John Snow LABS

100% OPEN SOURCE

Accelerating NLP Workflows with Spark NLP 6.x

A comprehensive overview of features, modules, capabilities, and architecture

State-of-the-art NLP at scale with 80K+ pretrained models in 200+ languages

NLP & AI

Apache Spark

Scalable

Multi-lingual

Presentation Agenda

Today we'll cover five key areas of Spark NLP technology and performance.

- Spark NLP Overview
- Spark NLP HW Acceleration Benchmarks
- Spark NLP 6.0 Multimodal AI at Scale
- Spark NLP 6.1 Expanded Data Ingestion
- Demo Notebooks

What is Spark NLP?

Spark NLP is a full-scale NLP library built on top of Apache Spark, providing distributed processing, scalability, and seamless integration with Spark ML pipelines.

100% Open Source - Apache 2.0 license with no vendor lock-in

Natively Scalable - Built on Apache Spark for distributed processing

Multi-lingual Support - 200+ languages supported out of the box

Full Language Support - Python, Scala, and Java APIs

Production-Ready - Designed for large-scale, enterprise deployments

State-of-the-Art Performance

93% F1
CONLL 2003 (NER)

90% F1
ONTONOTES 5.0 (NER)

APACHE 2.0 LICENSED

162.02M
TOTAL DOWNLOADS

80,000+
PRETRAINED MODELS

55,000+
PRETRAINED PIPELINES

200+
LANGUAGES

100%
OPEN SOURCE

Most Widely Used NLP Library
IN ENTERPRISE - GRADIENT FLOW 2021

Architecture & Workflow

Spark NLP runs on Apache Spark, enabling parallel, distributed data processing with seamless integration into Spark ML pipelines and DataFrames.

System Requirements

Java: Java 8 and 11

Apache Spark: 3.3.x, 3.4.x, 3.5.x

Built on Apache Spark - Leverages distributed computing for scalable NLP

Pipeline Architecture - Chain annotators for complete workflows

DataFrame Integration - Annotations become DataFrame columns

Spark ML Compatible - Seamless integration with Spark ML pipelines

Streaming Support - Process real-time text streams at scale

Production-Grade - Handles millions of documents efficiently

Spark NLP Pipeline Flow

1. **DocumentAssembler**
   Converts raw text into Document annotations
2. **SentenceDetector**
   Splits documents into sentences
3. **Tokenizer**
   Breaks sentences into tokens
4. **Annotators (Embeddings/POS/Lemma)**
   Apply transformations (BERT, Word2Vec, etc.)
5. **Model (NER/Classification)**
   Apply trained models for predictions
6. **Output (DataFrame)**
   Structured results in Spark DataFrame columns

Annotation Structure

```
Annotation(annotatorType, begin,end,result, metadata, embeddings)
```

Core Capabilities Overview

Spark NLP provides a comprehensive suite of NLP capabilities built on Apache Spark, enabling scalable text processing from basic preprocessing to advanced generative AI tasks.

Text Preprocessing
Tokenization, normalization, POS tagging, dependency parsing

Embeddings
Word2Vec, GloVe, BERT, RoBERTa, sentence embeddings

Named Entity Recognition
Deep learning NER, transformer-based, zero-shot NER

Classification
Text classification, sentiment analysis, multi-lingual

Question Answering
Extractive QA, table QA, context-based answering

Summarization
Document summarization with T5 and transformers

Generative Models
GPT-2, T5 generation, text-to-SQL, LLM integration

Multimodal
Translation, image classification, speech recognition

Specialized Tasks
Spell checking, keyword extraction, graph extraction

Training & Fine-tuning
Custom model training, transfer learning, scalable

Multi-language
200+ languages, multilingual models and embeddings

Model Hub
80k+ pretrained models, 55k+ pipelines

All capabilities scale seamlessly with Apache Spark for production-grade deployments

Text Preprocessing & Basic Annotations

The foundational layer for NLP workflows. Spark NLP provides rich annotators and transformers for comprehensive text preprocessing and linguistic analysis.

Core Annotators

- DocumentAssembler
- Tokenizer
- DocumentSplitter
- RegexMatcher
- SentenceDetector
- SentenceDetectorDL
- EntityRuler
- DateMatcher
- Normalizer
- Stemmer
- Lemmatizer
- POSTagger
- DependencyParser
- TypedDependencyParser
- StopWordsCleaner
- NGramGenerator

Annotator Types

Approach

Extends Estimators. Trained through fit(). Examples: NerDLApproach, ClassifierDLApproach

Model

Extends Transformers. Transform DataFrames through transform(). Examples: NerDLModel, TokenizerModel

Input/Output Types: Each annotator has specific input/output annotation types (DOCUMENT, TOKEN, SENTENCE, etc.) that must match in pipeline configuration

Named Entity Recognition (NER)

Extract structured information from unstructured text by identifying and classifying named entities such as people, organizations, locations, products, and more.

