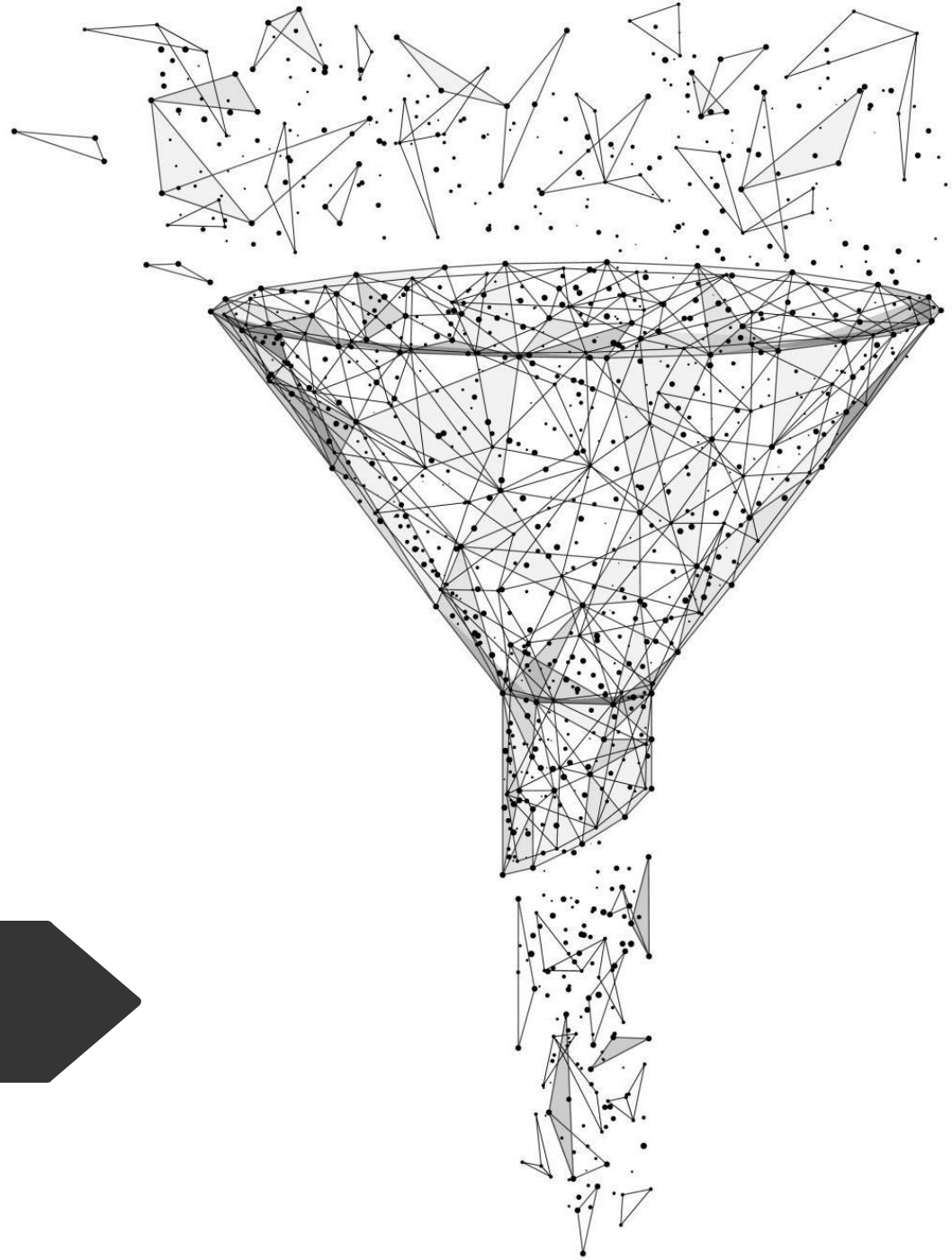
Data Mining and Text Analytics

Lecture_05 "Text Mining_2"

Postgraduate Programme in AI for Business and Society

Prof Alessandro Bruno

Data Annotation and Processing



Outline

- Data Annotation
- Preprocessing Steps
- IMDB case study
- Web Scraping (Crawling)
- Robots.txt
- Tokenization
- Word Form Normalisation
- NER (Named Entity Recognition)
- Syntax and Dependecy Tree

Data Acquisition

Data Acquisition sources → Different Mining Tasks

Data Sources

- Open Domain Data
- Social Media Networks

Data Sources

- Closed Domain Data
- Financial field
- Healthcare systems

Data Acquisition

Acquiring data from by solely relying on a specific domain might not be sufficient

Why is that so?

specific domain implies professional domain knowledge.

Healthcare systems:

- EHR (Electronic Health Records) are full of medica terms

Financial Field

- Very specific terms referring to investments, fundings.

Data Acquisition

Closed Domain

Publicly available data sources are often used to counterbalance the missing information from closed domain scenarios

- Wikipedia
- Baidu Encyclopedia
- Textbooks

It should be noted that Public Networks usually contain much more noisy and ill-formed expression.

Overhead processing: Cleaning, Pre-processing routines are needed!

