulfide and ATP-sensitive potassium channels - CPR:8 (Cofactor) for polyinosinic:polycytidylic acid and IFNa/β - CPR:9 (Substrate) for ZIP9 and Cd(2+) cells - CPR:10 (Not Related) for GSK-3β and ATP	ChemProt Paper

NLU OCR

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

NLU Spell	Transformer Class
<code>nlu.load(img2text)</code>	ImageToText
<code>nlu.load(pdf2text)</code>	PdfToText
<code>nlu.load(doc2text)</code>	DocToText

Image to Text

“The Old Pond” by Matsuo Bashō

An old silent pond

Sample image:

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 Bashō
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+ Healthchare 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
Sr. Data Scientist
christian@johnsnowlabs.com

NLP Server - Powerful private deployable NLP Rest API

A screenshot of the AWS Marketplace search results for 'John Snow Labs'. The search bar at the top contains 'John Snow Labs'. The results page shows 15 items under the heading 'john Snow Labs (15 results) showing 1 - 15'. A red arrow points from the left margin towards the first item in the list, which is 'John Snow Labs - Annotation Lab'. Other visible items include 'John Snow Labs - NLP Server', 'Diagnosed Diabetes Prevalence 2004-2013', 'Respiratory Syncytial Virus Disease', 'Dobutamine Stress Echocardiography Data', and 'Studies On Safe Drugs During Pregnancy'.

John Snow Labs - NLP Server

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

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

[Show more](#)

Linux/Unix

[Continue to Subscribe](#)

[Save to List](#)

Typical Total Price

\$0.384/hr

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

Overview

Pricing

Usage

Support

Reviews

Product Overview

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

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

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

Highlights

- Spark NLP text annotation

Version

NLP Server 0.3.0
[Show other versions](#)

By

John Snow Labs

Categories

Natural Language Processing
Text

Operating System

Linux/Unix, Ubuntu 20.04

Delivery Methods

Amazon Machine Image

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

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

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

API Usage in 3 Steps :

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

1. Choose a spell

en.class

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

en.classify.spam
Spam Classifier

en.classify.spam.use
Spam Classifier

en.classify.toxic
Toxic Comment Classification

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

en.classify.trec50
TREC(50) Question Classifier

en.classify.trec50.relp

1. Choose a spell

en.ner

Recognize Entities DL Pipeline for English

2. Enter the Text to analyze

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

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



Group results by: No grouping Document Sentence Entity Word

3. Preview the Results:

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

Step 1 : Post a request

- Post a data processing request
- Returns UUID status

POST /api/results/

Request samples

Payload

Content type
application/json

Copy Expand all Collapse all

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

Response samples

200 404 422

Content type
application/json

Copy Expand all Collapse all

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

Step 2 : Poll for result

- Poll Server for processing status

GET /api/results/{uuid}/status

▼

Response samples

200 404 422

Content type
application/json

Copy Expand all Collapse all

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

Step 3 : Get result as JSON

- Get result for UUID
- Contains NLP predictions

GET /api/results/{uuid}

Response samples

200 404 422

Content type
application/json

Copy Expand all Collapse all

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

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



NLP_server_cheatsheet.py

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

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

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

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

YouTube Tutorials | NLP Server | Deployments on AWS

Thank you.



christian@JohnSnowLabs.com



@ckl_it



In/Christian-Kasim-Loan



Medium.com/@Christian.Kasim.Loan

Bonus:

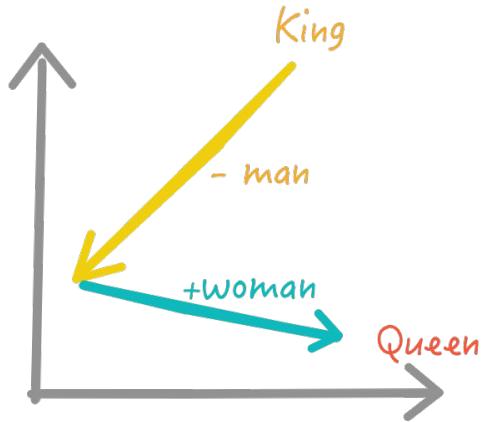
- ❖ Semantic Search and Similarity with Textual Embeddings
- ❖ Visualize Embeddings via Manifold and Decomposition Algorithms
- ❖ Notebook 11 Similarities and Dimension Reduction