Rule-Based & Statistical NER

Pattern matching, EntityRuler with custom dictionaries and regex patterns for domain-specific entities

Deep Learning NER (NERDL)

Neural network models for token classification, supporting custom training with labeled datasets

Transformer-Based NER

BERT, RoBERTa, and other transformer architectures for state-of-the-art entity recognition

Zero-Shot NER

Detect entities without dedicated training sets using large language models

Key Capabilities

- Pre-trained models for common entity types
- Custom entity training with your labeled data
- Multi-lingual entity recognition (200+ languages)
- Entity linking and disambiguation
- Nested entity detection support
- Confidence scores for predictions

Common Entity Types

Extract structured data from any domain

- PERSON
- ORGANIZATION
- LOCATION
- DATE
- PRODUCT
- MONEY
- MEDICAL
- LEGAL

Text Classification & Sentiment Analysis

Classify text units into categories or predict sentiment with deep learning and transformer-based approaches.

ClassifierDL - Deep learning-based text classification with neural networks

Transformer Sequence Classification - BERT, RoBERTa, DistilBERT for advanced classification

Sentiment Analysis - Pre-trained pipelines for movie reviews, product feedback, social media

Multi-lingual Classification - Support for 200+ languages with XLM-RoBERTa

Zero-Shot Classification - Classify without dedicated training data

CLASSIFIERDL | BERT | ROBERTA | ZERO-SHOT

Key Use Cases

- Spam Detection & Filtering
- Topic Categorization
- Sentiment Scoring
- Intent Detection
- Complaint Classification

Question Answering & Summarization

Leverage transformer-based models for advanced question answering and text summarization tasks, enabling intelligent document processing and information extraction.

Extractive QA - Extract answer spans from context documents

Table Question Answering - Query structured tabular data with natural language

Text Summarization - Generate concise summaries from longer documents

T5 Models - Text-to-Text Transfer Transformer for multiple tasks

FAQ Systems - Build intelligent chatbot backends for customer support

TRANSFORMER-POWERED

Standard QA Pipeline

Given a context text and a query, extract the precise answer span using pre-trained BERT, RoBERTa, or domain-specific models.

Summarization with T5

Generate abstractive or extractive summaries from long documents, reports, or articles using Google's T5 architecture.

Use Cases

FAQ automation, document summarization, chatbot backends, research paper analysis, and intelligent information retrieval systems

Training Custom Models

Spark NLP provides full support for training and fine-tuning custom models on your domain-specific data, leveraging Spark's scalability for large-scale training workflows.

NERDL Training - Custom Named Entity Recognition with deep learning models

ClassifierDL - Train custom text classifiers for your specific use cases

Custom Embeddings - Train Word2Vec, doc2vec on domain-specific corpora

Transfer Learning - Fine-tune transformer models (BERT, RoBERTa) for your tasks

Multi-lingual Training - Adapt models to any language or domain

SPARK ML INTEGRATION

Training Workflow

Prepare Training Data → Configure Pipeline → Train & Evaluate → Deploy Model

Scalable Training

Leverage Spark clusters for distributed training

Model Persistence

Save, load, and version your trained models

Hyperparameter Tuning

Optimize model performance with grid search

Evaluation Metrics

Built-in metrics for model performance assessment

Multi-Language, Scale & Internationalization

Spark NLP is built for global scale and enterprise deployment, supporting hundreds of languages and leveraging distributed computing for massive text processing workloads.

200+ Languages - Multi-lingual models, embeddings, and pipelines for global applications

Distributed Processing - Spark-based parallel computing for large-scale text corpora

Multi-Language APIs - Full Python, Java, and Scala support for JVM environments

Cluster-Ready - Deploy on Spark clusters, handle streaming data and big data workflows

CPU & GPU - Flexible hardware support for transformer model training and inference

200+

LANGUAGES

3

API LANGUAGES

Production-Grade Platforms

Deploy seamlessly on major cloud and cluster platforms

- Databricks
- AWS EMR
- Apache Spark
- Google Cloud
- Azure
- On-Premise

Pre-built Pipelines & Model Hub

Spark NLP provides thousands of pretrained models and pipelines that accelerate development and reduce time-to-value for production NLP applications.

