ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DIPARTIMENTO DI
SCIENZE STATISTICHE "PAOLO FORTUNATI"

5 - Introduction to Natural Language Processing

Giovanni Della Lunga
giovanni.dellalunga@unibo.it

Halloween Conference in Quantitative Finance

Bologna - October 26-28, 2021

Subsection 1

Introduction

What is NPL (Natural Language Processing)?

Introduction

- Natural language is what people use to communicate with each other. Unlike formal languages (e.g. programming languages), which are defined by strict rules, natural language is flexible, contextual, and evolving.
- As a result, natural language is not as straightforward for a computer program to process as a script written in a language like Java, Python, or SQL.
- When we talk about Natural Language Processing (or **NLP** for short), we are talking about the ways in which we can use computers to process and interact with human language.

Introduction: What is NLP ?

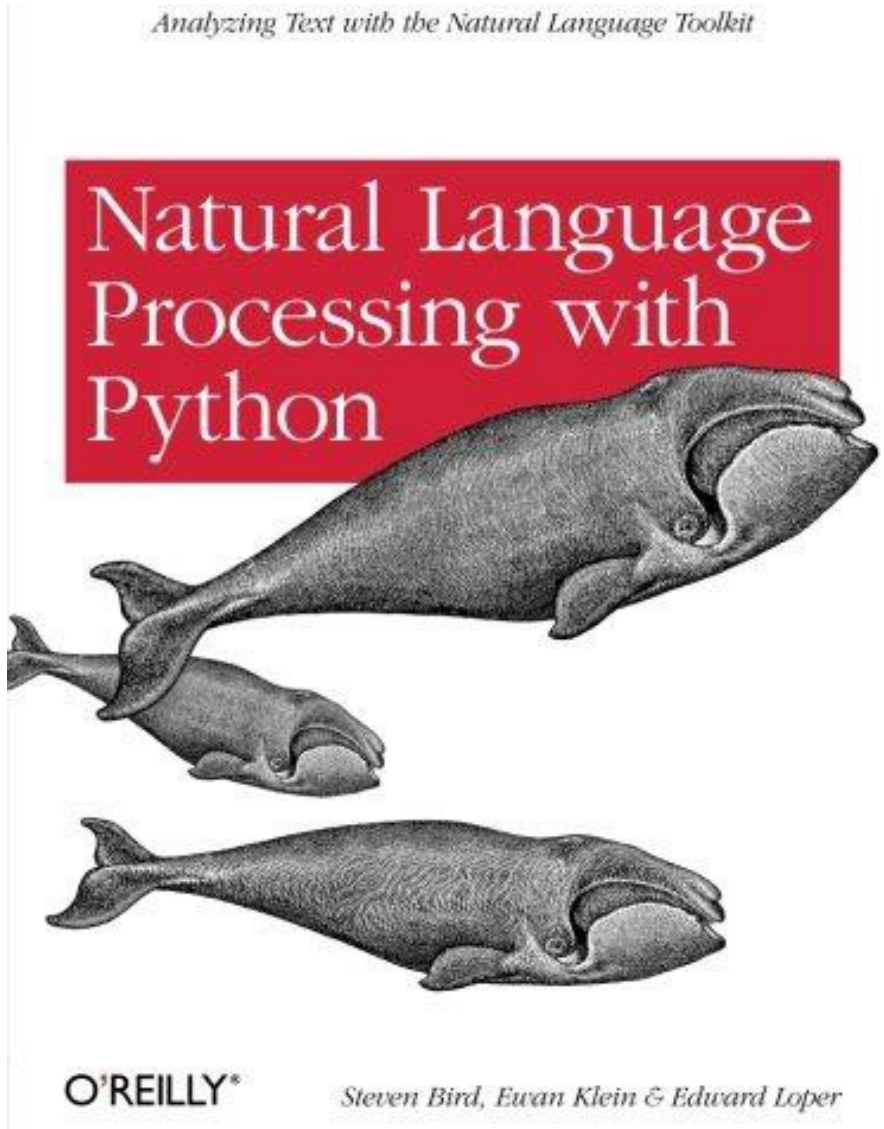
- Having an insight into what people are talking about can be very valuable to financial traders.
- NLP is being used to track news, reports, comments about possible mergers between companies, everything can then be incorporated into a trading algorithm;
- Banks can determine what customers are saying about a service or product by identifying and extracting information in sources like social media.
- This sentiment analysis can provide a lot of information about customers choices and their decision drivers.