Spark NLP
for Data Scientists

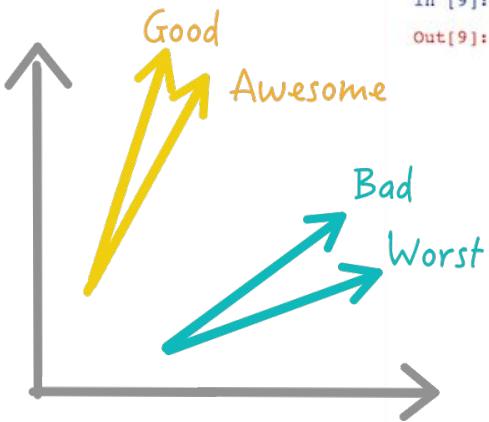


Christian Kasim Loan
Sr. Data Scientist
christian@johnsnowlabs.com

Word & Sentence Embeddings - reminder



a) Learns Analogy



b) Similar Words have same angles

In [9]: doc[3].vector

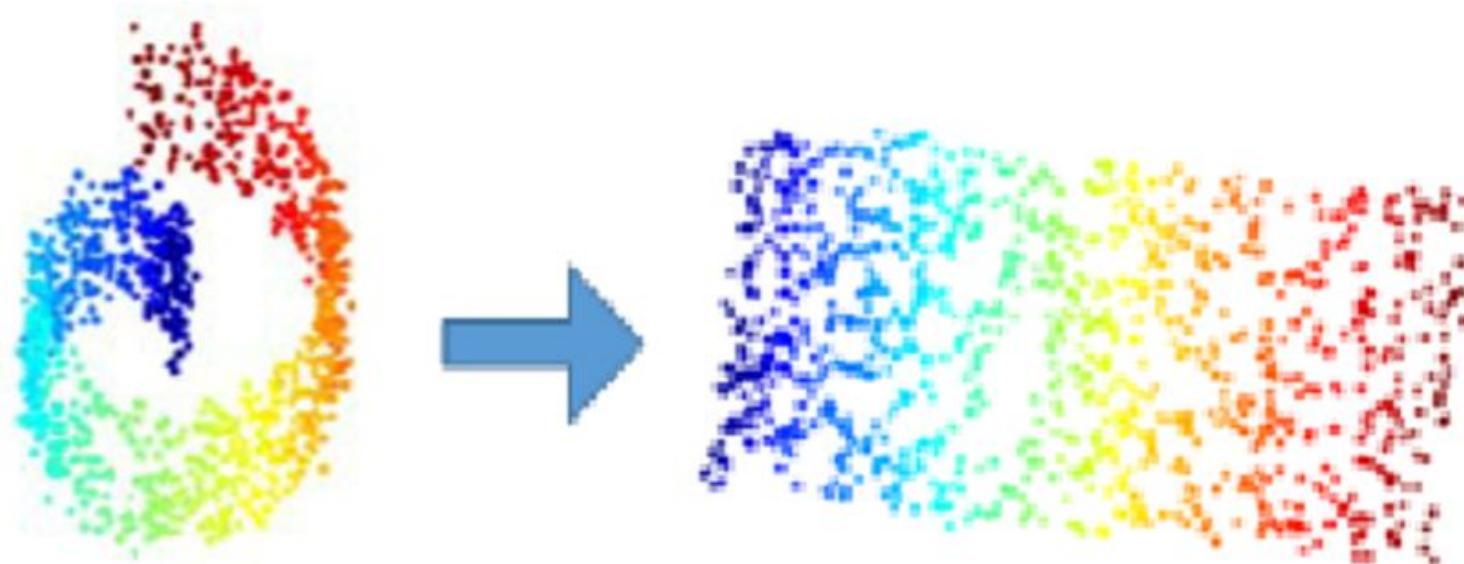
```
Out[9]: array([ 0.037103 , -0.31259 , -0.17857 ,  0.30001 ,  0.078154 ,
  0.17958 ,  0.12048 , -0.11879 , -0.20601 ,  1.2849 ,
 -0.20409 ,  0.80613 ,  0.34344 , -0.19191 , -0.084511 ,
  0.17339 ,  0.042483 ,  2.0282 , -0.16278 , -0.60306 ,
 -0.53766 ,  0.35711 ,  0.22882 ,  0.1171 ,  0.42983 ,
  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 ,
  0.00211152,  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 ,  0.00404073,  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

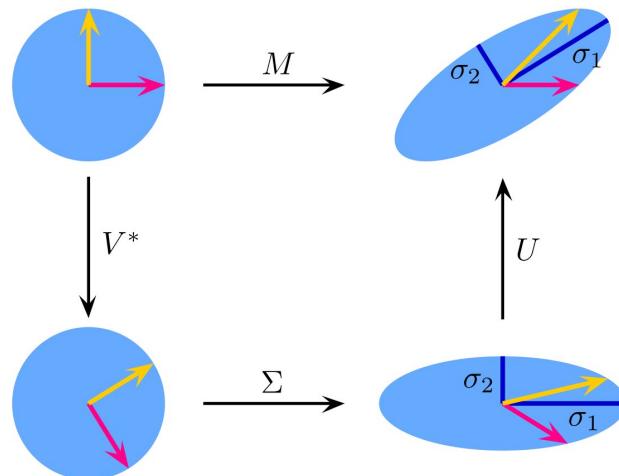
Reduce dimensionality by learning some lower dimensional structure

Which preserves relationship between original hyperspace

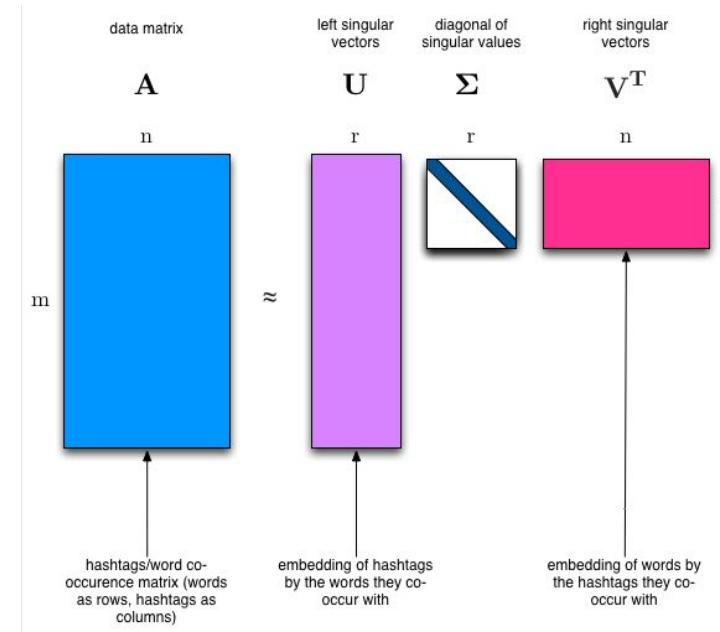


Matrix Decompositions

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



$$M = U \cdot \Sigma \cdot V^*$$





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

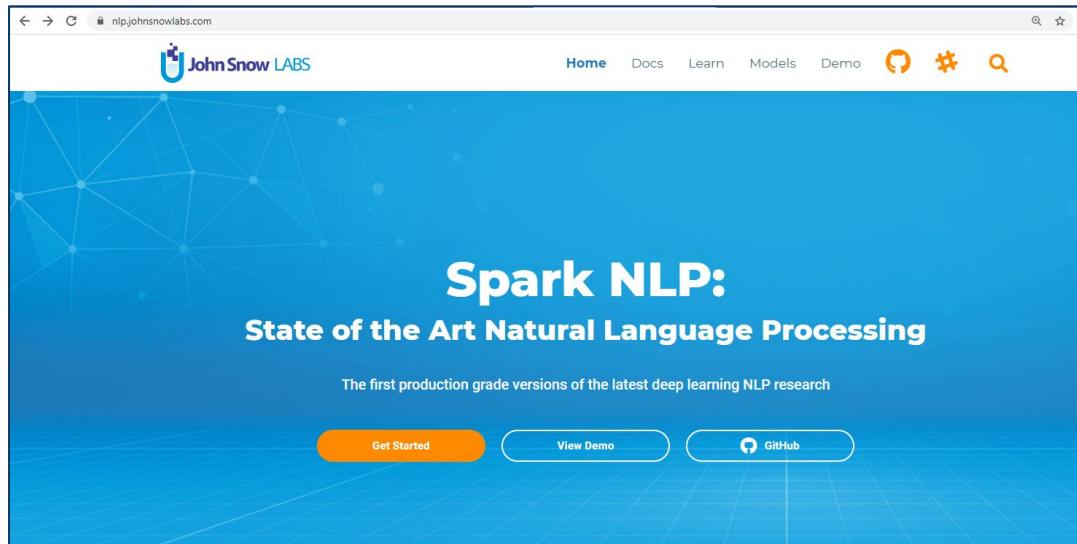
→ How to Use

→ Demo Pages

→ Colab Notebooks

→ Sample Code Snippets

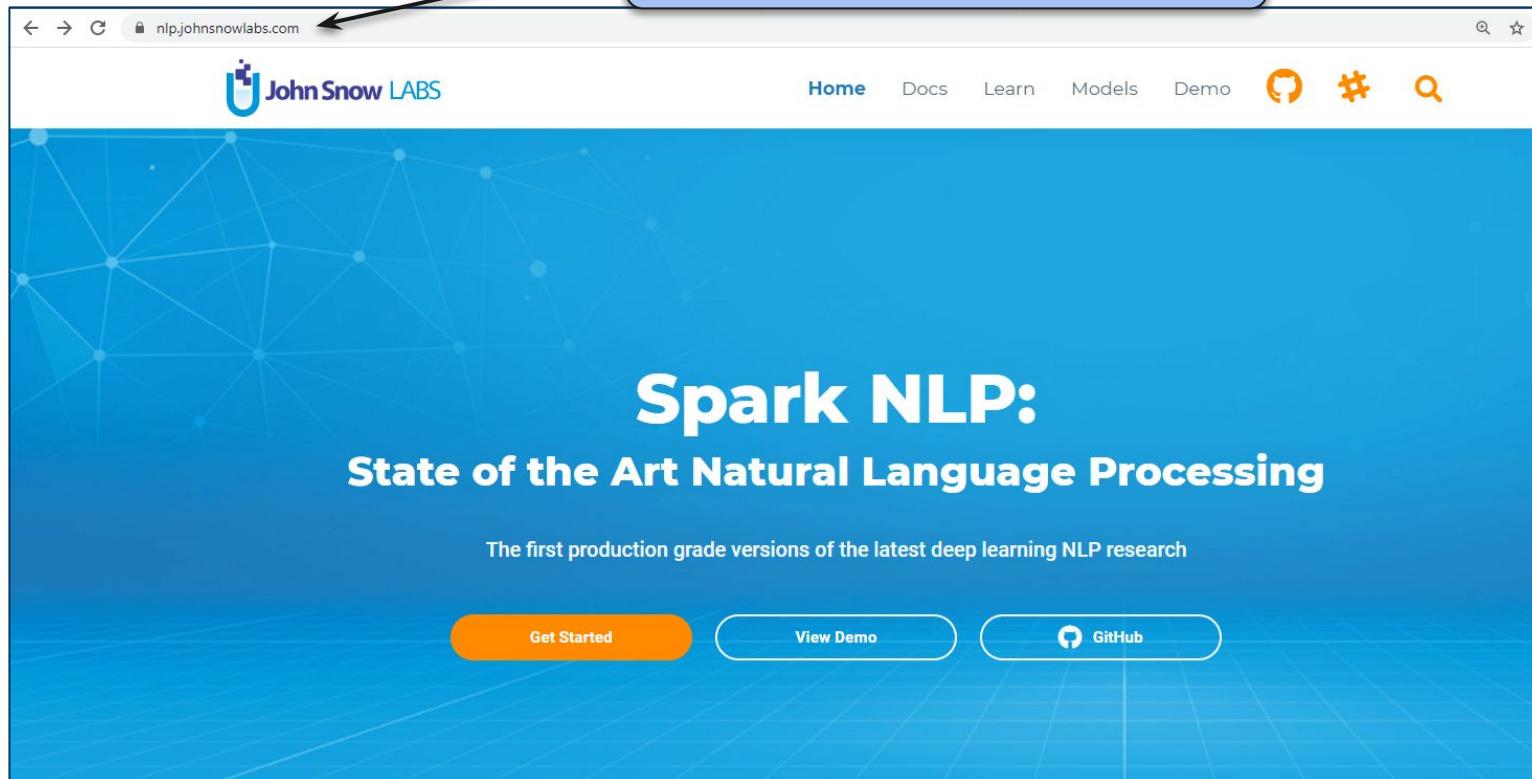
→ Benchmarking (Metrics)





Main Page

http://nlp.johnsnowlabs.com/



A screenshot of a web browser displaying the John Snow LABS main page. The URL "nlp.johnsnowlabs.com" is visible in the address bar. The page features a blue header with the "John Snow LABS" logo and navigation links for Home, Docs, Learn, Models, and Demo. Below the header is a large blue background image with a network graph. The main content area has a white background and displays the text "Spark NLP: State of the Art Natural Language Processing". Below this, a subtext reads "The first production grade versions of the latest deep learning NLP research". At the bottom are three orange call-to-action buttons: "Get Started", "View Demo", and a GitHub link.

nlp.johnsnowlabs.com

John Snow LABS

Home Docs Learn Models Demo

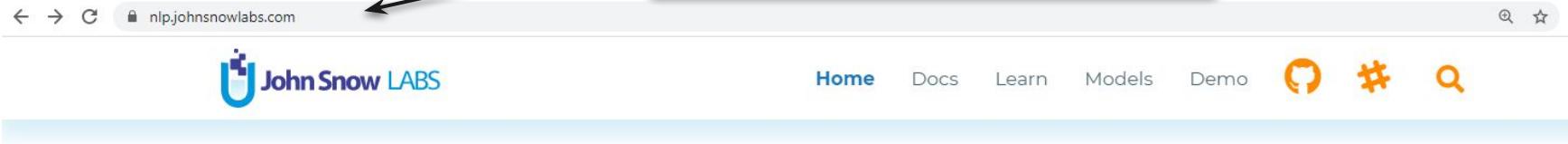
Spark NLP:
State of the Art Natural Language Processing

The first production grade versions of the latest deep learning NLP research

Get Started View Demo GitHub

Models Hub

http://nlp.johnsnowlabs.com/



Right Out of The Box

Spark NLP ships with many **NLP features**, pre-trained **models** and **pipelines**

Models Hub

Models Hub

Models Hub

Available Models and Pipelines