One-Line Loading - Load any pretrained pipeline with a single command

Composable Pipelines - Chain multiple annotators for complex workflows

Multi-language Coverage - Models available in 200+ languages

Rapid Prototyping - Test ideas quickly without training from scratch

Easy Customization - Fine-tune pretrained models for domain-specific needs

```
pipeline = PretrainedPipeline('explain_document_dl', lang='en')
result = pipeline.annotate("Your text here")
```

FAST DEPLOYMENT READY

80,000+
PRETRAINED MODELS

55,000+
PRETRAINED PIPELINES

200+
LANGUAGES

100%
OPEN ACCESS

Popular Pipeline Examples

- explain_document_dl - Full text analysis pipeline
- analyze_sentimentdl_use_imdb - Sentiment analysis
- entity_recognizer_md - Named entity recognition
- check_spelling - Spell checking pipeline

Use Cases & Industry Applications

Spark NLP powers critical NLP workloads across diverse industries, enabling organizations to extract insights, automate workflows, and enhance decision-making at scale.

Information Extraction

Extract structured data from unstructured documents: contracts, medical records, financial reports, legal filings

Document Classification

Automated categorization: complaint routing, news topic tagging, document type identification, sentiment scoring

Conversational AI

Question answering systems, chatbots for customer support, FAQ automation, intent detection

Text Summarization

Automated report generation, news summarization, document condensation for rapid review

Healthcare

- Clinical entity extraction
- Medical coding automation
- Patient record de-identification
- Adverse event detection
- Clinical decision support

Legal

- Contract analysis & review
- Legal entity recognition
- E-discovery automation
- Compliance monitoring
- Case law summarization

Finance

- Financial sentiment analysis
- Risk assessment automation
- Fraud detection patterns
- Regulatory compliance
- Market intelligence extraction

Customer Support

- Ticket classification & routing
- Sentiment & urgency detection
- Automated response generation
- Knowledge base QA systems
- Multi-lingual support

Course & Learning Path

Spark NLP for Data Scientists

Udemy course by John Snow Labs covering state-of-the-art NLP solutions from fundamentals to advanced applications

20,000+
PRETRAINED MODELS

250+
LANGUAGES

What You'll Learn

- Utilize 20,000+ State-of-the-Art NLP models in 200+ languages
- Train & tune your own NLP models on custom datasets
- Perform NLU tasks in one line - generate, summarize, answer
- Deploy models as APIs with NLP Server Docker container

FREE CERTIFICATION OPPORTUNITY

IT & Software › IT Certifications › Natural Language Processing (NLP)

Spark NLP for Data Scientists

Unlock your NLP power with Spark NLP, the most popular NLP library in enterprises

Created by Ace Vo, David Talby, Jiri Dobes, Veysel Kocaman

Last updated 6/2023 English English [Auto]

Premium

Access this top-rated course, plus 30,100+ more top-rated courses, with a Udemy plan. See Plans & Pricing

4.7
66 ratings

318
learners

What you'll learn

- Utilize 20,000+ State-of-the-Art NLP models in 200+ languages
- Train & tune your own NLP models by leveraging the Spark NLP's pre-defined classifier architecture on your own datasets
- Perform popular NLU tasks in one line of code - like generate texts, summarize texts, answer questions
- Deploy models as API's with NLP Server, a Docker container that contains all Spark NLPs capabilities

Explore related topics

Natural Language Processing (NLP) | IT Certifications | IT & Software

This course includes:

- 12.5 hours on-demand video
- Access on mobile and TV
- Certificate of completion

Personal | Teams

This Premium course is included in plans

Subscribe to Udemy's top courses

Get this course, plus 30,100+ of our top-rated courses, with Personal Plan. Learn more

Try Personal Plan for free

Starting at $10.00 per month after trial

Cancel anytime

or

$19.99

Add to cart

Buy now

30-Day Money-Back Guarantee

Full Lifetime Access

Share | Gift this course | Apply Coupon

Deployment & Production Readiness

Spark NLP is production-grade and enterprise-ready, with comprehensive support for deployment, scalability, and integration into existing data infrastructure.

Serializable Pipelines - Save, load, and reuse pipelines across environments

Spark DataFrame Integration - NLP annotations become DataFrame columns seamlessly

CPU/GPU Support - Optimized for both CPU and GPU environments

Model Persistence - Store and version control trained models efficiently

Streaming Support - Process real-time data with Spark Streaming integration

Performance Optimized - Distributed processing for millions of documents

Databricks

AWS EMR

Google Cloud

Azure HDInsight Fabric

Enterprise-Grade Deployment

Deploy on-premise or in the cloud with full support for major platforms and seamless integration with existing infrastructure

Community & Support

Active GitHub community, Slack channel, comprehensive documentation, and workshop notebooks

Spark NLP Hardware Acceleration

Spark NLP runs natively and efficiently on Apache Spark, achieving superior performance both on single machines and distributed clusters. Hardware acceleration enables massive optimization for deep learning-based tasks.