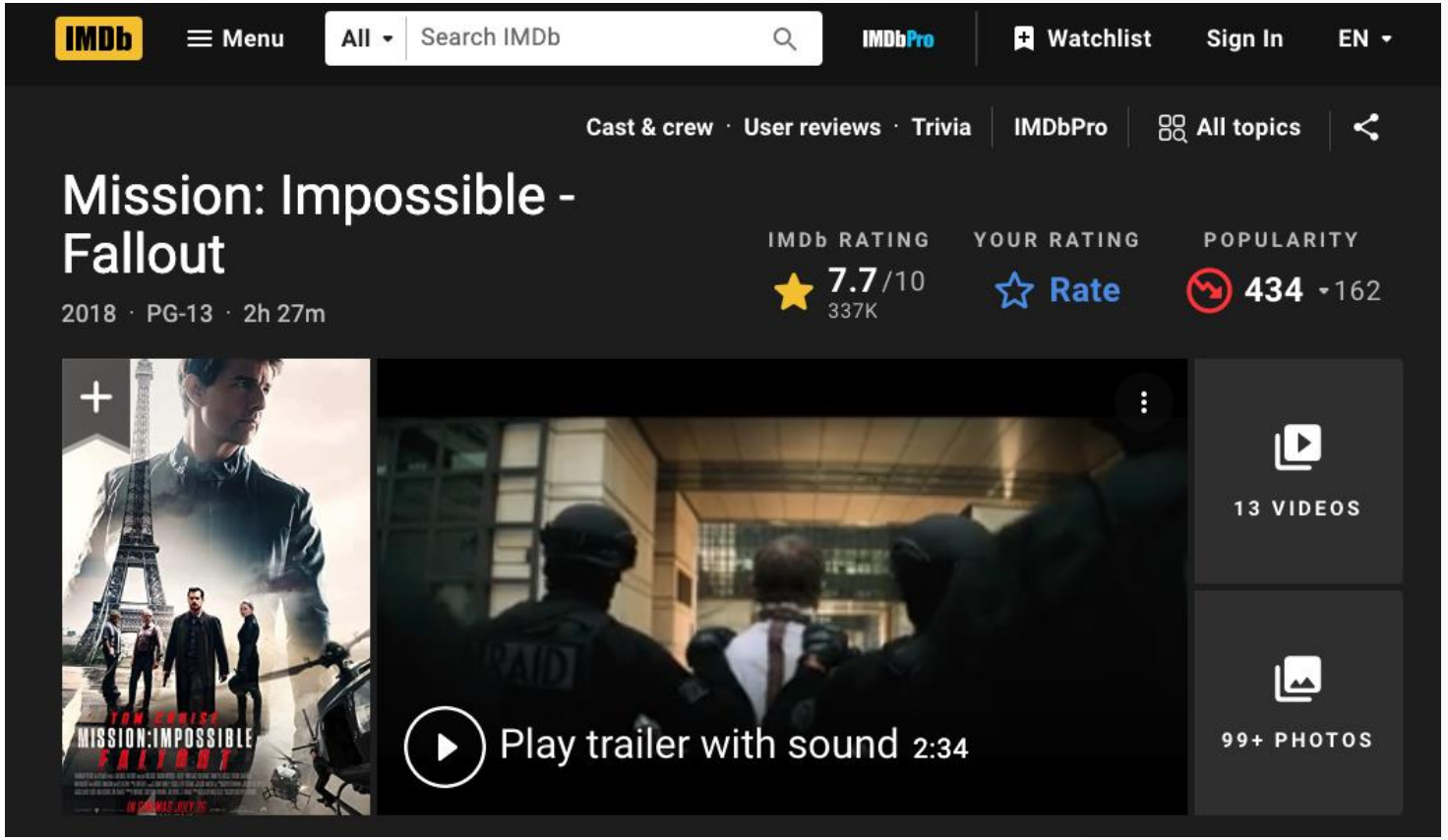
Data Acquisition: the IMDb use case

- IMDb is a pretty popular website that provides users with comments on movies.
- IMDb webpages are full of links to movies

Website link to IMDb: https://www.imdb.com

The IMDb use case

- Users can easily check out rates and comments.
- Lots and lots of comments can be read by clicking on the yellow star.



The IMDb use case

One can easily scroll down the users' reviews all the way down to the bottom of the page.

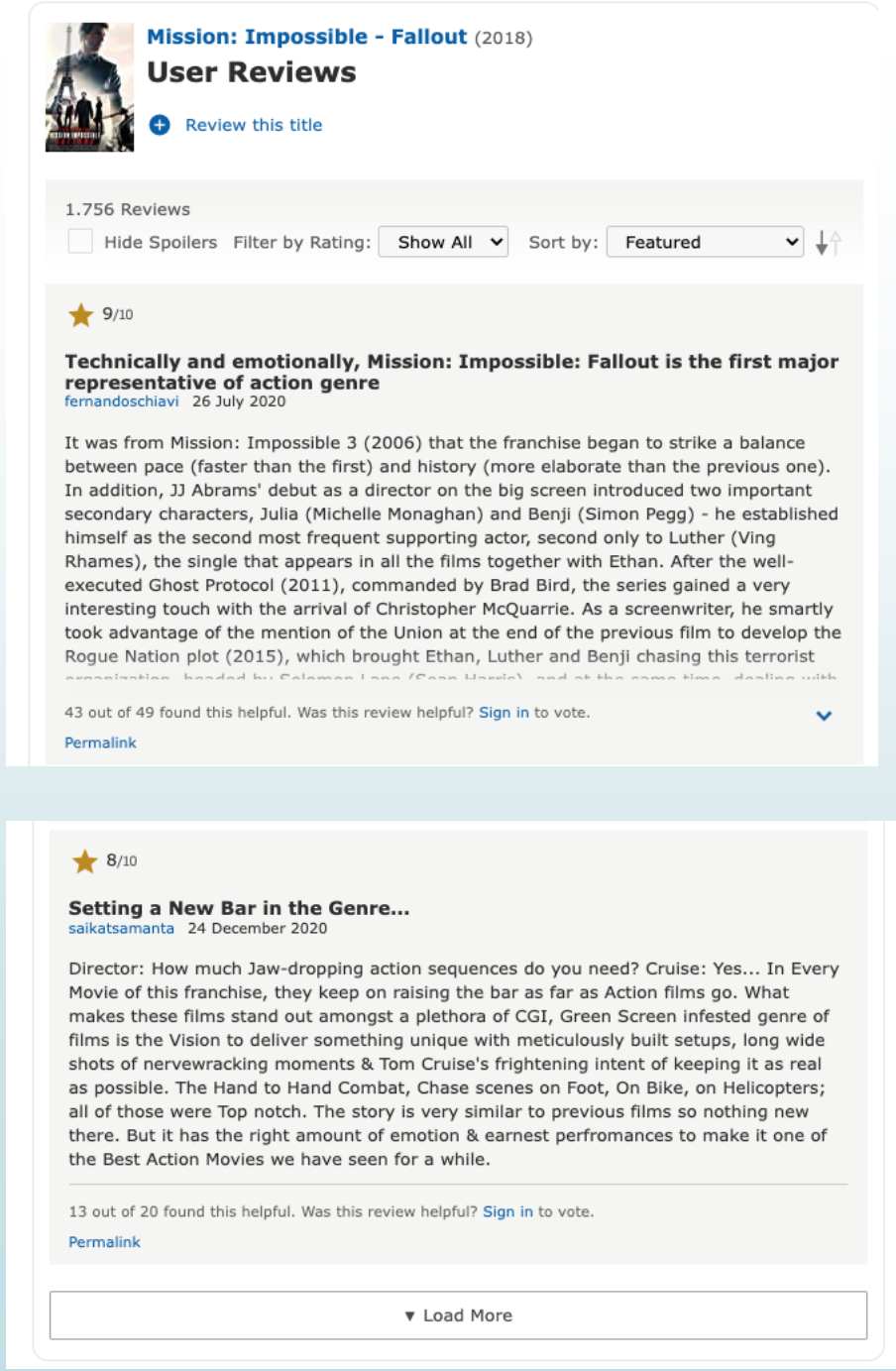
Is there a way to automatically crawl the whole "reviews" contents?