Subsection 2

The NLTK Package

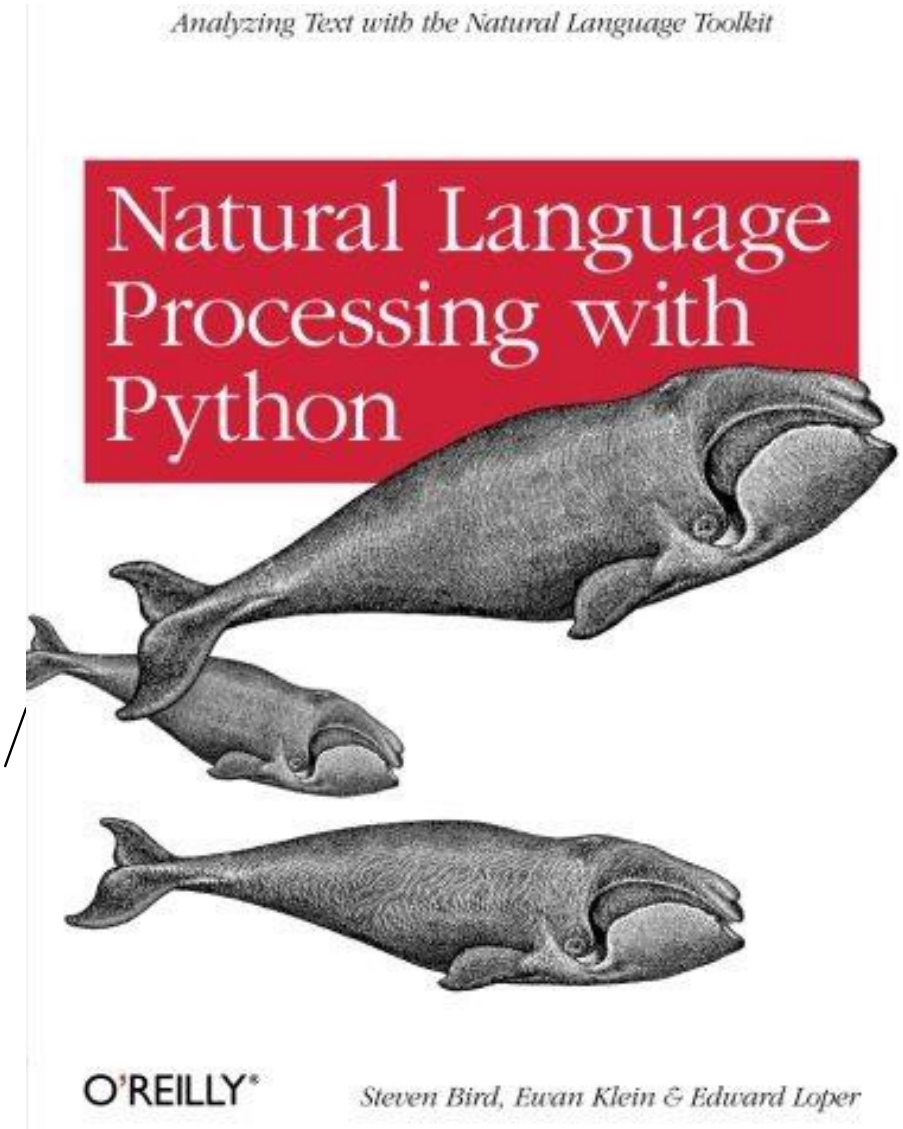
NLTK Package

- The [Natural Language Toolkit](https://www.nltk.org/), or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language.
- It was developed by Steven Bird and Edward Loper in the Department of Computer and Information Science at the University of Pennsylvania.