Show All 1036

aav⁴ af⁵ afa⁴ alv⁴ am² ar⁹ art² ase² assertion² az⁴ bat⁴ bcl⁴ bem⁴ ber⁴
bg⁷ bh² bho² bi⁴ bn⁵ bnt⁴ br³ bzs⁴ ca⁷ cau² ccs² ceb⁴ cel⁴ chk⁴ classifier¹⁹
clinical¹⁰¹ cn⁷ cpf⁴ cpp⁴ crs⁴ cs⁷ cus⁴ cy⁴ da⁶ de¹² deidentify⁴ dra⁴ ee⁴ efi⁴ el⁵
embeddings⁵⁵ en⁸⁰⁹ entity_resolution²⁹ eo⁵ es²² et⁶ eu⁷ euq⁴ fa⁵ fi⁷ fiu⁴ fij⁴ fr⁷ ga⁷
gaa⁴ gem⁴ gil⁴ gl⁷ gmq⁴ gmw⁴ grk⁴ guw⁴ gv⁴ ha⁵ he⁷ hi⁷ hil⁴ ho⁴ ht⁴ hu⁷
hy⁷ id⁷ ig⁴ iir⁴ ilo⁴ inc⁴ ine⁴ is⁴ iso⁴ it⁷ itc⁴ ja⁷ jap⁴ ka² kab² kg⁴ kj⁴ kl²
ko⁶ kqn⁴ kwn⁴ kwy⁴ la³ language_detection¹⁶ lemmatizer⁴¹ lg⁴ licenced¹ licensed⁸² ln⁴ loz⁴
lu⁴ lua⁴ lue⁴ lun⁴ luo⁴ lus⁴ lv⁵ map² mfe⁴ mg⁴ mh⁴ mk⁴ mkh⁴ ml⁴ mos⁴ mr⁷
mt⁴ mul⁴ nb² ner⁷⁹ ng⁴ nic⁴ niu⁴ nl⁹ nn¹ nso⁴ ny⁴ nyk⁴ om⁴ open_source⁸¹⁰ pa²
pag⁴ pap⁴ phi⁴ pipeline³⁵¹ pis⁴ pl⁸ pon⁴ pos⁴⁶ poz² pqe⁴ pqw² pt⁶ question_answering¹
re³ relation_extraction² relation_extraction⁴ rn⁴ rnd⁴ ro⁵ roa⁴ ru¹⁰ run⁴ rw⁴ sal⁴ sem⁴
sentence_detection⁵ sentiment¹⁰ seq2seq⁶⁴⁹ sg⁴ sit² sk⁷ sl³ sla⁴ sm⁴ sn⁴ so¹ sq⁴ srm²

Description Page



2020/05/10/wikiner_6B_300_pt.html

John Snow LABS

Home Docs Learn Models Demo

Description

WikiNER is a Named Entity Recognition (or NER) model, meaning it annotates text to find features like the names of people, places, and organizations. This NER model does not read words directly but instead reads word embeddings, which represent words as points such that more semantically similar words are closer together. WikiNER 6B 300 is trained with GloVe 6B 300 word embeddings, so be sure to use the same embeddings in the pipeline.

Predicted Entities

Persons- PER , Locations- LOC , Organizations- ORG , Miscellaneous- MISC .

Live Demo Open in Colab Download

May 18 Detect Persons, Locations, Organizations and Misc Entities in Polish (WikiNER 6B 300)

May 18 Detect Persons, Locations, Organizations and Misc Entities in Portuguese (WikiNER 6B 300)

May 18 Detect Persons, Locations, Organizations and Misc Entities in Dutch (WikiNER 840B 300)

May 18 Detect Persons, Locations, Organizations and Misc Entities in Polish (WikiNER 840B 300)

May 18 Detect Persons, Locations, Organizations and Misc Entities in Portuguese (WikiNER 840B 300)

... DETECTING OTHER LANGUAGES



How to Use

How to

Python

Scala

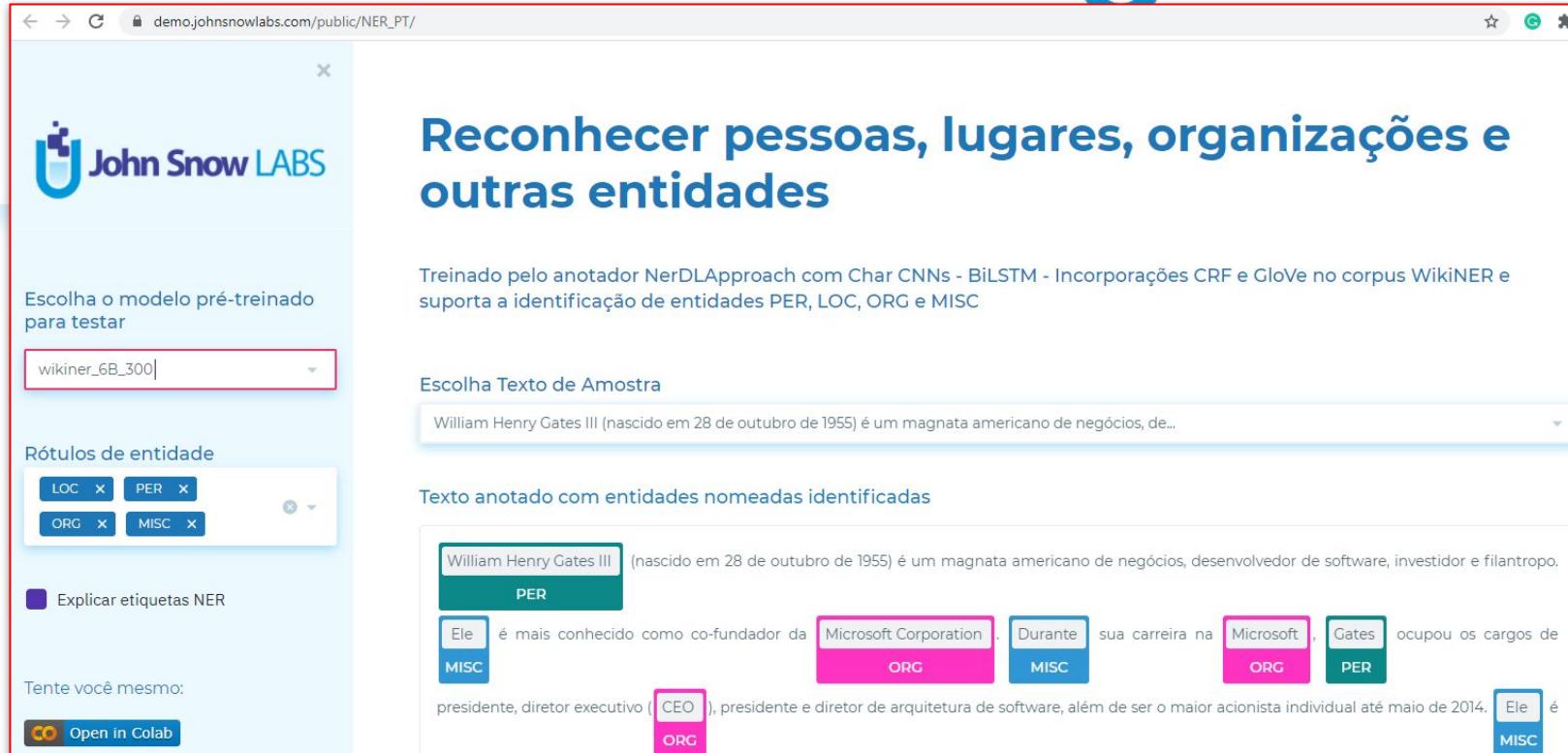
```
...  
ner_model = NerDLModel.pretrained("wikiner_6B_300", "pt") \  
    .setInputCols(["document", "token", "embeddings"]) \  
    .setOutputCol("ner")  
...  
nlp_pipeline = Pipeline(stages=[documentAssembler, sentence  
pipeline_model = nlp_pipeline.fit(spark.createDataFrame([[[  
  