Python programming language library named **urllib2** helps users download the entire section contents

That goes under the name of **Web Crawling (or Web Scraping)**

Top

Bottom



Webpage content crawling

Python Programming Language

Urllib2 library

Checking out the **robots protocol** of the website

Robots.txt is a file containing all the limitations enforced against crawling

https://www.imdb.com/robots.txt → this URL contains the robots protocol with a limitation list

Webpage content extraction

- Since no restriction are in place to download reviews content, the owner allows users/developers to crawl these contents.
- **Webpage** content **crawling** is recommended during low networking activity, usually overnight.
- **Beautiful Soup** is a Python based toolkit to extract content and obtain the link to the next webpage.
- It should also be noted that webpage contents are full of special symbols with no semantic meaning.
- The task of content analysis is named “parsing”
- Special symbols like “ ”, “&lt” represent space and less-than, respectively.

Does robots.txt enforce limitations by the Law?

- Introduced as "Robots Exclusion Protocol"
- The file `robots.txt` does not enforce legal constraints against web scraping.
- Nevertheless, it plays an important role in setting the **expectations and permissions** of a website owner regarding automated access.

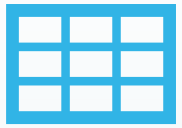
Special Symbols in Web Data

Result	Description	Entity Name
	non-breaking space	
<	less than	&lt;
>	greater than	&gt;
&	ampersand	&amp;
¢	cent	&cent;
£	pound	&pound;
¥	yen	&yen;
€	euro	&euro;
©	copyright	&copy;
®	registered trademark	&reg;

What is the end for Special Symbols?

- **Remove Special Symbols**
  - **Noise Reduction:** Eliminates irrelevant characters that can hinder analysis.
  - **Standardization:** Ensures consistent data format for efficient processing.
  - **Improved Accuracy:** Focuses on essential text elements for accurate insights.
- **Impact on Text Mining Techniques**
  - **Sentiment Analysis:** More precise sentiment detection without distractions.
  - **Topic Modeling:** Clearer identification of underlying themes.
  - **Text Classification:** Enhanced categorization accuracy.

Webpage Content Extraction



Once the reviews content is obtained, it undergoes data cleaning.

Furthermore, any noise word is removed.

Noise Processing

Removal of comments that are too short (meaningless)

Mappings of labels

Zooming in on Content Processing within Data Acquisition

Noise Processing

The IMDb website may contain symbols such as “@”, advertisement links, and so on.

“@” may be followed by a user name.

Are those textual elements and symbols meaningful to Text Mining?

They are not meaningful to it.

Rule-based or template-based approaches are used to remove noisy symbols or textual elements within contents

Removal of too short text

- Depending on the content language, there needs word segmentation in order to have the correct word counting in place.
- For instance, counting the number of words in English content is straightforward as it can be done by counting the number of spaces.
- In Chinese content, you may have some separate characters to be combined to form words.
  - A simple rule-based system sees removal of words shorter than a certain threshold. (i.e. remove all words consisting of less than 3 characters).

Mappings of labels

Websites "hide" labels within their “html” code.

Those labels might be different in numbers and name than the ones considered by a classifier.

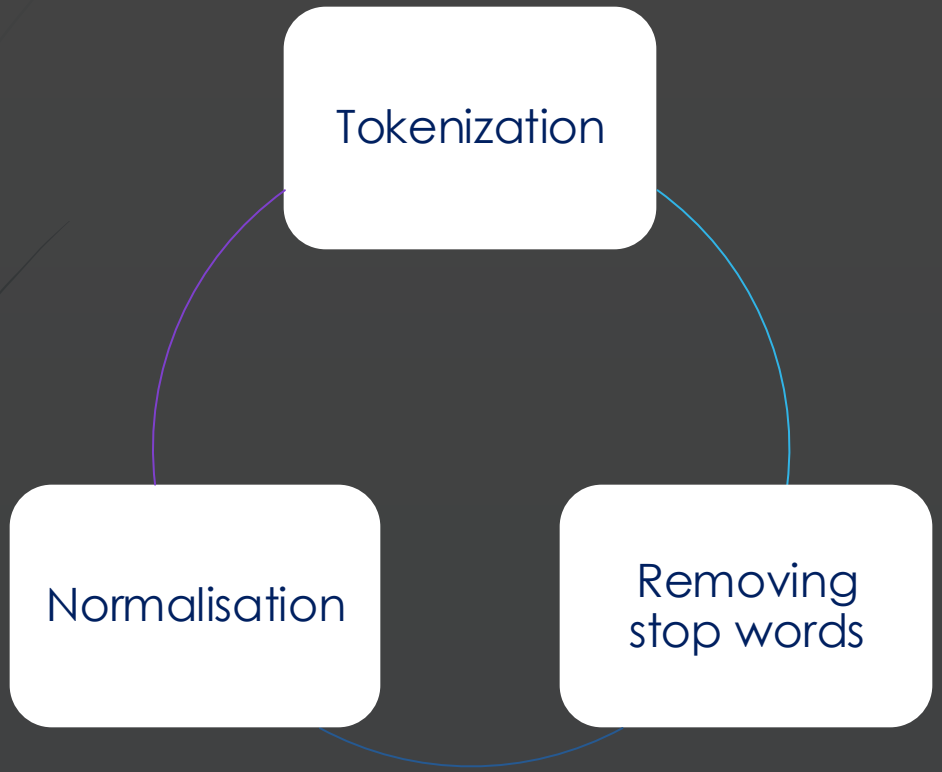
A label mapping steps is needed to sort out any forms of ambiguity between the two groups of labels-categories