Key Acceleration Features:

- **GPU Acceleration (up to 8.1x faster)**
- **Intel oneDNN CPU Optimization (up to 97% faster)**
- **Zero-Code Change Implementation**

Perfect for transformer models, embeddings, and language model tasks

Performance Comparison

CPU Baseline

1.0x

CPU + oneDNN

~2x faster

GPU (CUDA)

Up to 8x faster

Enabling hardware acceleration requires minimal changes - just start with GPU flag or set environment variable

GPU Acceleration in Spark NLP

Run Spark NLP seamlessly on GPU with zero code changes to your existing pipelines

Up to 8x speed improvement for transformer models (BERT, RoBERTa, XLM, etc.)

Ideal for computationally intensive tasks: embeddings, text classification, NER, and language understanding

Automatic fallback to CPU for operations not supported by GPU

8.1x
DeBERTa Base speedup

7.6x
DistilBERT speedup

7.4x
XLM-RoBERTa speedup

Zero-Code GPU Activation

Method 1: Spark Session Start

```
# Python import sparknlp # Enable GPU with one parameter spark = sparknlp.start(gpu=True)
```

Method 2: Maven Package

```
# Use GPU-enabled package spark-nlp-gpu
```

Requirements

NVIDIA software for GPU support:

- NVIDIA drivers ≥ 450.80.02
- CUDA Toolkit 11.2
- cuDNN SDK 8.1.0

GPU Performance Results

Benchmark comparing Spark NLP 3.4.3 vs. 4.0.0 performance on GPU shows significant improvements

Transformer models see up to 8x speedups with the latest GPU optimizations

Most models show a 5-7x performance boost when running on GPU

Model-Specific Performance Gains on GPU

DeBERTa Base **+713%** (8.1x faster)	**DistilBERT** **+659%** (7.6x faster)	**XLM-RoBERTa Base** **+638%** (7.4x faster)	**RoBERTa Base** **+560%** (6.6x faster)	**Albert Base** **+587%** (6.9x faster)
DeBERTa Large **+477%** (5.8x faster)	**XLNet Base** **+449%** (5.5x faster)	**XLM-RoBERTa Large** **+365%** (4.7x faster)	**Albert Large** **+332%** (4.3x faster)	**Longformer Base** **+788%** (8.9x faster)

Vision, Computer Vision & Multimodal Annotators

Image Classification / Vision-Only

Specialized models for image-only classification tasks:

SwinForImageClassification — A Vision Transformer (Swin) model for image classification

ViTForImageClassification — Vision Transformer (ViT) classification model

IMAGE ANALYSIS | TRANSFORMERS | VISION ONLY

Zero-Shot Image Classification

Classification without dedicated training data:

CLIPForZeroShotClassification — Image + text labels for zero-shot classification

ZERO-SHOT | CLIP

Multimodal Vision + Text Models

Explore Spark NLP's new AI capabilities for images and text combined:

LLama Family

- MLLamaForMultimodal

Google Models

- Gemma3ForMultiModal
- PaliGemmaForMultiModal

Microsoft & Alibaba

- Phi3Vision
- Qwen2VLTransformer

Other Leading Models

- SmolVLMTransformer
- JanusForMultiModal
- InternVLForMultiModal
- Florence2Transformer

Common capabilities across multimodal models:

- Visual question answering and captioning
- Object detection and segmentation

Audio & Automatic Speech Recognition (ASR) Annotators

ASR Models

High-performance speech recognition models for transcription and multilingual support:

Wav2Vec2ForCTC - Speech to text transcription using Connectionist Temporal Classification Ideal for high-accuracy English transcription tasks

HuBERTForCTC - Advanced speech to text model with self-supervised pretraining Robust against background noise and accent variations

WhisperForCTC - Multilingual ASR with translation capabilities Transcribe in native language or translate directly to English

SPEECH RECOGNITION TRANSCRIPTION MULTILINGUAL

Audio Utilities

Essential components for audio preprocessing and feature extraction:

AudioAssembler - Converts arrays of floats/doubles into AUDIO annotations Bridge between raw audio data and ASR model inputs

Audio feature helpers - Tools for spectrogram/Mel feature extraction Low-level audio processing for custom speech workflows

PREPROCESSING FEATURE EXTRACTION

POWERED BY TRANSFORMERS

Speech Integration

Seamless audio processing within Spark NLP workflows:

- Combine with text and vision for complete multimodal pipelines
- Process audio at scale using distributed Spark compute
- Build complex speech-to-text-to-action workflows

Spark NLP 6.0: Multimodal AI at Scale

Introducing native multimodal inference with GGUF quantized models directly within Spark pipelines

AutoGGUFVisionModel 6.0+ NEW V6.0+

Native multimodal inference for GGUF quantized models directly within Spark pipelines. Run powerful vision-language models like Qwen2 VL fully on-premises.