NLTK Package

- NLTK includes graphical demonstrations and sample data.
- It is accompanied by a book that explains the underlying concepts behind the language processing tasks supported by the toolkit, plus a text book available also on line [here](https://www.nltk.org/book/



NLTK Package

- The latest version is NLTK 3.3. It can be used by students, researchers, and industrialists. It is an Open Source and free library. It is available for Windows, Mac OS, and Linux.
- You can install nltk using *pip installer* if it is not installed in your Python installation. To test the installation:
- Open your Python IDE or the CLI interface (whichever you use normally)
- Type **import nltk** and press enter if no message of missing nltk is shown then nltk is installed on your computer.

NLTK Package

- A **CorpusReader** is a programmatic interface to read, seek, stream, and filter documents, and furthermore to expose data wrangling techniques like encoding and preprocessing for code that requires access to data within a corpus.
- A **CorpusReader** is instantiated by passing a root path to the directory that contains the corpus files, a signature for discovering document names, as well as a file encoding (by default, UTF-8).
- NLTK comes with a variety of corpus readers (66 at the time of this writing) that are specifically designed to access the text corpora and lexical resources that can be downloaded with NLTK.

Corpus Reader

PlaintextCorpusReader. *A reader for corpora that consist of plain-text documents, where paragraphs are assumed to be split using blank lines.*

```
import nltk
from nltk.corpus import PlaintextCorpusReader

corpus_root = '.\corpus\EBA'
corpus_list = PlaintextCorpusReader(corpus_root, '.*', encoding='latin-1')
corpus_list.fileids()
```

```
['Final Guidelines on Accounting for Expected Credit Losses (EBA-GL-2017-06).txt',
 'Final Guidelines on the management of interest rate risk arising from non-trading activities.txt',
 'Final Report on Guidelines on LGD estimates under downturn conditions.txt',
 'Final Report on Guidelines on default definition (EBA-GL-2016-07).txt',
 'Final Report on Guidelines on uniform disclosure of IFRS9 transitional arrangements (EBA-GL-2018-01).txt',
 'Final report on updated GL Funding Plans (EBA 9.12.2019).txt',
 'Final report on updated GL Funding Plans_(EBA-GL-2019-05)_09122019.txt']
```

Corpus Reader

- **TaggedCorpusReader**. *A reader for simple part-of-speech tagged corpora, where sentences are on their own line and tokens are delimited with their tag.*
- **BracketParseCorpusReader**. *A reader for corpora that consist of parenthesis-delineated parse trees.*
- **ChunkedCorpusReader**. *A reader for chunked (and optionally tagged) corpora formatted with parentheses.*
- **TwitterCorpusReader**. *A reader for corpora that consist of tweets that have been serialized into line-delimited JSON.*
- **WordListCorpusReader**. *List of words, one per line. Blank lines are ignored.*
- **XMLCorpusReader**. *A reader for corpora whose documents are XML files.*
- **CategorizedCorpusReader**. *A corpus readers whose documents are organized by category.*