result = pipeline_model.transform(spark.createDataFrame(pd.D
```

How to use

Python Scala

```
...  
val ner_model = NerDLModel.pretrained("wikiner_68_300", "pt")  
    .setInputCols(Array("document", "token", "embeddings"))  
    .setOutputCol("ner")  
...  
val pipeline = new Pipeline().setStages(Array(documentAssembler, sentenceDetector, 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.johnsnowlabs.com/public/NER_PT/



Escolha o modelo pré-treinado para testar
wikiner_6B_300

Rótulos de entidade
LOC X PER X
ORG X MISC X

Explicar etiquetas NER

Tente você mesmo:
[Open in Colab](#)

Reconhecer pessoas, lugares, organizações e outras entidades

Treinado pelo anotador NerDLApproach com Char CNNs - BiLSTM - Incorporações CRF e GloVe no corpus WikiNER e suporta a identificação de entidades PER, LOC, ORG e MISC

Escolha Texto de Amostra

William Henry Gates III (nascido em 28 de outubro de 1955) é um magnata americano de negócios, de...

Texto anotado com entidades nomeadas identificadas

William Henry Gates III (nascido em 28 de outubro de 1955) é um magnata americano de negócios, desenvolvedor de software, investidor e filantropo.

Ele é mais conhecido como co-fundador da Microsoft Corporation. Durante sua carreira na Microsoft, Gates ocupou os cargos de presidente, diretor executivo (CEO), presidente e diretor de arquitetura de software, além de ser o maior acionista individual até maio de 2014. Ele é

Live Demo  Open in Colab  Download

Live Demo

 Open in Colab

 Download

Colab Notebooks



colab.research.google.com/github/JohnSnowLabs/spark-nlp-workshop/blob/master/tutorials/streamlit_notebooks/NER_PT.ipynb

File Edit View Insert Runtime Tools Help

Table of contents

Detect entities in Portuguese text

Colab Setup

Start the Spark session

Select the DL model

Some sample examples

Define Spark NLP pipeline

Run the pipeline

Visualize results

+ Section

John Snow LABS

Open in Colab

▼ Detect entities in Portuguese text

▼ 1. Colab Setup

```
[ ] # Install Java
! apt-get update -qq
! apt-get install -y openjdk-8-jdk-headless -qq > /dev/null
! java -version

# Install pyspark
! pip install --ignore-installed -q pyspark==2.4.4
```

Live Demo

Open in Colab

Download

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

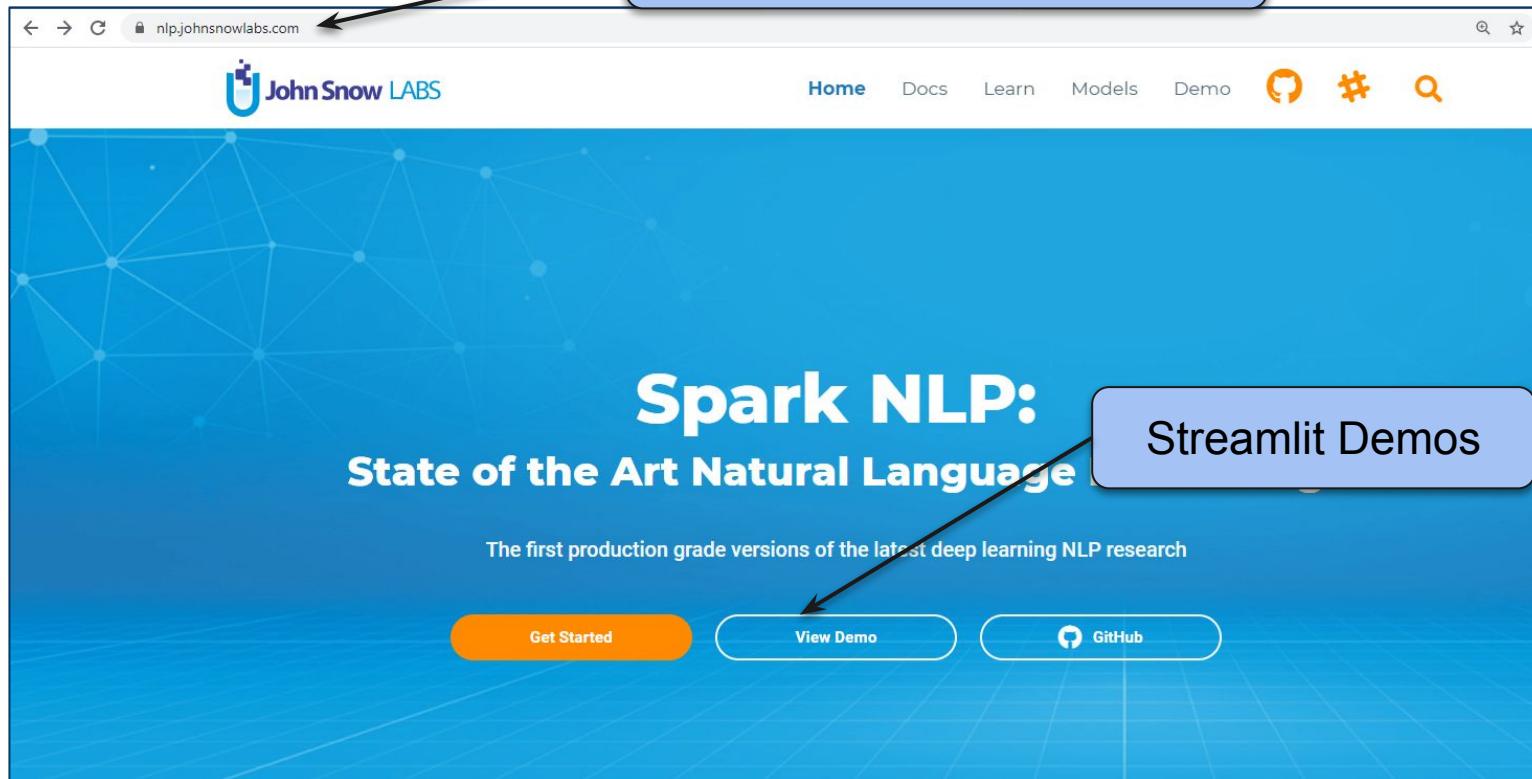
Micro-average prec: 0.91378844, rec: 0.9259801, f1: 0.9198439

Streamlit Demos

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

Main Page

http://nlp.johnsnowlabs.com/



The screenshot shows the main page of the John Snow LABS website at <http://nlp.johnsnowlabs.com/>. The page features a blue header with the logo and navigation links: Home, Docs, Learn, Models, Demo, GitHub, Hash, and Search. The main content area has a blue background with a network graph pattern. It prominently displays "Spark NLP: State of the Art Natural Language" and describes it as "The first production grade versions of the latest deep learning NLP research". Three calls-to-action are visible: "Get Started" (orange button), "View Demo" (white button), and "GitHub" (button). A callout box highlights the "Streamlit Demos" link.