MULTIMODAL | ON-PREMISES | LOW MEMORY

- Batch & streaming multimodal inference without external servers
- Native Llama.cpp integration for optimized GPU/CPU performance
- Supports Qwen2 VL, Phi3Vision, Gemma3 & more

Complete Multimodal Ecosystem

This release boosts Spark NLP's multimodal processing power with Vision, Audio, and Text capabilities integrated in a single pipeline.

VISION + TEXT + AUDIO | ENTERPRISE-SCALE

- Vision-language models: image classification, zero-shot, and multimodal VQA
- Audio & ASR: speech recognition models including Wav2Vec2, HuBERT, and Whisper
- AutoGGUFReranker: semantic reranking for enterprise search at scale
- Full data privacy with on-premises deployment - no API dependencies

POWERED BY LLAMA.CPP & TRANSFORMERS

LLM & Text (GGUF/llama.cpp) Annotators

Text Generation NEW V6.0+

Advanced document-to-text generation capabilities:

AutoGGUFModel – Text completion and generation using GGUF quantized for all majors open source models including Phi4, LLama3.1, and Gemma 3

AutoGGUFVisionModel – Multimodal inference with image + text input to text output

TEXT COMPLETION MULTIMODAL BATCH PROCESSING ONNX OPENVINO

Text Embeddings

Generate vector representations for semantic search and analysis:

AutoGGUFEmbeddings – Fast text embedding generation using GGUF models

BGEEmbeddings – Sentence embeddings optimized for retrieval (non-GGUF)

SEMANTIC SEARCH VECTOR DATABASE

POWERED BY LLAMA.CPP & TRANSFORMERS

Reranking & Retrieval

Improve search relevance with semantic reranking:

AutoGGUFReranker – Semantic reranking for search and retrieval applications

Provides relevance scoring for search results based on semantic understanding

Efficient semantic reranking at scale with GPU support

Key Benefits & Practical Applications

Why Spark NLP 6.0 Matters

True Multimodal AI

Seamless image + text in a unified Spark pipeline with integrated workflows

Production-Ready

Batch & streaming, private deployments, no vendor lock-in

Open Source Support

Latest models: Qwen2 VL, Gemma, and more

MULTIMODAL | PRODUCTION-READY | OPEN SOURCE | SCALABLE

Practical Use Cases

Smart Document Processing
Extract insights from images and text in documents at scale

Image Captioning & Visual Q&A
Generate descriptions and answer questions about visual content

Enterprise Search & Content Retrieval
Enhanced semantic search across text and image content

Secure On-Premises AI at Scale
Process sensitive data privately within your infrastructure

ENTERPRISE-READY MULTIMODAL AI WITH NO EXTERNAL DEPENDENCIES

Spark NLP 6.1 Expanded Data Ingestion

An Introduction to Reader2Doc, Reader2Table, Reader2Image, and ReaderAssembler

Reader2Doc

Reader2Table

Reader2Image

ReaderAssembler

Unlock advanced multimodal ingestion and processing in distributed AI pipelines

Spark NLP 6.1.0 - 6.1.5

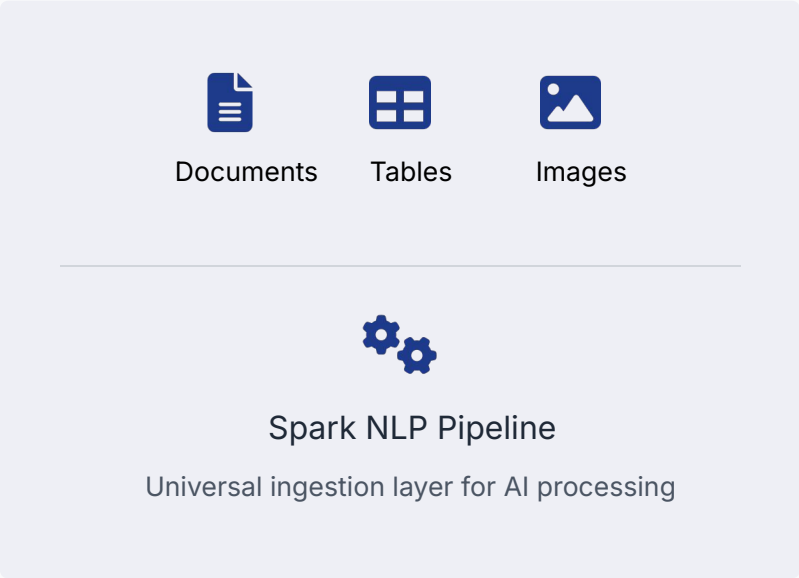
What Are Reader Annotators?