Subsection 3

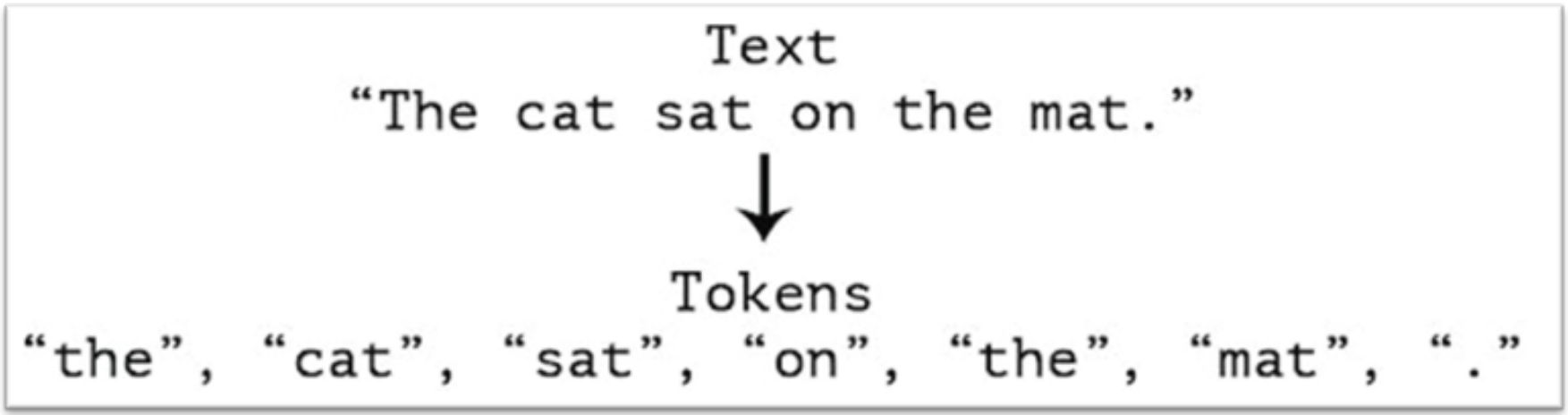
Text Processing

Text Processing

- Removing Punctuation
- Converting to Lower Case
- Removing Numbers (?)

Preprocessing

- Tokenization Is the process of segmenting text into sentences and words.
- In essence, it is the task of cutting a text into pieces called tokens, and at the same time throwing away certain characters, such as punctuation.



Tokenization

- In this process some very common words that appear to provide little or no value to the NLP objective are filtered and excluded from the text to be processed, hence removing widespread and frequent terms that are not informative about the corresponding text.
- Ex.: common language articles, pronouns and prepositions such as **and**, **the** or **to** in English.
- In many situations, stop words can be safely ignored by carrying out a lookup in a pre-defined list of keywords, freeing up database space and improving processing time.
- But **this is not always the case**.

Stop Words Removal

Be aware that:

- There is no universal list of stop words!
- stop words removal can wipe out relevant information and modify the context in a given sentence.
- For example, if we are performing a **sentiment analysis** we might throw our algorithm off track if we remove a stop word like **not**.
- Under these conditions, you might select a minimal stop word list and add additional terms **depending on your specific objective**.

Stemming and Lemmatization

- Stemming refers to the process of slicing the end or the beginning of words with the intention of removing affixes (lexical additions to the root of the word).
- Affixes that are attached at the beginning of the word are called **prefixes** (e.g. **astro** in the word **astrobiology**) and the ones attached at the end of the word are called **suffixes** (e.g. **ful** in the word **helpful**).

Stemming and Lemmatization

Typically a large corpus will contain many words that have a common root – for example: offer, offered and offering. Lemmatisation and stemming both refer to a process of reducing a word to its root. The difference is that stem might not be an actual word whereas, a lemma is an actual word. It's a handy tool if you want to avoid treating different forms of the same word as different words. Let's consider the following example:

- **Stemming**: considered, considering, consider $\rightarrow$ "consid"
- **Lemmatising**: considered, considering, consider $\rightarrow$ "consider"

Stemming and Lemmatization

- Has the objective of reducing a word to its base form and grouping together different forms of the same word.
- For example, verbs in past tense are changed into present (e.g. **went** is changed to **go**) and synonyms are unified (e.g. **best** is changed to **good**), hence standardizing words with similar meaning to their root.
- Although it seems closely related to the stemming process, lemmatization uses a different approach to reach the root forms of words.

Stemming and Lemmatization

- Lemmatization resolves words to their dictionary form (known as lemma) for which it requires detailed dictionaries in which the algorithm can look into and link words to their corresponding lemmas.
- For example, the words **running** , **runs** and **ran** are all forms of the word **run** , so **run** is the lemma of all the previous words.

Stemming and Lemmatization

- lemmatization is a much more resource-intensive task than performing a stemming process.
- At the same time, since it requires more knowledge about the language structure than a stemming approach, it **demands more computational power** than setting up or adapting a stemming algorithm.

