QUANSIGHT

Natural Language Processing: An Introductory Tutorial, Part I

Instructors

Fatma Tarlaci & Pamela Wadhwa

Data Science Fellow

QUANSIGHT

Data Science Fellow

QUANSIGHT

Prerequisites

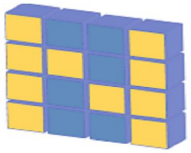
- **Python**
- **Jupyter**

We will also use

Natural Language Toolkit (NLTK)

- Open Source Library for NLP in Python

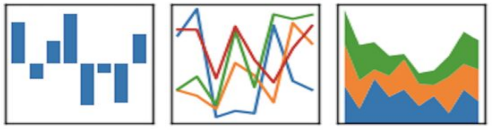
Nice-to-know



NumPy

pandas

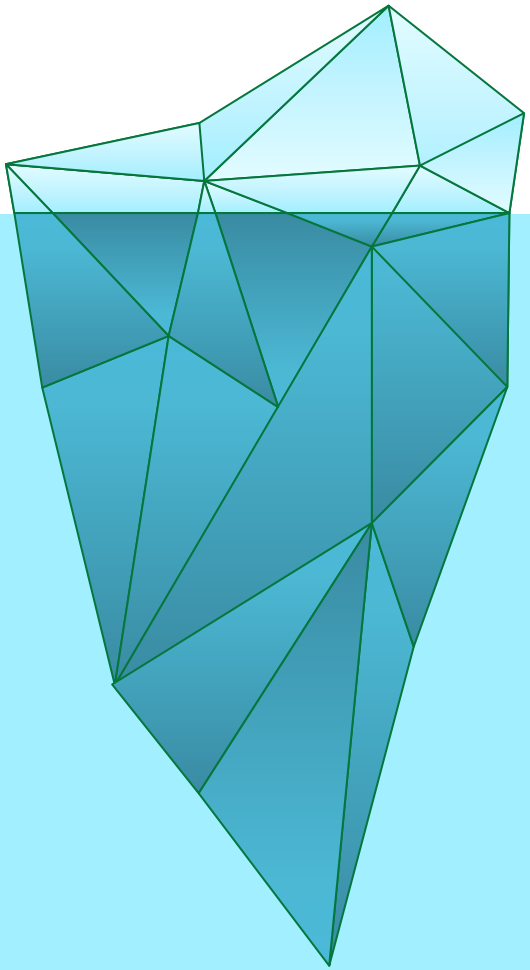
$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



Outline

- Brief overview of NLP concepts: Sentiment Analysis
- The steps for preprocessing the data, using NLTK
- Using scikit-learn to implement a sentiment classifier
- Evaluating the model results

Natural Language Processing (NLP)



Applications

- Machine translation
- Question answering
- Language modeling
- **Sentiment Analysis**
- ...

Sentiment Analysis

Extracting affective states and subjective information from text.

Examples: Customer reviews, tweets, etc.

Sentiment Analysis

Sentiment Analysis is the process of identifying and extracting **opinions** from text (or voice) data.

Polarity analysis is the most common version of sentiment analysis.

Positive

Negative

This is a fairly difficult task.

What about opinions that are not simply *positive* or *negative*?

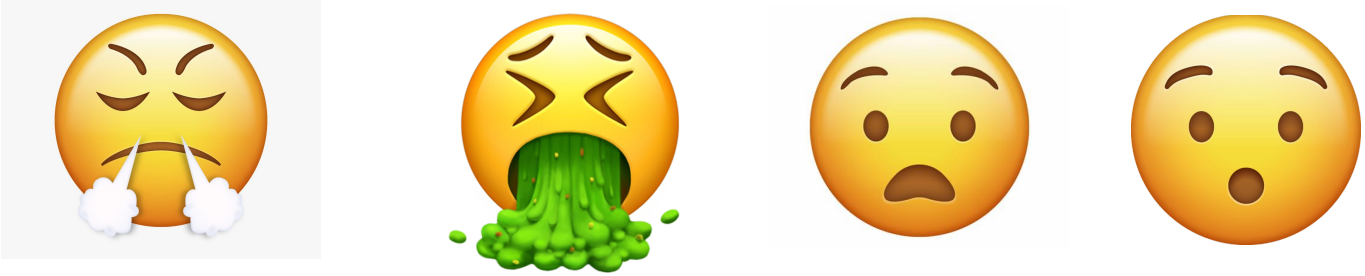
"Like most things in life, wearing #facemask properly ensures best outcomes." 

"If Disney requires masks, I won't go. Period. They will lose a lot of money" 

"my ears are currently carrying sunglasses, headphones, and a face mask. ears are a purse" 🤔

"'I'm not working out with a mask on' is my new favorite excuse for not working out." 🤔

More advanced types of sentiment analysis include **Emotion Detection**



Polarity

Positive? **Negative?**

Is this a classification question?

Classification

Sentiment analysis is modeled as a **classification problem**, where a predefined class label is predicted for a given example of input data.