Reader annotators in Spark NLP enable seamless ingestion of diverse data formats, unifying multimodal content (text, tables, images) into the Spark NLP pipeline.

Core Purpose:

- **Simplify**
- **Scale**
- **Optimize**

Document processing for real-world AI workflows



Reader2Doc: Unified Document Ingestion

Unified annotator for ingesting a wide range of document formats into Spark NLP pipelines

Provides simplified, scalable document processing with distributed execution

Introduced in Spark NLP v6.1.0 as the foundation of the universal ingestion layer

PDF | DOC/DOCX | XLS/XLSX | PPT/PPTX

Email (.eml, .msg) | HTML | Markdown | Plain Text

v6.1.3 Enhancements

outputAsDocument

Concatenates all sentences into a single document

```
val reader = new Reader2Doc()
.setContentType("text/html")
.setContentPath("my/html/files")
.setOutputCol("html")
.setParams("outputAsDocument", "true")
```

excludeNonText

Filters out non-textual elements (e.g., tables, images)

```
val reader = new Reader2Doc()
.setContentType("text/html")
.setContentPath("my/html/files")
.setOutputCol("html")
.setParams("excludeNonText", "true")
```

Reader2Table: Tabular Data Extraction

Extracts structured tables from various document formats with high fidelity

Unified API with reader-specific configuration for different source formats

Streamlines tabular data workflows in Spark NLP distributed pipelines

HTML | DOC/DOCX | XLS/XLSX | PPT/PPTX | Markdown | CSV

Spark NLP v6.1.1 Release

Simplified Integration

Access table data directly in your Spark NLP pipeline

Example Usage

```
val reader = new Reader2Table()
.setContentType("text/html")
.setContentPath("my/html/files")
.setOutputCol("table_data")
```

Reader2Image: Multimodal Image Extraction

Extracts and structures embedded images from a variety of document formats

Enables multimodal workflows combining text and image analysis in the same pipeline

Unlocks new capabilities for vision-language modeling (VLM), multimodal search, and document understanding

PDF | DOC/DOCX | XLS/XLSX | PPT/PPTX

Email (.eml, .msg) | HTML | Markdown

Output Fields

File name | Height | Width | Channels | Mode

Binary data | Metadata

Vision-Language Model Integration

Seamlessly connect to:

- Multimodal LLMs
- Image embeddings
- Visual question answering
- Document AI pipelines

Sample Usage

```
reader = Reader2Image()
.setContentType("text/html")
.setContentPath("my/html/files")
.setOutputCol("images")
```

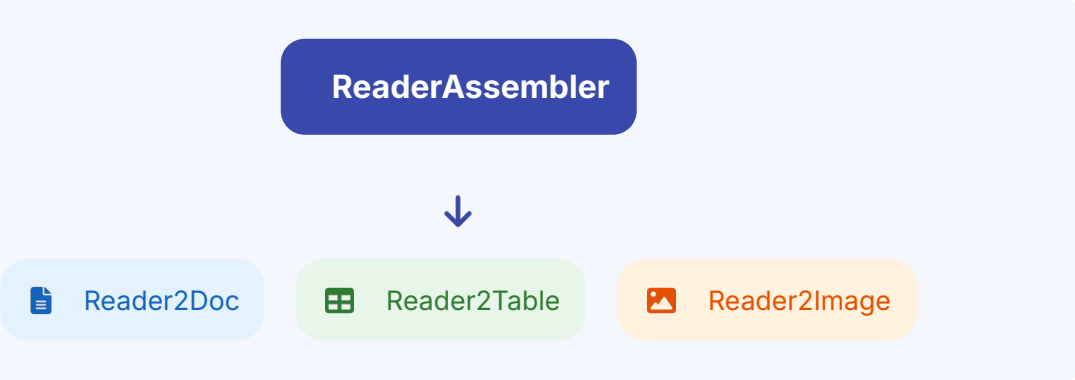
ReaderAssembler: Unified Multiformat Ingestion

Meta-annotator that orchestrates all Reader2X components in a single unified interface

Intelligently selects appropriate readers based on file types and configuration

Provides a declarative approach to complex ingestion pipeline assembly

Introduced in Spark NLP v6.1.5 to enhance pipeline flexibility and reliability



Advanced Capabilities

Automatic Reader Selection

Chooses optimal readers based on file format

Error Handling & Fallbacks

Graceful recovery and robust pipeline operation

Configuration Example

```
ReaderAssembler()
.setReaders(["doc", "table", "image"])
.setContentType("text/html")
.setContentPath("my/html/files")
```