Mappings of labels

Some examples of labels and categories

- Download Evaluation score uses a 5-point system
- A sentiment classifier uses only a 2-point system
- A label mapping is necessary to avoid any mismatch
- Generally 1 and 2 for the Download Evaluation score represent negative feedback
- At the same time, 4 and 5 are positive feedback
- If we cast it upon a sentiment scale, 3 is halfway through it (it could be considered as a neutral feedback).
- Solution: 1 and 2 are mapped into Negative Feedback;
- 4 and 5 are mapped into Positive Feedback
- 3 is removed as it represent a neutral evaluation of the download process.

Data Preprocessing



Tokenisation

- What is Tokenisation?
- How does it work?
- Give it a go at the link below:
- https://platform.openai.com/tokenizer
- Type in a paragraph of your choice and check out the generated tokens

Tokenization

- Preprocessing steps come into play Immediately after data acquisition.
- Tokenization refers to a process of segmenting a given text into “lexical” units.
- Latin and inflectional languages (e.g. English) use spaces as word separators.
- Only space and punctuation marks are required to realise lexicalization.
- Other than above, languages with no separation marks (Chinese), and some agglutinative languages (e.g. Japanes, Korean, Vietnamese) go first with word segmentation.

Removal of stop words

Functional words: auxiliary words, prepositions, conjunctions, modal words, high frequency words that often appear in documents carrying only little text information.

"The, is, at, which, on" are examples of functional words.

GOAL: Minimising the storage space for text mining.

Functional Words are discarded during the phase of text representation.

In the implementation phase of a Text Mining module, a list of stop words is established.

All stop words are removed before proceeding to feature extraction

Word Form Normalisation

Word form normalisation consists of two concepts:

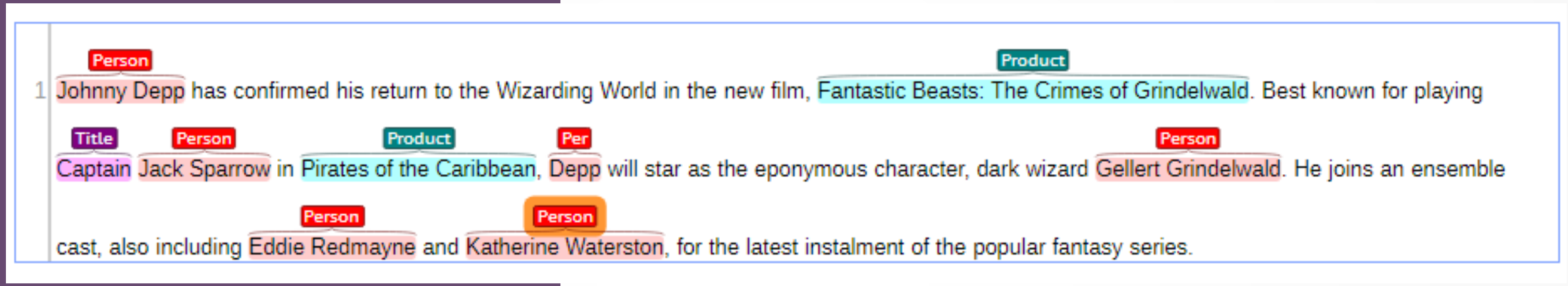
- Lemmatisation
- Stemming

Definitions

- Lemmatisation is the restoration of arbitrarily deformed words into original forms
  (e.g. cats → cat, did → do)
- Stemming is the process of removing all affixes to obtain roots
  (fisher → fish, effective → effect)

Word Form Normalisation

- It is usually realised by rules of regular expressions
- Porter stemming is a a widely employed stemming algorithm consisting of four main steps:
  - 1. Dividing letters into vowels and consonants
  - 2. utilising rules to process words ending with suffixes of –s, -ing, and –ed.
  - 3. designing special rules to address complicated suffixes (e.g. –ational)
  - 4. fine-tuning the processing results by rules.
- It should be noted that several stemming algorithms exist and bring different results (even with the same language).
- On an application oriented note, NLTK toolkit in Python provides calling functions for Porter stemming algorithm.



Data Annotation

Data Annotation

It represents the foundation of Supervised Machine Learning tasks

Data (e.g. statements extracted from the Internet)