nlp.johnsnowlabs.com

John Snow LABS

Home Docs Learn Models Demo

GitHub Hash Search

Spark NLP:
State of the Art Natural Language

The first production grade versions of the latest deep learning NLP research

Get Started View Demo GitHub

Streamlit Demos

Streamlit Demos

Spark NLP in Action



Open Source



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

Open Source
FREE

Languages
FREE

Healthcare

Spark OCR

De-identification

Languages



Detect language



Recognize entities in
English text



Recognize entities in
French text



Recognize entities in
German text

Live Demo

Live Demo

Live Demo

Live Demo

Colab Netbook

Colab Netbook

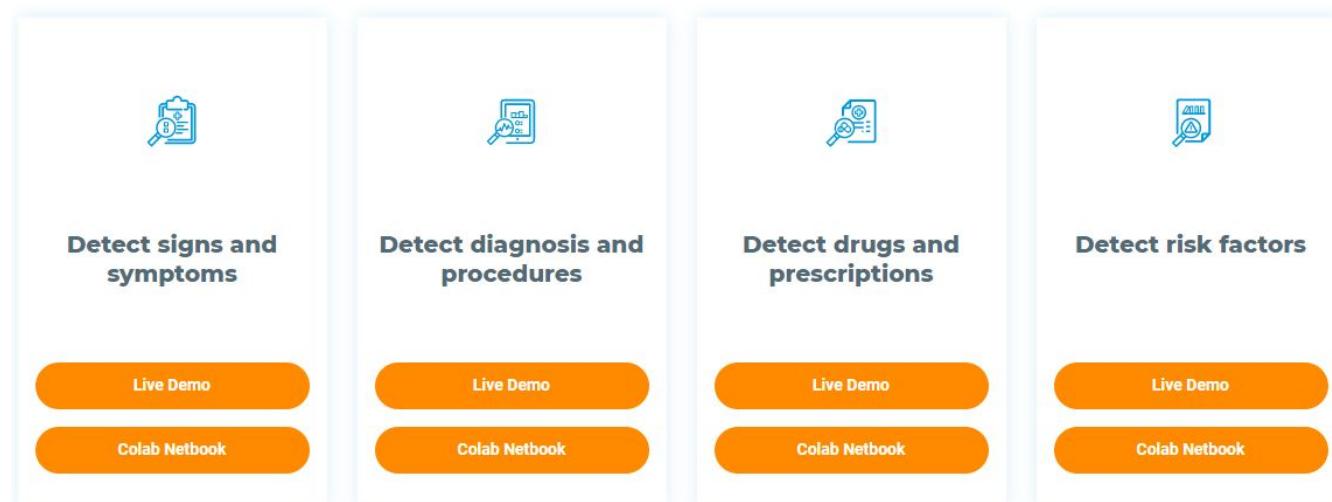
Colab Netbook

Colab Netbook

Healthcare



Healthcare



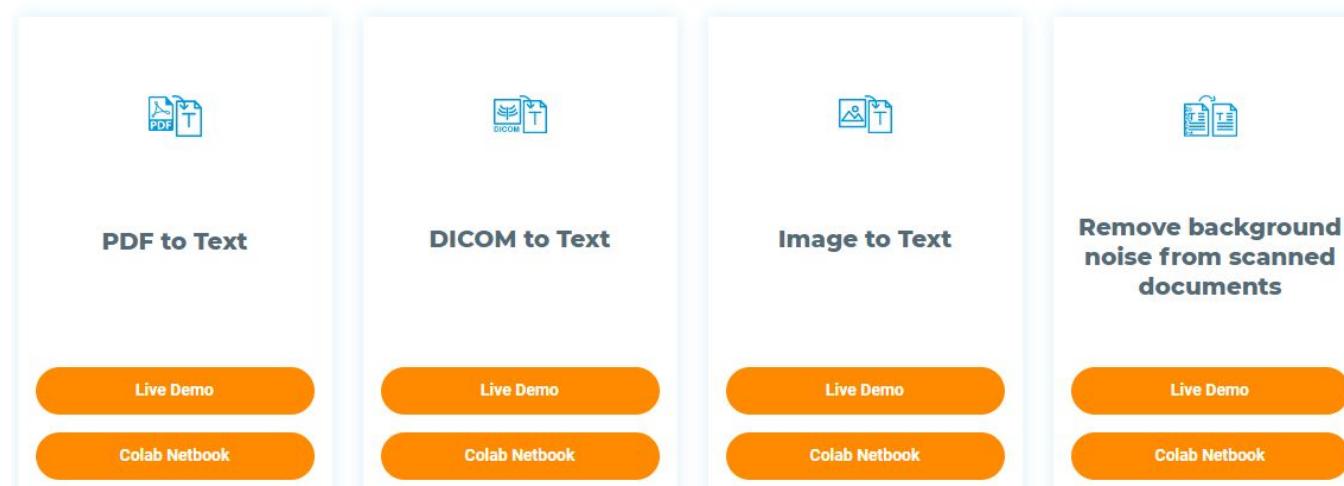
The image displays four cards, each representing a different task in the Healthcare domain:

- Detect signs and symptoms**: Accompanied by a magnifying glass icon over a clipboard. Buttons below provide "Live Demo" and "Colab Netbook".
- Detect diagnosis and procedures**: Accompanied by a magnifying glass icon over a medical device. Buttons below provide "Live Demo" and "Colab Netbook".
- Detect drugs and prescriptions**: Accompanied by a magnifying glass icon over a prescription bottle. Buttons below provide "Live Demo" and "Colab Netbook".
- Detect risk factors**: Accompanied by a magnifying glass icon over a document. Buttons below provide "Live Demo" and "Colab Netbook".

Spark OCR



Spark OCR



The page displays four main features of the Spark OCR service:

- PDF to Text**: Converts PDF files into text. Includes "Live Demo" and "Colab Netbook" buttons.
- DICOM to Text**: Converts DICOM files into text. Includes "Live Demo" and "Colab Netbook" buttons.
- Image to Text**: Converts images into text. Includes "Live Demo" and "Colab Netbook" buttons.
- Remove background noise from scanned documents**: A feature for cleaning scanned documents. Includes "Live Demo" and "Colab Netbook" buttons.

De-Identification



De-identification



Deidentify structured data



Deidentify free text documents



Deidentify DICOM documents



De-identify PDF documents - HIPAA Compliance

[Live Demo](#)

[Colab Netbook](#)

[Live Demo](#)

[Colab Netbook](#)

[Live Demo](#)

[Colab Netbook](#)

[Live Demo](#)

[Colab Netbook](#)