Flexible, High-Performance Pipelines

Direct support for string input columns in readers, no need to write temporary files

Zero I/O overhead when data is already available as strings (generated, preprocessed)

Enhanced maintainability and scaling for streaming or in-memory pipelines

Traditional Workflow

DataFrame → Write to disk → Read files → Process

❌ Extra I/O operations
❌ Storage overhead
❌ Additional failure points

String Input Support

DataFrame → Direct processing

✅ No intermediate storage
✅ Reduced latency
✅ Simplified pipeline

Performance Benefits

String Column Input

Feed raw text from DataFrame directly to readers

```
val reader = new Reader2Doc()
.setInputCol("text_column")
.setOutputCol("document")
```

Streaming Compatibility

Ideal for real-time streaming applications

Stream → Reader → Process

Summary & Key Benefits

Spark NLP Reader annotators transform your AI workflows with powerful document processing capabilities:

- **Fast, fault-tolerant ingestion for text, tables, and images**
- **One unified API for all document formats**
- **Seamless multimodal content processing in Spark**
- **Powers state-of-the-art LLM and VLM pipelines**

Get started with Reader annotators today!

Accelerate AI Workflows With:

Reader2Doc
Unified document processing

Reader2Table
Structured data extraction

Reader2Image
Multimodal image processing

ReaderAssembler
Unified pipeline orchestration

Unstructured.io vs Spark NLP

Side-by-Side Mapping

Unstructured.io
Element-aware document processing

↔

Spark NLP
Annotator-driven NLP pipelines

A technical comparison of document processing approaches for RAG/ETL pipelines

Spark NLP vs Unstructured.io

While both platforms work with unstructured data, they focus on different areas and can complement each other in a complete AI/ML pipeline.

Feature	Spark NLP	Unstructured.io
Primary Purpose	End-to-end NLP workflows for text analysis, understanding, and generation at scale	Document ETL platform focused on extracting and structuring documents for downstream use
Architecture	Built on Apache Spark for distributed processing and scaling	Python-based ETL framework with cloud-based platform options
Document Processing	Growing capability in Spark NLP 6.0+ for PDF, Excel, PowerPoint processing	Core strength: 64+ file formats with advanced document extraction capabilities
NLP Capabilities	Full NLP suite: preprocessing, embeddings, NER, classification, QA, summarization, LLMs	Limited NLP: Primarily extraction, chunking, and basic document analysis
Model Library	80,000+ pretrained NLP models and 55,000+ pipelines for multiple languages and tasks	Focused models for document extraction, table parsing, and OCR
Licensing	100% Apache 2.0 open source with no commercial limitations	Freemium model: Open source core with commercial tiers for advanced features
Scalability	Distributed processing on Spark clusters, native parallel processing	Platform-based scaling with SaaS and In-VPC deployment options
Deployment	CPU/GPU support, Databricks, AWS EMR, cloud-agnostic, embedded	SaaS, In-VPC, and open source self-deployed options

Complementary Use: Use Unstructured.io for initial document processing and extraction, then feed the structured output to Spark NLP for advanced NLP tasks like entity recognition, classification, and LLM integration.

Spark NLP vs Unstructured.io: Feature Comparison

Side-by-side mapping with Spark NLP

Task	Unstructured.io Approach	Spark NLP Equivalent	Spark NLP Advantage
Document Ingestion	**partition**() family Returns typed Elements (Title, NarrativeText, ListItem, Table) Strategies: auto/hi_res/ocr_only	**Reader2Doc/Table/Image, Reader Assembler** Focus on reading into DOCUMENT/IMAGE for pipeline use besides metadata (Title, NarrativeText, ListItem, Table)	Scalable distributed processing Seamless pipeline integration
Cleaning	unstructured.cleaners.* Function-per-task approach Apply per-element or bulk	DocumentNormalizer (doc-level) Normalizer (token-level) Configured within ML pipelines	Declarative configuration Multi-level processing
Chunking	unstructured.chunking.* Element-aware (preserves sections) By title, by page, similarity methods	DocumentCharacterTextSplitter DocumentTokenSplitter InternalDocumentSplitter (licensed)	Optimized for ML pipeline throughput

Key Advantages of Spark NLP Over Unstructured.io

Enterprise Scalability

Spark NLP leverages Apache Spark's distributed computing capabilities for processing massive document collections with linear scalability across clusters.