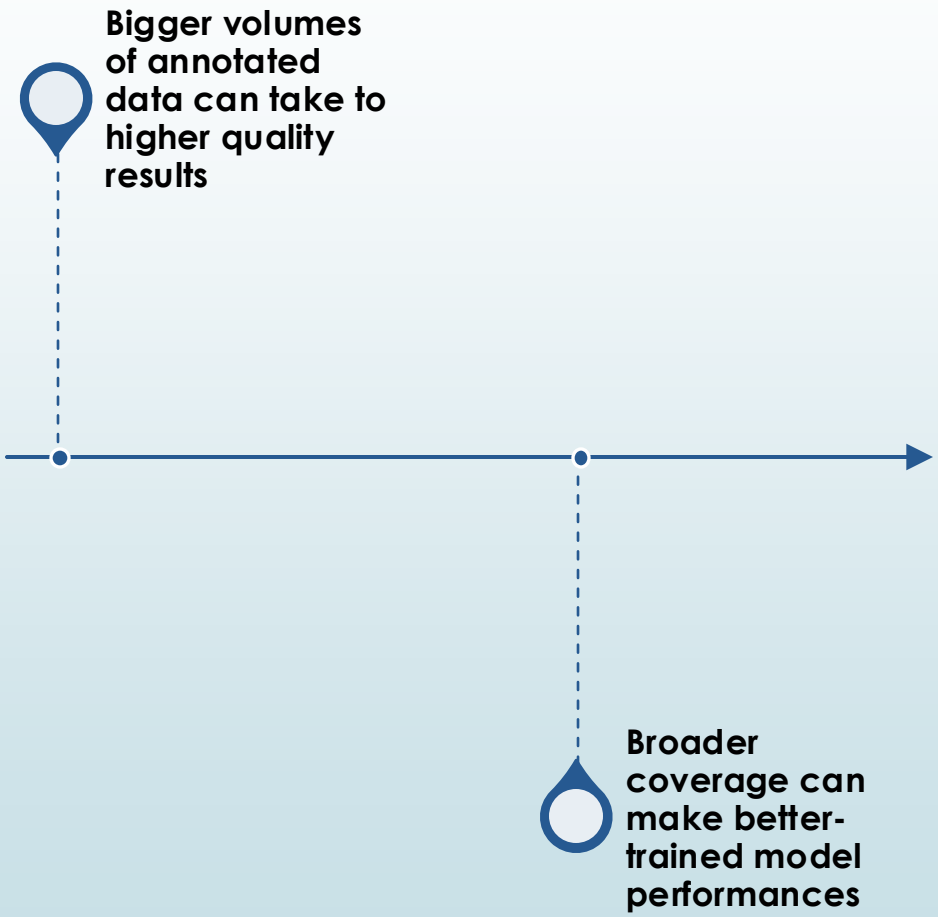
Annotated Data

Statements on Marketing Topics

Statements on topics different than Marketing

Classifier

Data Annotation



Data Annotation, some examples:

- Here is a **textual description** of a patient's postoperative course after heart failure symptoms appear.
- *Mr Shinabery is a 73-year-old gentleman who returned to Surgluthe Leon Calcner Healthcare to the emergency room on 9/9/02 with crescendo spontaneous angina and shortness of breath. He is three-and-one-half months after a presentation with subacute left cyrcumflex thrombosis, ischemic mitral regurgitation, pulmonary edema and a small nontransmural myocardial infarction. Dilation of the left circumflex resulted in extensive dissection but with eventual achievement of a very good angiographic and clinical result after placement of multiple stents, and his course was that of gradual recovery and uneventful return home.*

Textual excerpt Data Annotation

- *Mr Shinabery is a 73-year-old gentleman who returned to* $[\text{Surgluthe Leon Calcner Healthcare}]_{Hosp}$ *to the emergency room on* $[\text{9/9/02}]_{Time}$ *with* $[\text{crescendo spontaneous angina}]_{sym}$ *and* $[\text{shortness of breath}]_{sym}$*. He is* $[\text{three-and-one-half months}]_{dur}$ *after a presentation with* $[\text{subacute left cyrcumflex thrombosis}]_{dis}$*,* $[\text{ischemic mitral regurgitation}]_{dis}$*,* $[\text{pulmonary edema}]_{dis}$ *and a small* $[\text{non-transmural myocardial infarction}]_{dis}$*.* $[\text{Dilation of the left circumflex}]_{Treat}$ *treat resulted in extensive dissection but with eventual achievement of a very good* $[\text{angiographic and clinical result}]_{TR}$ *after* $[\text{placement of multiple stents}]_{Treat}$*, and his course was that of gradual recovery and uneventful return home.*

Hospital, Time, symptoms, duration, disease, Treatment, Treatment Result

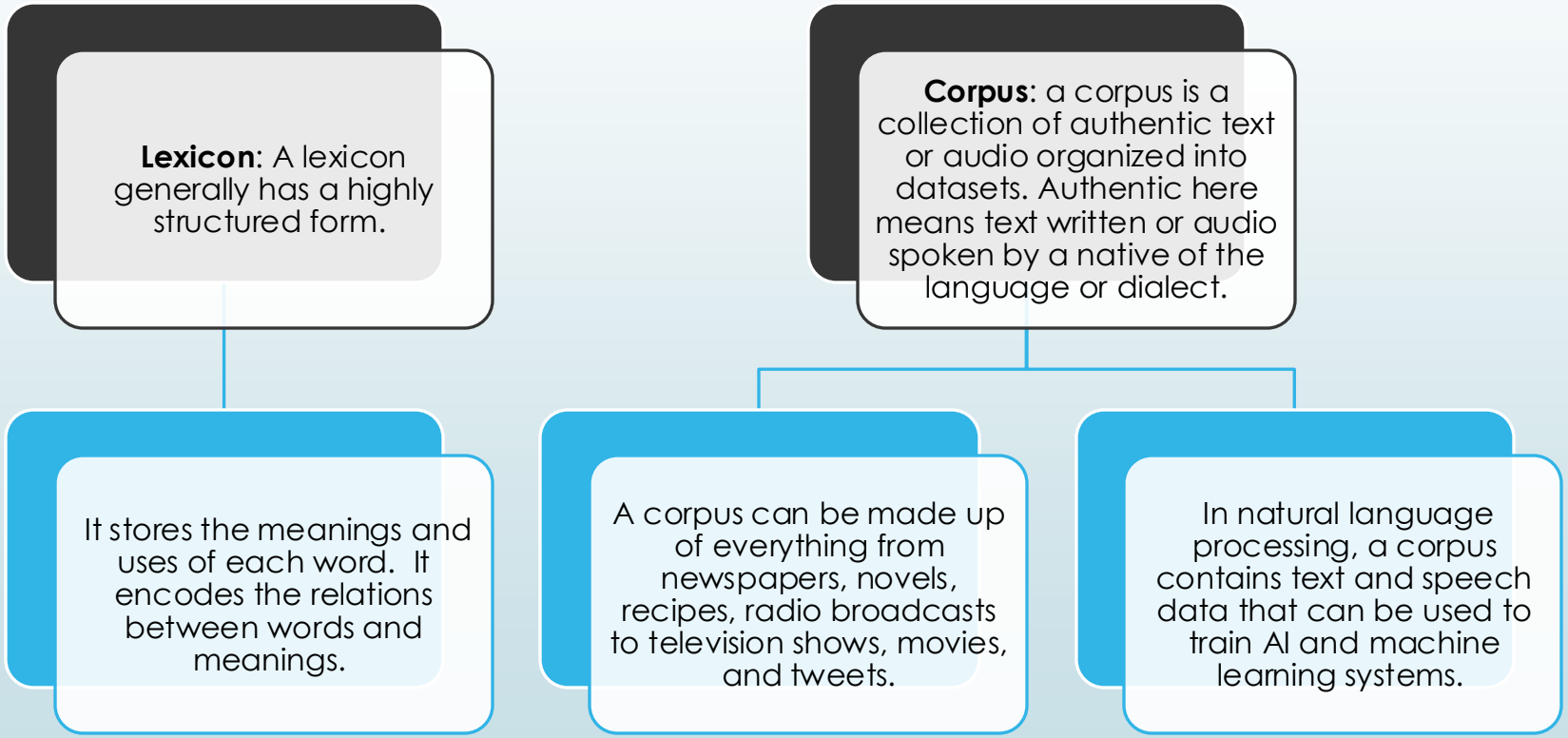
Data Annotation - Challenges

As in the last example, data annotation task is not a straightforward one!