Classify a given textual input as:

- Positive
- Negative

Part I

Getting Started

01_Project_Twitter.ipynb

QUANSIGHT

Natural Language Processing Tutorial Part II

Recap from Last Week:

- Basic text pre-processing steps
  - Tokenization
  - Stemming
  - Word embedding

- Sentiment analysis - modeling
  - Picking a classifier from scikit-learn
  - Training a classifier from training data
  - Observing results/ evaluation metrics

Today

Sentiment Analysis with Spark NLP

Why do multiple NLP libraries exist?

- NLTK falls short if used in production and/or industry level NLP applications.
- With increasing level of data we need to process, we need more robust tools.
- Spark NLP is one of the industry level NLP libraries that streamlines many of the procedures of creating and deploying NLP applications.

Spark NLP

- Built upon Apache Spark and Spark ML (robust and efficient), takes advantage of TensorFlow behind the scenes
- Widely used in various industries that use NLP applications
- More advanced and most recent NLP research is continuously incorporated in this library

Deep Learning ?

- **Deep Learning**
  - inspired by our understanding of the biology of our brains,
  - artificial neural networks have discrete layers, connections, and directions of data propagation.
- **Pretrained (Language) Models**
  - Large models that have been trained in large datasets and have acquired significant amount of *learning* and *understanding of* the language
- **Transfer Learning**
  - Enables us to use pretrained models' learning without having to train (learn) the language from scratch.

Transfer Learning is briefly

- "...a means to extract knowledge from a source setting and apply it to a different target setting."
  - The general practice is to pretrain representations on a large unlabelled text corpus using your method of choice and then to adapt these representations to a supervised target task using labelled data
    - **Pretraining reduces the need for large amounts of labeled data**

Some of the most powerful Pretrained Language Models include

- BERT
- GPT-2
- XLNET
- RoBERTa

How to use them?

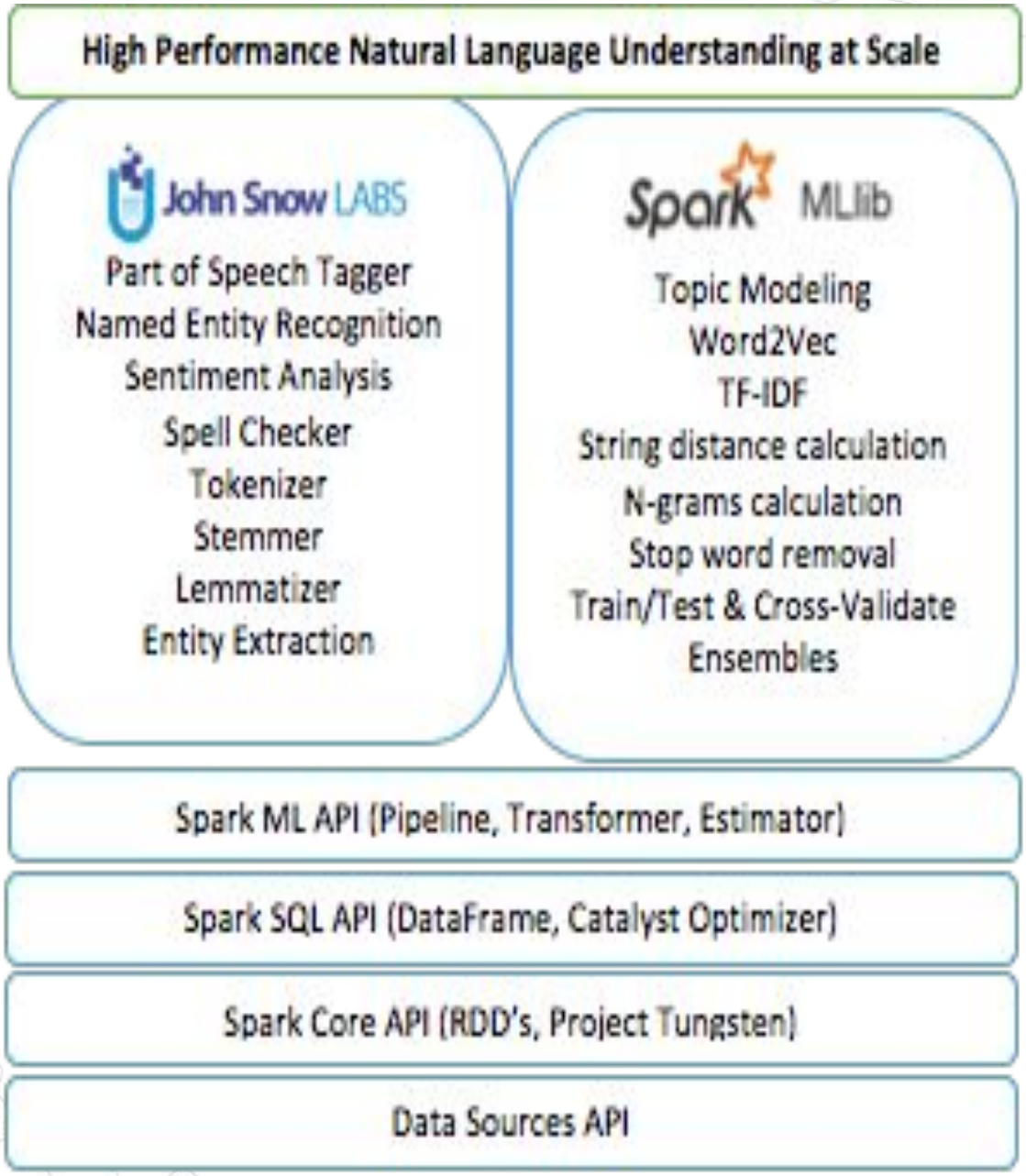
- PyTorch
- Tensorflow (Keras)
- Spark NLP
- ...