GitHub Repositories



→ spark-nlp

→ spark-nlp-workshop

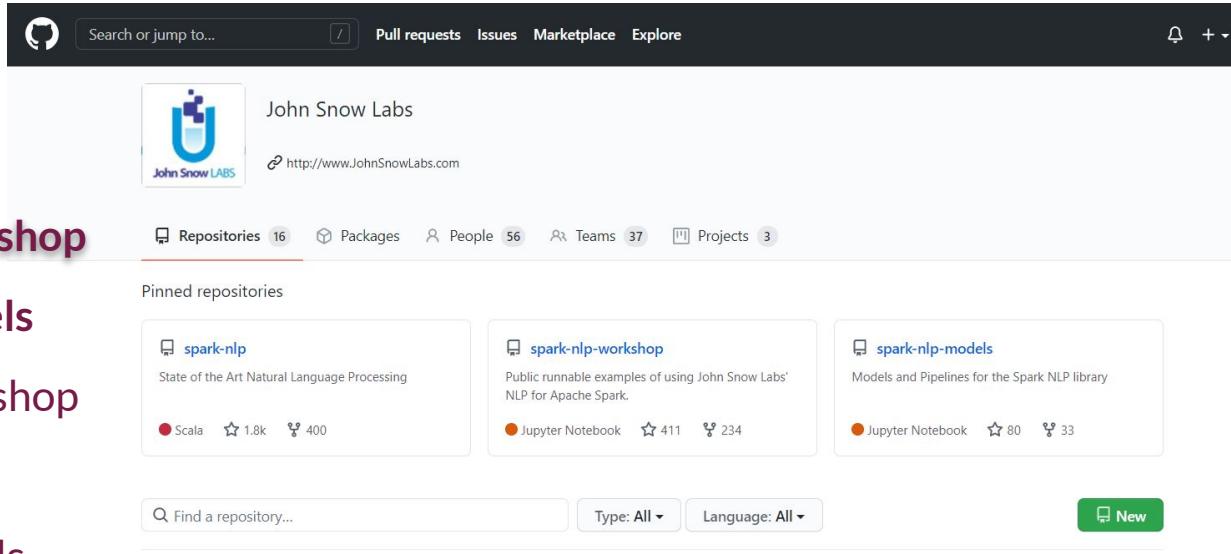
→ spark-nlp-models

→ spark-ocr-workshop

→ nlu

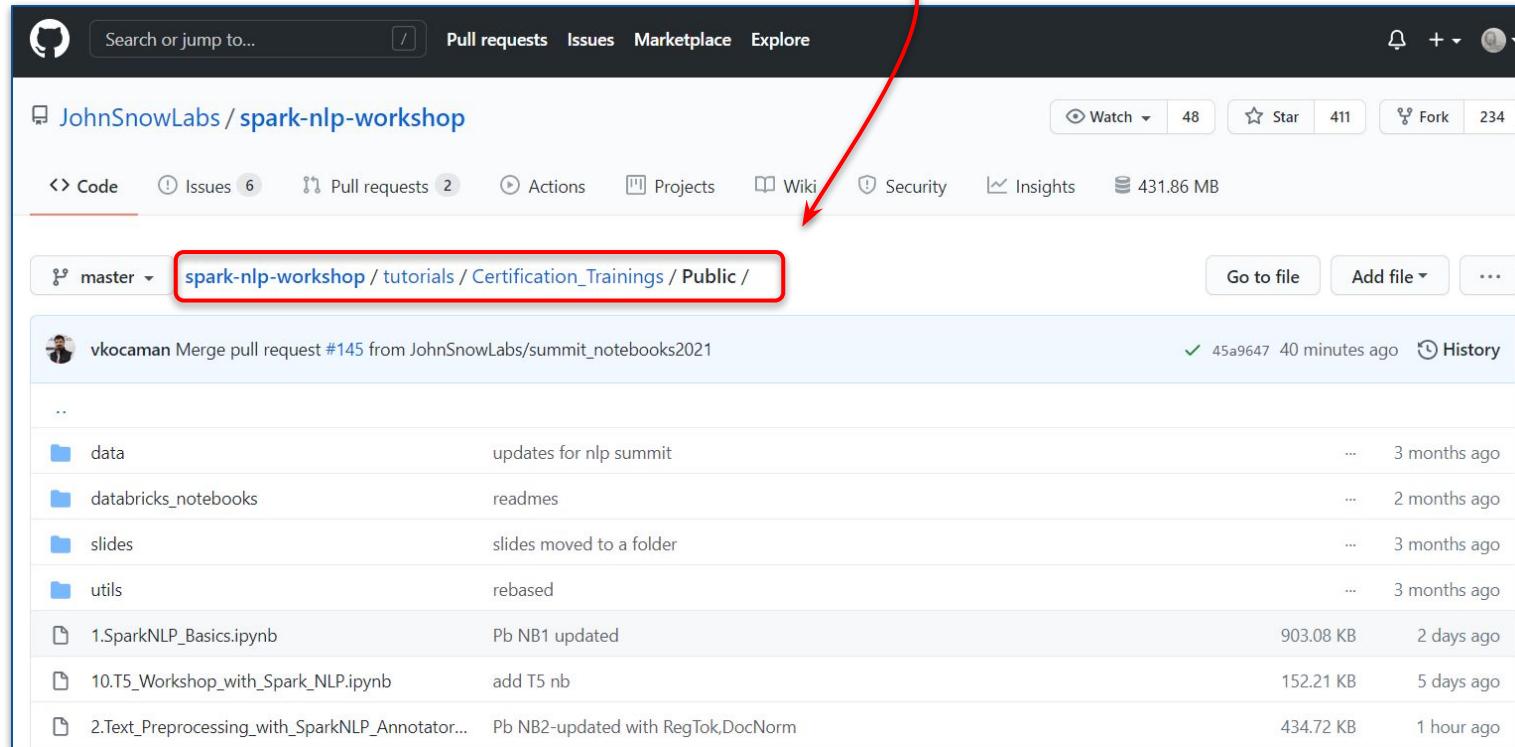
→ spark-nlp-models

→ spark-nlp-display



The screenshot shows the GitHub profile of 'John Snow Labs'. At the top, there's a search bar with placeholder text 'Search or jump to...', and navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. A notification bell icon and a '+' button are also present. Below the header, the 'John Snow Labs' organization profile is shown, featuring its logo, a link to their website (<http://www.JohnSnowLabs.com>), and statistics for repositories (16), packages (1), people (56), teams (37), and projects (3). A 'Pinned repositories' section displays three repositories: 'spark-nlp' (State of the Art Natural Language Processing, Scala, 1.8k stars, 400 forks), 'spark-nlp-workshop' (Public runnable examples of using John Snow Labs' NLP for Apache Spark, Jupyter Notebook, 411 stars, 234 forks), and 'spark-nlp-models' (Models and Pipelines for the Spark NLP library, Jupyter Notebook, 80 stars, 33 forks). A search bar at the bottom allows users to 'Find a repository...', with dropdown menus for 'Type: All' and 'Language: All'. A green 'New' button is located in the bottom right corner.

Certification Training



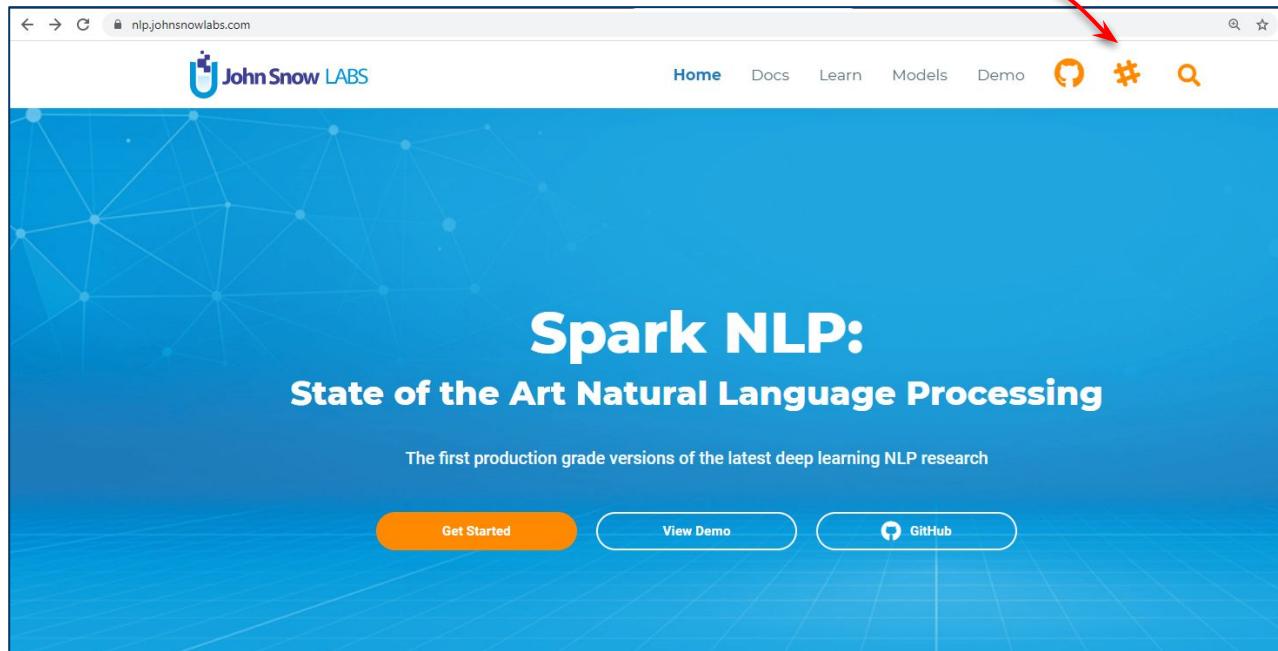
A screenshot of a GitHub repository page for 'JohnSnowLabs / spark-nlp-workshop'. The repository has 48 pull requests, 411 stars, and 234 forks. A red arrow points from the top right towards the breadcrumb navigation bar, which shows the path: 'spark-nlp-workshop / tutorials / Certification_Trainings / Public /'. Below the path, a list of files and commits is displayed.