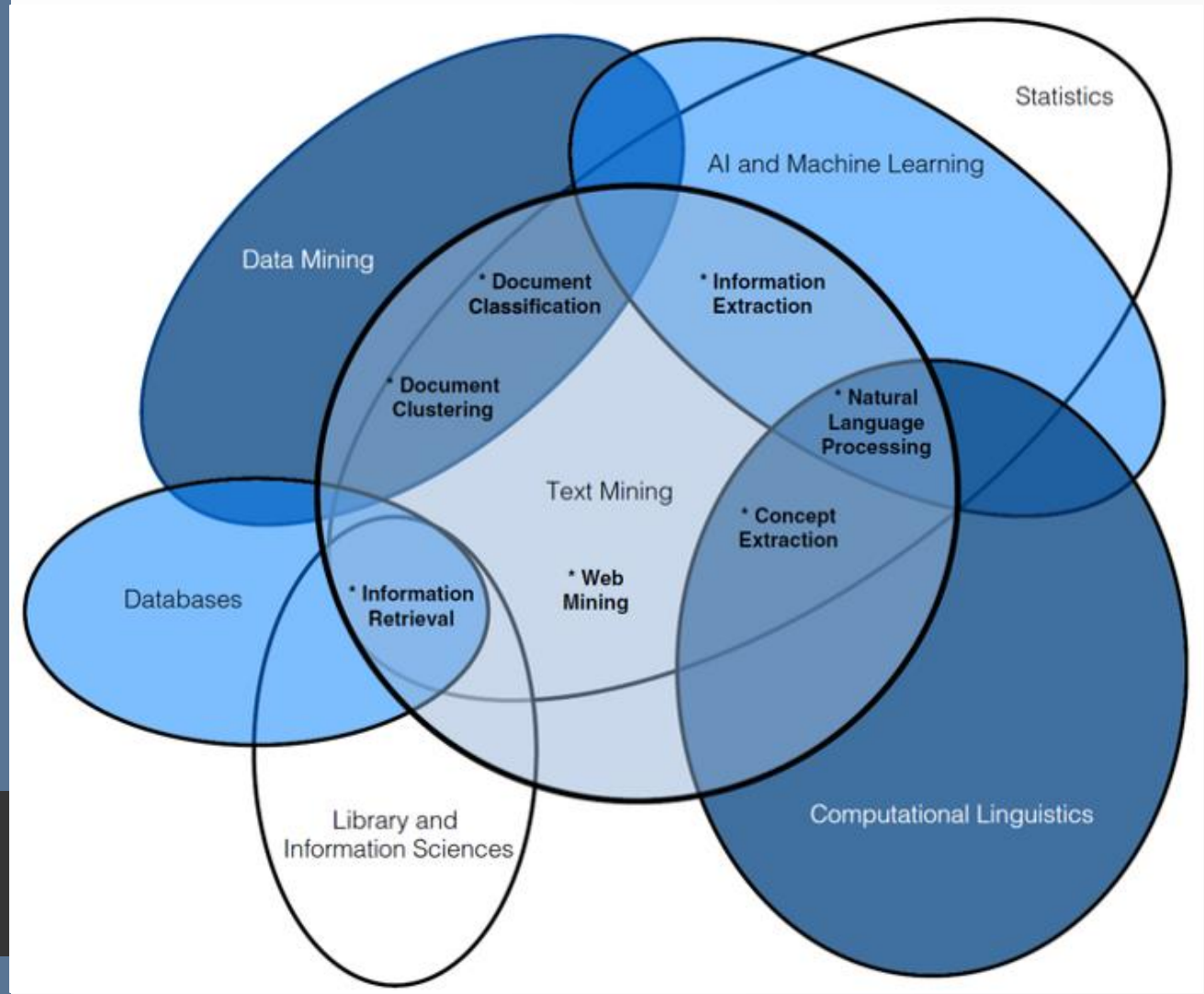
Professional Knowledge can be necessary in order to get the right labels and words.

Data Mining techniques have also multimodal data annotation in place especially when text, videos and images are to be annotated.

Data Annotation (some definitions)



Text Mining and NLP

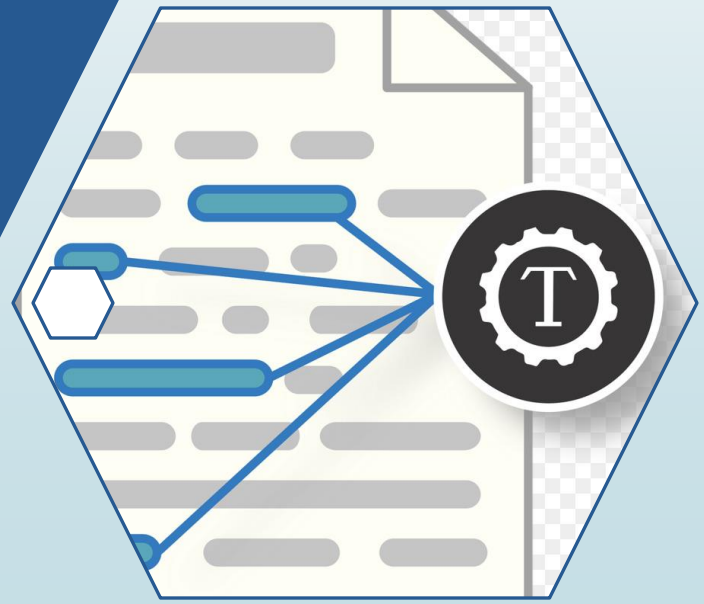


Basic Tools of NLP

Text Mining involves several tasks from NLP (Natural Language Processing), pattern classification, and machine learning.