Spark NLP

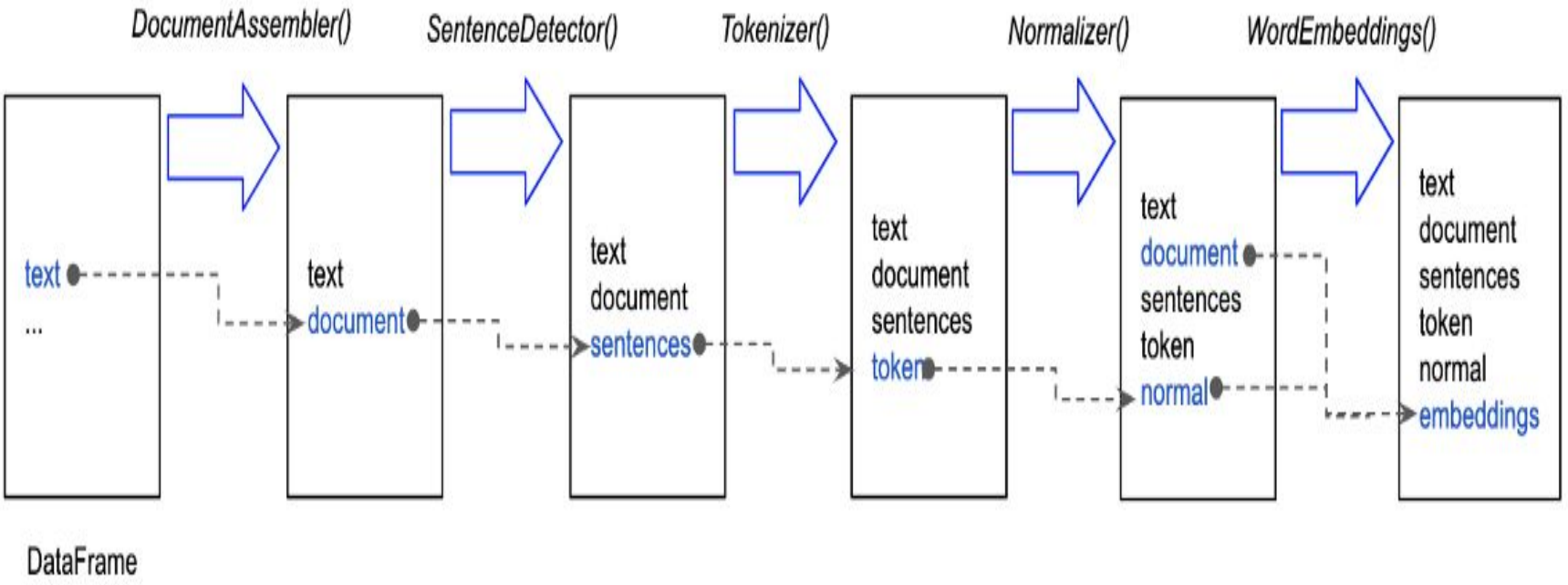
Spark NLP takes advantage of pretrained models and transfer learning by integrating them into its system and making them readily available to its users

Spark NLP Concepts - built upon “Spark ML” concepts

- **Pipeline**
  - a sequence of algorithms to process and learn from data
- **Estimators**
  - have a method called fit() which secures and trains a piece of data
- **Transformer**
  - the result of a fitting process, applies changes to the the target dataset
- **Annotator**
  - An annotator (or annotation) is the basic form of the result of a Spark NLP operation


High Performance Natural Language Understanding at Scale
John Snow LABS
Part of Speech Tagger
Named Entity Recognition
Sentiment Analysis
Spell Checker
Tokenizer
Stemmer
Lemmatizer
Entity Extraction
Spark MLlib
Topic Modeling
Word2Vec
TF-IDF
String distance calculation
N-grams calculation
Stop word removal
Train/Test & Cross-Validate
Ensembles
Spark ML API (Pipeline, Transformer, Estimator)
Spark SQL API (DataFrame, Catalyst Optimizer)
Spark Core API (RDD's, Project Tungsten)
Data Sources API

Pipeline


DocumentAssembler()
SentenceDetector()
Tokenizer()
Normalizer()
WordEmbeddings()
text
...
text
document
text
document
sentences
text
document
sentences
token
text
document
sentences
token
normal
text
document
sentences
token
normal
embeddings
DataFrame

NLP Pipeline Example

```
pipeline = Pipeline(stages=[
 document_assembler,
 sentenceDetector,
 tokenizer,
 normalizer,
 word_embeddings,
 ])
```

Let’s practice with Spark NLP!

Thank you!