File/Commit	Description	Last Modified
..		40 minutes ago
data	updates for nlp summit	3 months ago
databricks_notebooks	readmes	2 months ago
slides	slides moved to a folder	3 months ago
utils	rebased	3 months ago
1.SparkNLP_Basics.ipynb	Pb NB1 updated	2 days ago
10.T5_Workshop_with_Spark_NLP.ipynb	add T5 nb	5 days ago
2.Text_Preprocessing_with_SparkNLP_Annotator...	Pb NB2-updated with RegTok,DocNorm	1 hour ago

Spark-NLP Slack Channels

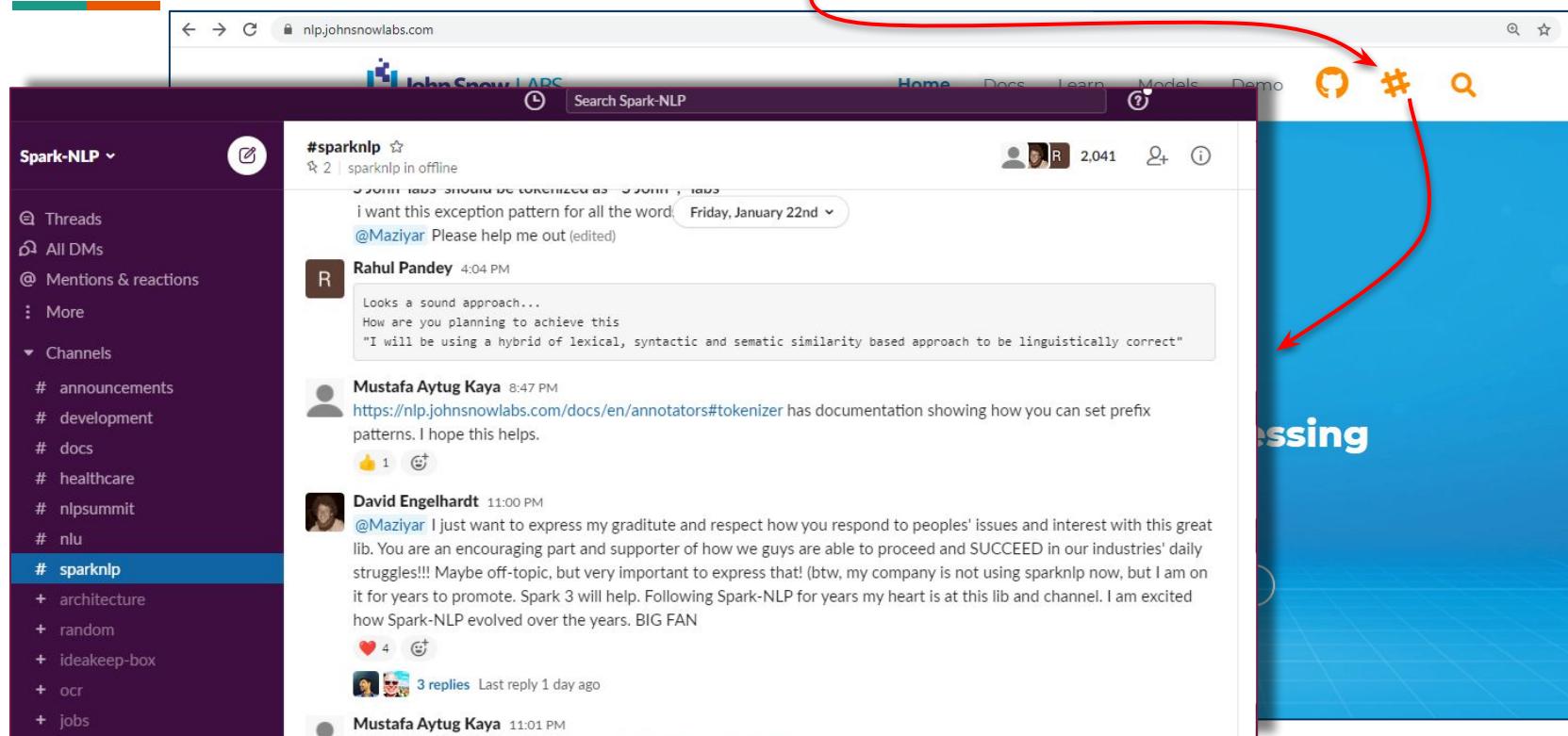


- #sparknlp
- #healthcare
- #nlu
- #ocr
- #jobs



Spark-NLP Slack Channels

spark-nlp.slack.com



The screenshot shows a web browser displaying the URL nlp.johnsnowlabs.com. The page has a purple header with the John Snow LABS logo and a search bar. A red arrow points from the text "spark-nlp.slack.com" at the top right to the Slack interface. The Slack interface shows a channel named "#sparknlp". The sidebar on the left lists various channels, with "# sparknlp" being the active one, indicated by a blue background. The main area shows messages from users like Maziyar, Rahul Pandey, Mustafa Aytug Kaya, and David Engelhardt. The interface includes standard Slack features like a message input field, a search bar, and a sidebar with user icons.

#sparknlp ☆
2 | sparknlp is offline

John Snow LABS should be tokenized as "John", "Snow", "LABS"
I want this exception pattern for all the words. Friday, January 22nd
@Maziyar Please help me out (edited)

Rahul Pandey 4:04 PM
Looks a sound approach...
How are you planning to achieve this
"I will be using a hybrid of lexical, syntactic and semantic similarity based approach to be linguistically correct"

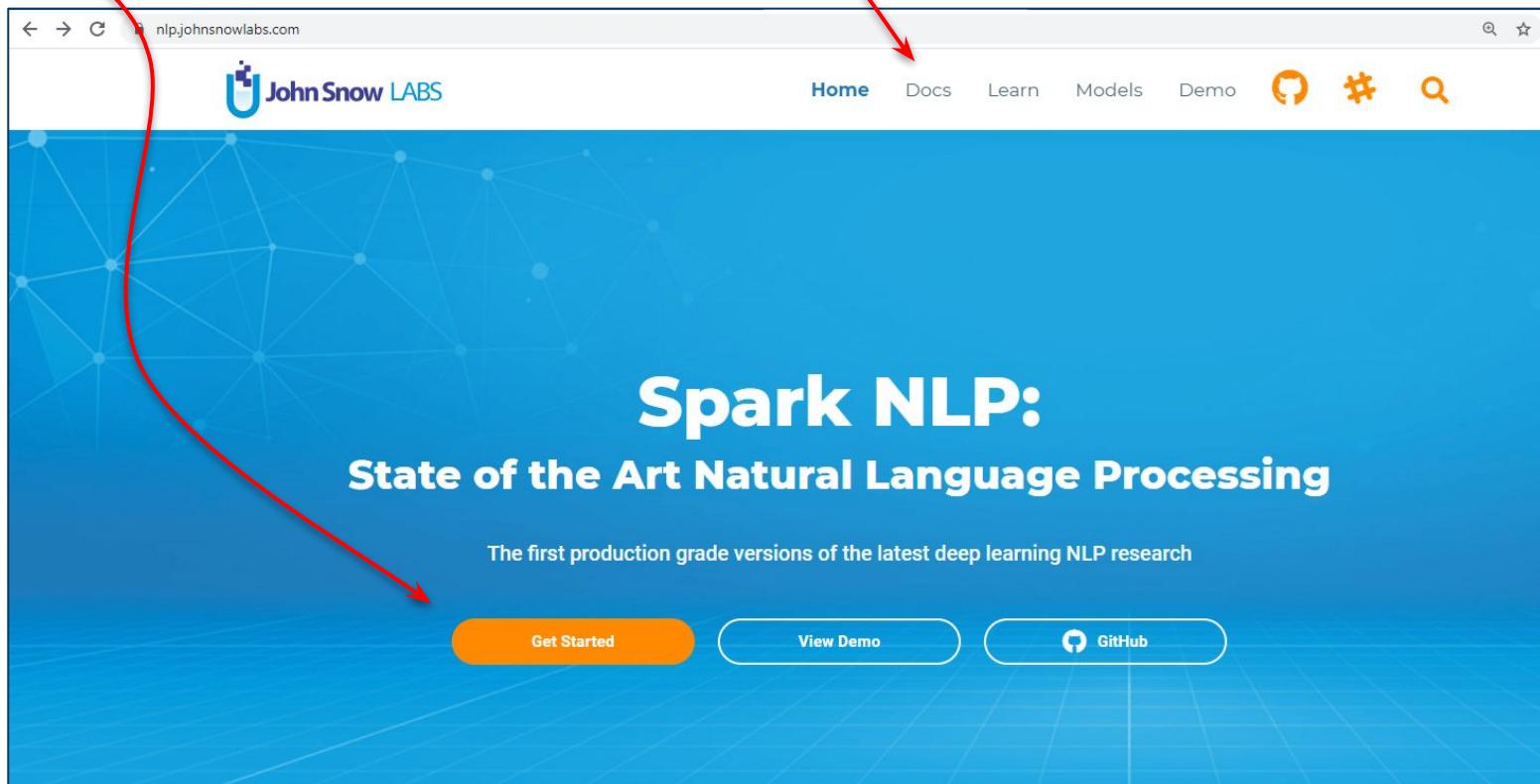
Mustafa Aytug Kaya 8:47 PM
<https://nlp.johnsnowlabs.com/docs/en/annotators#tokenizer> has documentation showing how you can set prefix patterns. I hope this helps.
1 thumbs up