Here is a list of basic NLP tools:

- Word segmenters;
- Syntatic parsers;
- Part-of-speech taggers;
- Chunkers



NLP tool (1) Tokenisation

It separates text into a sequence of "words" called "tokens".

Examples:

- "That's" → tokenisation → that, 's (two tokens)
- "rule-based" → tokenisation → rule, -, based (3 tokens)

The NLTK toolkit provides a tokenization package.

NLTK toolkit URL: https://www.nltk.org/api/nltk.tokenize.html

NLP tool (1) Tokenisation

- You are asked to run tokenisation on a given paragraph from a marketing website:
- "The Gatherverse community is squarely focused on the metaverse and the community implications these new technologies present to business leaders and organizations. The group leads with wellness, ethics, and safety which is a refreshing change from many other tech events that are more focused on ROI and business outcomes. Very much like the tenets of my Metaverse Manifesto, there is a strong focus on digital citizenship, inclusion, and accessibility of the metaverse for all."
- Copy and paste it onto the form at link below:
  - http://text-processing.com/demo/tokenize/

NLP tools (some considerations)

Think of the following words:

Take, takes, taken, took, taking (different forms of the same word due to grammatical reasons)

Token, tokenize, tokenization

When using statistical methods, words sharing the same stem are to be considered as the same word.

Syntactic parsing refers to mining phrase structure, and dependency

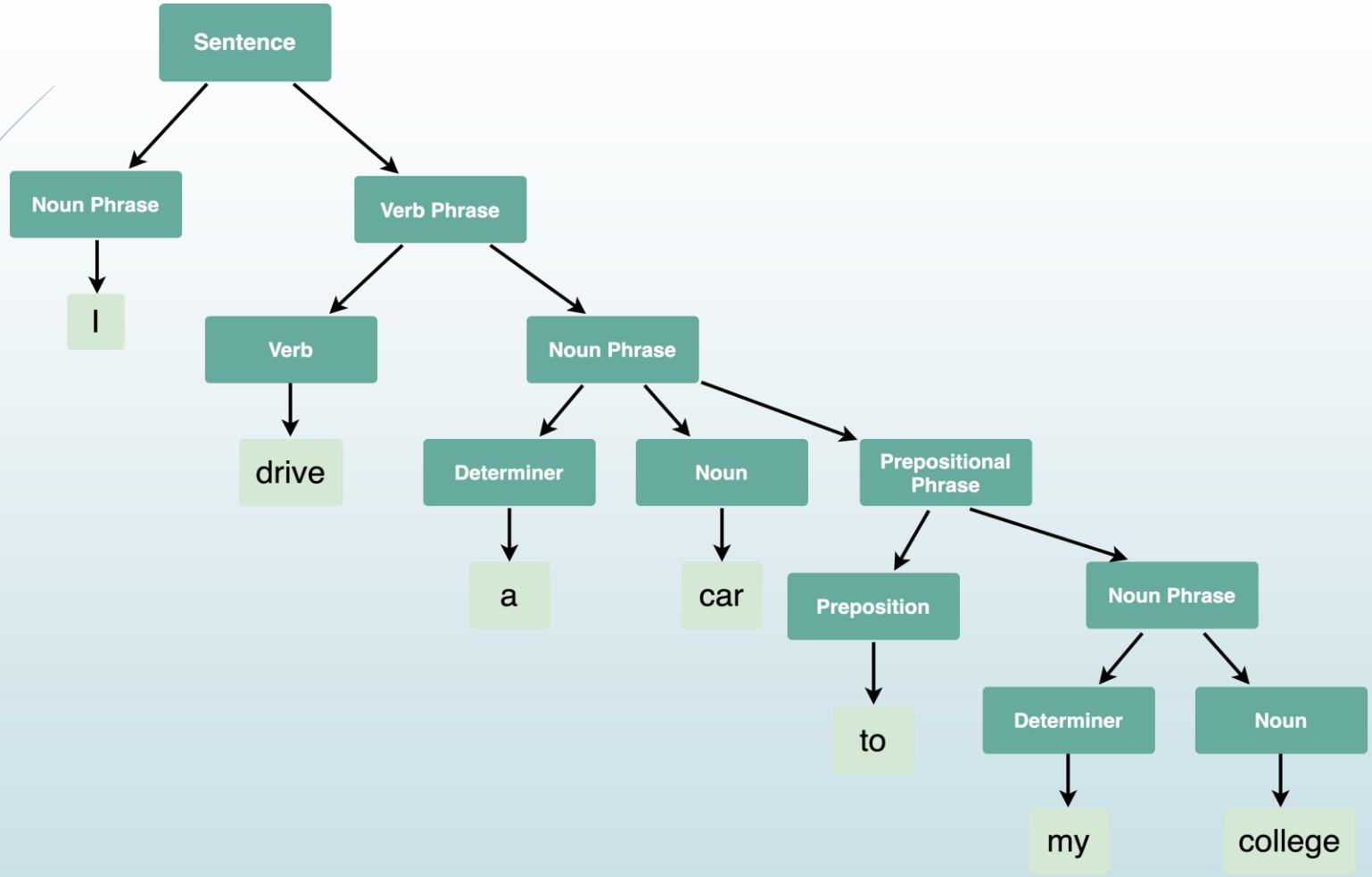
Automatic analysis of phrase structure relation in a sentence