Example

- Using **05-natural-language-processing** Notebook
- Par. 5.1-5.4

Subsection 4

Part-of-Speech (POS) Tagging

Part-of-Speech (POS) Tagging

- The process of classifying words into their parts of speech and labeling them accordingly is known as **part-of-speech tagging** or **POS-tagging**, or simply **tagging**.
- The part of speech explains how a word is used in a sentence

```
sentence = "My name is Giovanni"
token = nltk.word_tokenize(sentence)
token
```

```
['My', 'name', 'is', 'Giovanni']
```

```
nltk.pos_tag(token)
```

```
[('My', 'PRP$'), ('name', 'NN'), ('is', 'VBZ'), ('Giovanni', 'NNP')]
```

POS Tagging

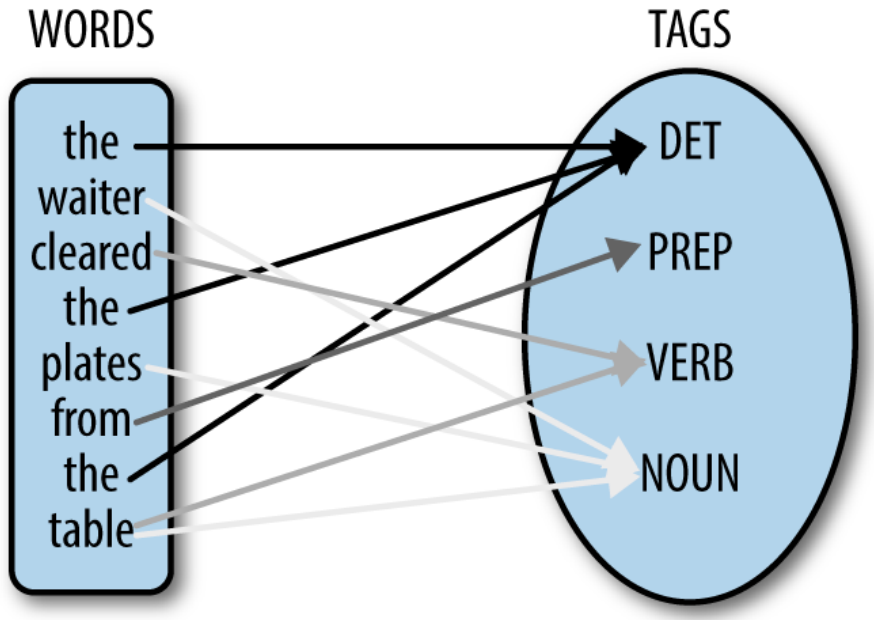
- There are eight main parts of speech:
- nouns,
- pronouns,
- adjectives,
- verbs,
- adverbs,
- prepositions,
- conjunctions and
- interjections.

POS Tagging

- **Noun (N)** : Daniel, London, table, dog, teacher, pen, city
- **Verb (V)** : go, speak, run, eat, play, live, walk, have, like, are, is
- **Adjective (ADJ)** : big, happy, green, young, fun, crazy, three
- **Adverb (ADV)** : slowly, quietly, very, always, never, too, well, tomorrow
- **Preposition (P)** : at, on, in, from, with, near, between, about, under
- **Conjunction (CON)** : and, or, but, because, so, yet, unless, since, if
- **Pronoun (PRO)** : I, you, we, they, he, she, it, me, us, them, him, her, this
- **Interjection (INT)** : Ouch! Wow! Great! Help! Oh! Hey! Hi!

POS Tagging

- The collection of tags used for a particular task is known as a Tagset.
- A part-of-speech tagger, or POS-tagger, processes a sequence of words, and attaches a part of speech tag to each word.



POS Tagging

AT	article
BEZ	the word *is*
IN	preposition
JJ	adjective
JJR	comparative adjective
MD	modal (*may, can, ...*)
MN	singular or mass noun
NNP	singular proper noun
NNS	plural noun
PERIOD	. : ? !
PN	personal pronoun
RB	adverb
RBR	comparative adverb
TO	the word *to*
VB	verb, base form
VBD	verb, past tense
VBG	verb, present participle
VBN	verb, past participle
VBP	verb, non 3d person singular present
VBZ	verb, 3d person singular present
WDT	wh-determiner (*what, which ...*)

Some POS Tags used in English

- **Partial parsing**: syntactic analysis
- **Information Extraction**: tagging and partial parsing help identify useful terms and relationships between them.
- **Question Answering**: analyzing a query to understand what type of entity the user is looking for and how it is related to other noun phrases mentioned in the question.

Example

- Using
  05-natural-language-processing
  Notebook
- Par. 5.5