Pipeline Integration

Seamless integration with ML pipelines through annotator-based design allows for complex workflows with preprocessing, embedding, and model stages.

Rich Model Ecosystem

Pre-trained models for NER, sentiment, embeddings, and more can be directly chained after document processing without leaving the pipeline.

Performance Optimization

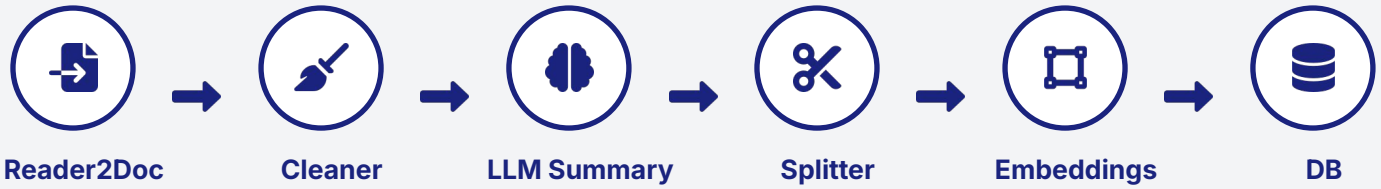
Built for production workloads with memory optimization and throughput-focused design patterns for high-volume document processing.

Demo Notebooks Overview

Explore practical, production-ready RAG ETL pipelines for knowledge retrieval, summarization, and healthcare automation in Spark NLP.

- Complete end-to-end ETL pipelines using Reader2Doc, Reader2Image, Reader2Table annotators for multimodal content ingestion
- Integration of LLMs for document enrichment and VLMs for image understanding within unified pipelines
- Specialized healthcare examples for clinical data extraction, medical reporting, and patient record analysis
- Designed for batch processing or real-time workflows, fully scalable on Spark clusters

RAG-Boost: LLM-Enriched Summaries ETL



Pipeline Description

First, ingest documents with Reader2Doc (PDF, Word, HTML, email formats)

Optional cleaning step to normalize text (minimal preprocessing)

Use LLM to generate abstractive summaries and keywords for each document or section

Split the enriched content into manageable chunks for retrieval

Create and store vector embeddings with enhanced semantic context

Use Cases

Executive summaries of lengthy reports and documents

Faster retrieval over long or verbose documents

Compliance and policy playbooks where concise abstraction improves precision

Knowledge distillation from technical documentation

Semantic search with enhanced context awareness

RAG-Vision: Image → Caption/Summary → Embeddings ETL

Pipeline Flow

Reader2Image → VLM Captioning → Splitter → Embeddings → Write to DB

Use Cases

Extract meaning from slide decks with charts and diagrams

Process scanned forms and documents with visual elements

Analyze infographics and data visualizations

Extract information from screenshots of EHR/portal interfaces

Make visual content searchable in multimodal knowledge bases

Chart | Medical Form | EHR Screen

Implementation Notes

Store both image-derived summary and lightweight OCR (if available) as parallel fields

Tag metadata with comprehensive identifiers: has_image=true figure_id slide_no page_no

Use vision-language models (VLMs) to generate contextual descriptions that capture semantic meaning

Extract text within charts and diagrams for improved searchability

Combine with document context for richer understanding of visual elements

Hybrid Search with Spark NLP and Reader2Doc

In modern information retrieval, delivering accurate and relevant search results requires more than just matching keywords. Hybrid Search combines both symbolic (sparse) and semantic (dense) retrieval techniques to overcome the limitations of each.

Sparse Retrieval (BM25, TF-IDF)

Excels at exact keyword matching and efficiency, but struggles to understand broader context or meaning behind queries.

Dense Retrieval (Neural Embeddings)

Captures semantic similarity between queries and documents, even with different wording, but may miss exact matches without proper filtering.

Why Hybrid Search?

- Retrieve documents that contain the exact terms (via sparse matching)
- Include documents that are semantically similar (via embeddings)
- Rank and combine results for the best of both worlds

What This Notebook Demonstrates

Learn to perform Hybrid Search using Spark NLP, leveraging its latest tools:

- Reader2Doc: ingests rich content (HTML, PDFs) and structures it into document chunks.
- BertSentenceEmbeddings: generates powerful sentence-level embeddings for semantic search.
- Filtering & transformation: prepares content for both dense and hybrid search scenarios.

What You'll Learn

- Parse structured content using Reader2Doc
- Extract sentence embeddings and metadata (chapters, section IDs)
- Prepare data: linking semantic embeddings with structured context
- Implement a semantic + keyword hybrid search pipeline

This example lays the foundation for building production-grade RAG (retrieval-augmented generation), QA, and enterprise search systems using Spark NLP's scalable infrastructure.