Output: Syntactic structure tree of the parsing sentence

NLP Tools (2) Syntactic Parser

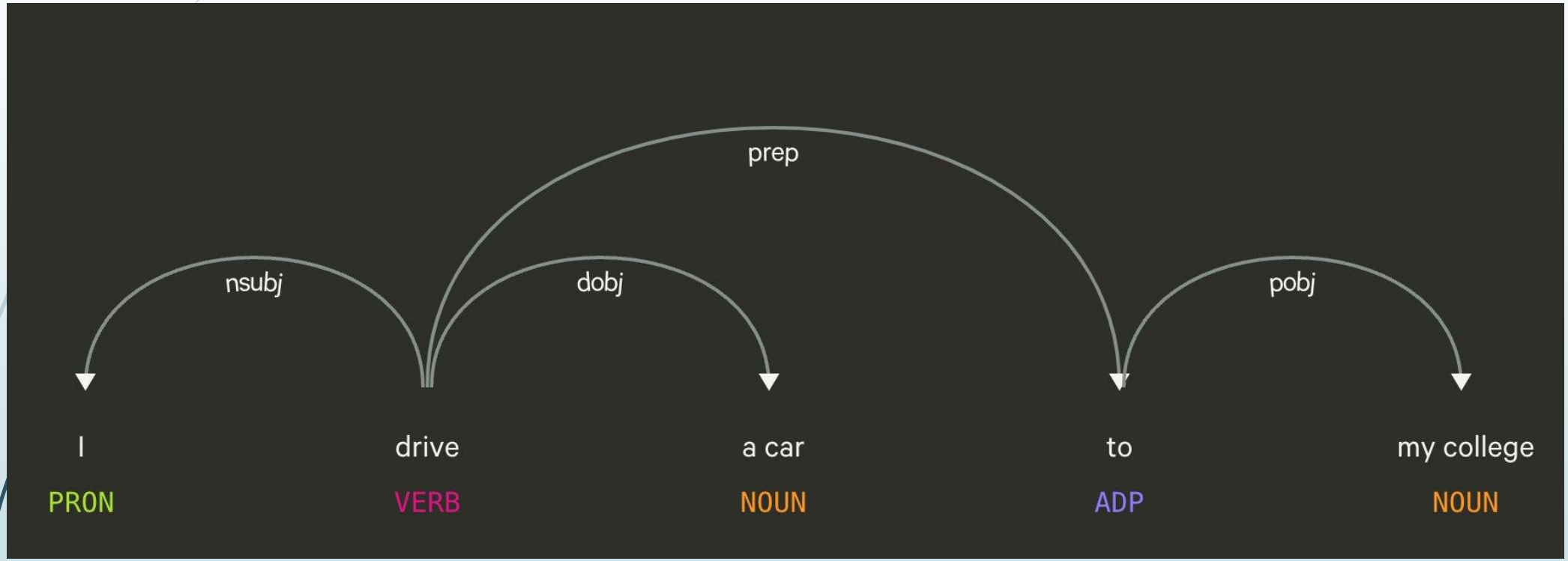
Streamlining Text Mining

- Several Processes, such as POS (Part-of-Speech) Tagging and NER (Named Entity Recognition), come into play to streamline Text Mining tasks.
- POS tagging, for instance, works out all grammatical functions played by each word in a sentence.
- That gives rise to a tree-shaped scheme, called dependency tree!
- We may also have some complementary information provided by syntactic tree, providing us with phrase-level information.
- Some examples are shown in the next slides.


Sentence
Noun Phrase
Verb Phrase
I
Verb
Noun Phrase
drive
Determiner
Noun
Prepositional Phrase
a
car
Preposition
Noun Phrase
to
Determiner
Noun
my
college

Syntax Tree

Dependency Tree



Syntactic and Dependency Tree

- A few differences are drawn below:
- Dependency Tree:
  - Shows direct relationships between words;
  - Each word connects directly to its head word
  - Focus on word-to-word relationships
  - No intermediate nodes
- Syntactic Tree:
  - Shows hierarchical phrase structure
  - Uses intermediate nodes
  - Groups words into constituents
  - Shows how phrases are built up

Syntactic and Dependency Tree

- Dependency Tree:
  - Better for showing grammatical relationships
  - Useful for relation extraction
  - More compact representation
  - Easier to process computationally
- Syntactic Tree:
  - Better for showing phrase structure
  - Useful for understanding sentence composition
  - Shows nested relationships
  - Important for grammar analysis

Syntactic and Dependency Tree

- Dependency Tree:
  - Word level analysis
  - Focus on functional relationships
  - Direct grammatical connections
- Syntactic Tree:
  - Phrase level analysis
  - Focus on structural composition
  - Hierarchical organization

NER – Named Entity Recognition

- Named Entity Recognition serves as a crucial preprocessing component in text analysis and NLP workflows.
- Its integration into the preprocessing pipeline enhances text understanding and preparation for downstream tasks.

NER – Named Entity Recognition

- Smart Content Filtering - Isolates meaningful entities from text automatically - Helps focus analysis on key information (e.g., extracting prominent figures and locations from news articles)
- Enhanced Text Cleaning - Removes non-essential content strategically - Focuses on maintaining identified entities while eliminating noise
- Advanced Feature Creation - Transforms recognized entities into model-ready features - Adds entity-type context (e.g., Person, Location, Organization) to improve model performance
- Standardization Techniques - Creates consistent entity representations across texts - Enables anonymization through entity-type placeholders (e.g., converting "Sarah Johnson" to ``)
- Knowledge Connection - Identifies entities for external database linking - Enriches content through connections to knowledge bases (e.g., linking company names to their database entries)
- Intelligent Text Segmentation - Preserves multi-word entities as single units - Improves tokenization accuracy (e.g., maintaining "San Francisco" as one entity)
- Specialized Processing - Adapts to domain-specific needs (e.g., medical terms, financial indicators) - Extracts field-relevant entities (e.g., gene names in biological texts, company symbols in financial reports) This structured preprocessing approach using NER helps prepare text data more effectively for various analytical tasks, ensuring important entities are properly identified and handled throughout the analysis pipeline.

NER – an example

Let's say we want process a sentence and map entities in it.

"Tim Cook met with Microsoft executives in Seattle last Friday to discuss AI developments worth $50 million."

NER output

- The output shows these entities:
- "Tim Cook" - **PERSON** (People, including fictional)
- "Microsoft" - **ORG** (Companies, agencies, institutions)
- "Seattle" - **GPE** (Countries, cities, states)
- "Friday" - **DATE** (Absolute or relative dates or periods)
- "AI" - **ORG** (Companies, agencies, institutions)
- "$50 million" - **MONEY** (Monetary values, including unit)

*GPE stands for Geo-political Entity.