David Engelhardt 11:00 PM
@Maziyar I just want to express my gratitude and respect how you respond to people's issues and interest with this great lib. You are an encouraging part and supporter of how we guys are able to proceed and SUCCEED in our industries' daily struggles!!! Maybe off-topic, but very important to express that! (btw, my company is not using sparknlp now, but I am on it for years to promote. Spark 3 will help. Following Spark-NLP for years my heart is at this lib and channel. I am excited how Spark-NLP evolved over the years. BIG FAN)
4 hearts, 1 thumbs up

3 replies Last reply 1 day ago

Mustafa Aytug Kaya 11:01 PM

Docs



The screenshot shows the homepage of the nlp.johnsnowlabs.com website. The page has a blue header with the John Snow LABS logo and navigation links for Home, Docs, Learn, Models, Demo, and a search bar. Below the header is a large blue banner featuring a network graph background and the text "Spark NLP: State of the Art Natural Language Processing". A red curved arrow points from the "Docs" heading in the top left towards the "Docs" link in the header. Another red arrow points from the "Docs" link in the header down to the "Docs" link in the banner.

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



← → C 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

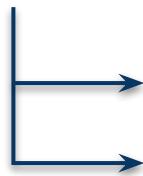
If you prefer learning by example, check this repository:

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.

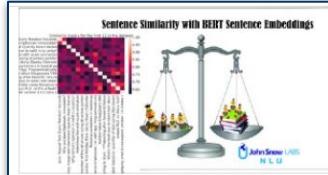
Articles-Blog Posts



Medium-Spark NLP (<https://medium.com/spark-nlp>)

Blog Posts

Medium-Spark NLP

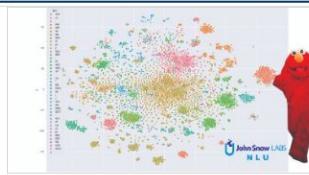


Easy sentence similarity with BERT Sentence Embeddings using John Snow Labs NLU

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.

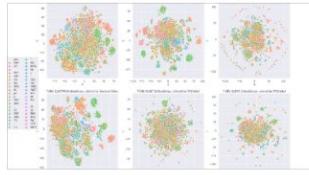
The NLU miracle allows us to produce a perfect CoNLL file and a perfect CoNLL file makes the Turkish NER model perfect. This is the first...



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>

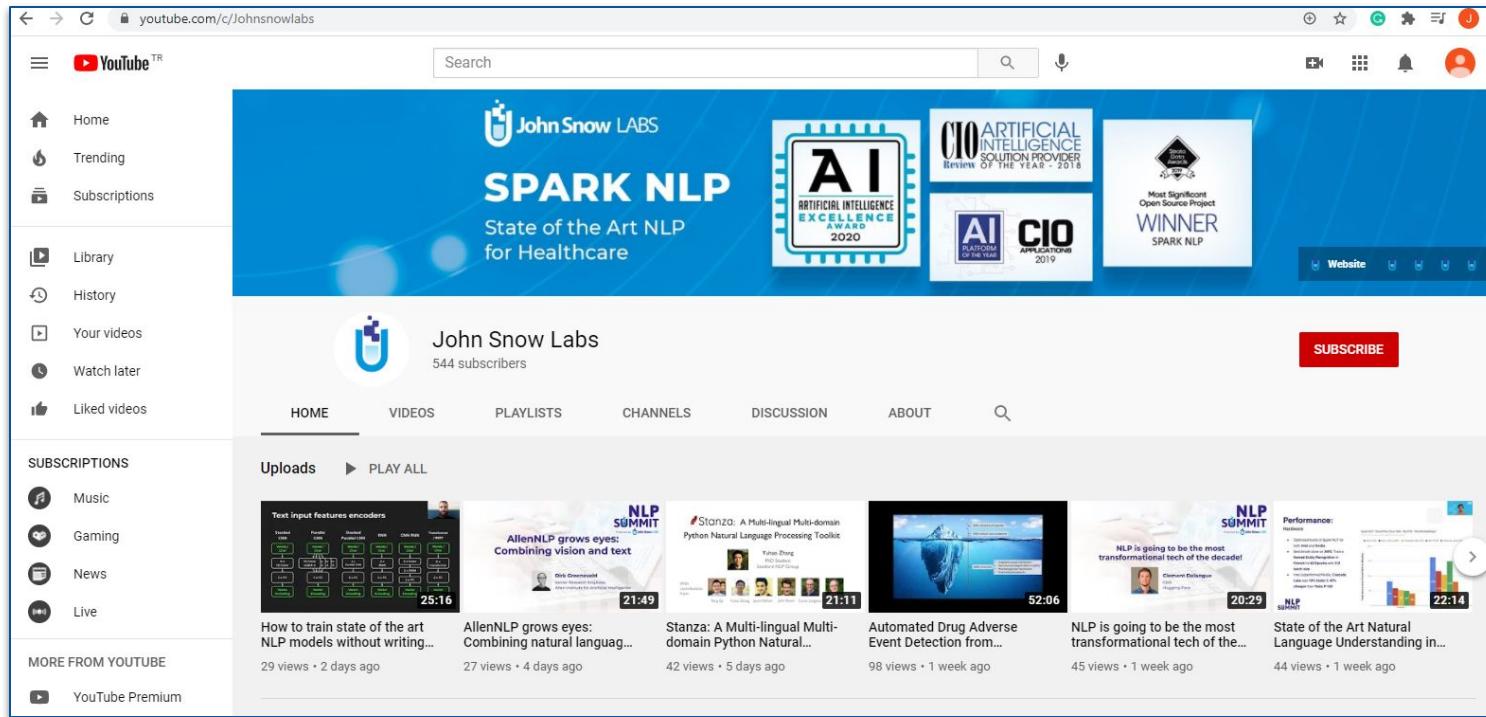
Article for NER and text classification in Spark NLP

<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



The screenshot shows the YouTube channel page for "John Snow Labs". The channel banner features the text "SPARK NLP" and "State of the Art NLP for Healthcare". It also displays several awards: "CIO ARTIFICIAL INTELLIGENCE SOLUTION PROVIDER OF THE YEAR - 2018", "AI EXCELLENCE AWARD 2020", "CIO PLATFORM OF THE YEAR 2019", and "CIO APPLICATIONS 2019". The channel has 544 subscribers and a "SUBSCRIBE" button.

HOME

VIDEOS

PLAYLISTS

CHANNELS

DISCUSSION

ABOUT

SUBSCRIPTIONS

Uploads ► **PLAY ALL**

Video Title	Length	Views	Last Updated
Text input features encoders	25:16	29 views	2 days ago
AllenNLP grows eyes: Combining vision and text	21:49	27 views	4 days ago
Stanza: A Multi-lingual Multi-domain Python Natural Language Processing Toolkit	21:11	42 views	5 days ago
Automated Drug Adverse Event Detection from...	52:06	98 views	1 week ago
NLP is going to be the most transformational tech of the...	20:29	45 views	1 week ago
Performance: State of the Art Natural Language Understanding in...	22:14	44 views	1 week ago

MORE FROM YOUTUBE

YouTube Premium

```
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  
 '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, 0, {'word': 'Record', 'confidence': '1.0'}),  
 Annotation(named_entity, 7, 10, 0, {'word': 'date', 'confidence': '1.0'}),  
 Annotation(named_entity, 11, 11, 0, {'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, 0, {'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, 0, {'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, 0, {'word': 'HISTORY', 'confidence': '1.0'})]
```

