NLP 101

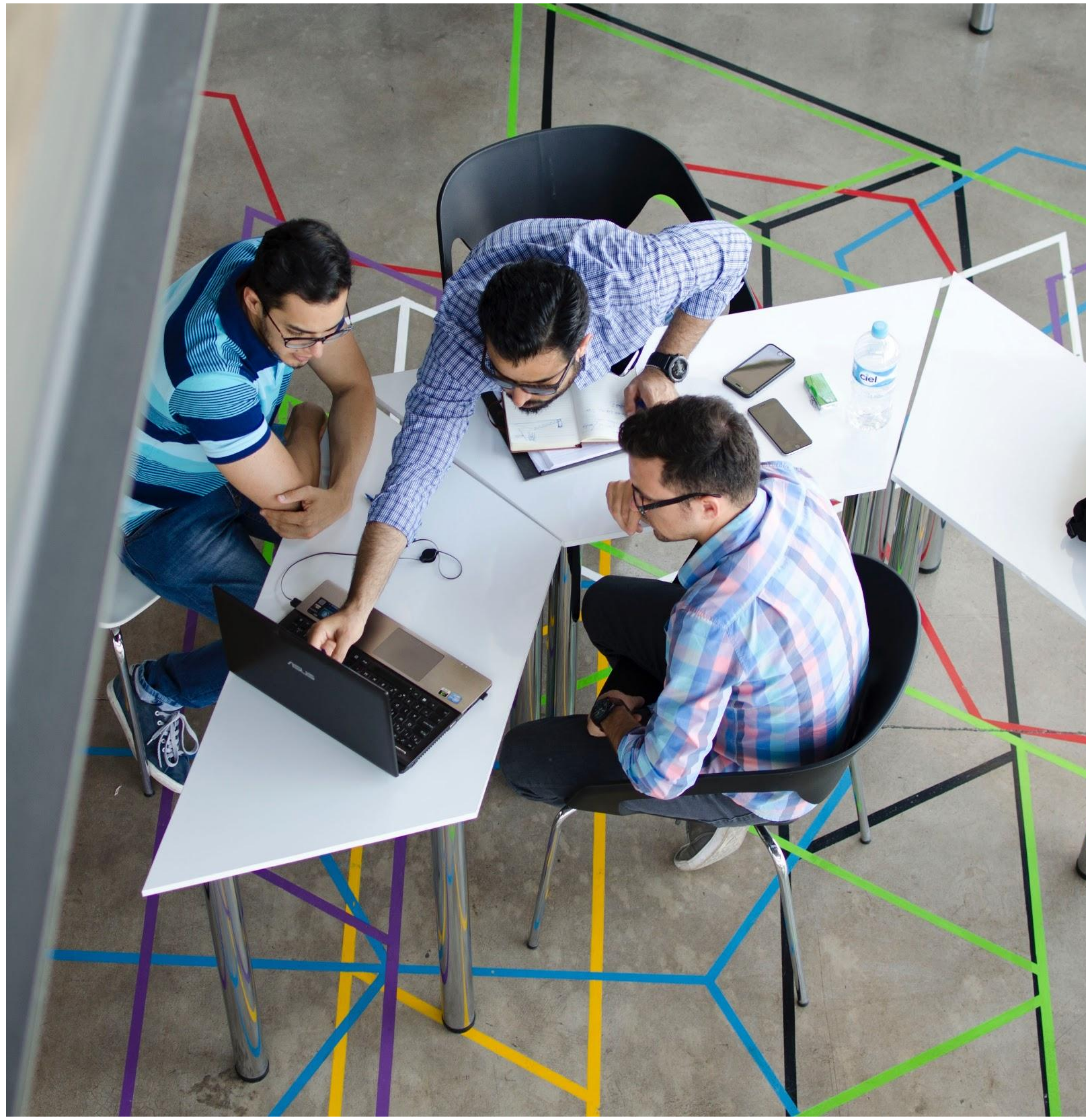
Data Science Development Series

Brian O'Connor
06/17/2020

PANDERA®

NLP 101 Agenda

- What is Natural Language Processing
- NLP Techniques
- Where to Apply NLP
- NLP Development
  - Text Preprocessing and Feature Engineering
  - ML Models
- Demonstration

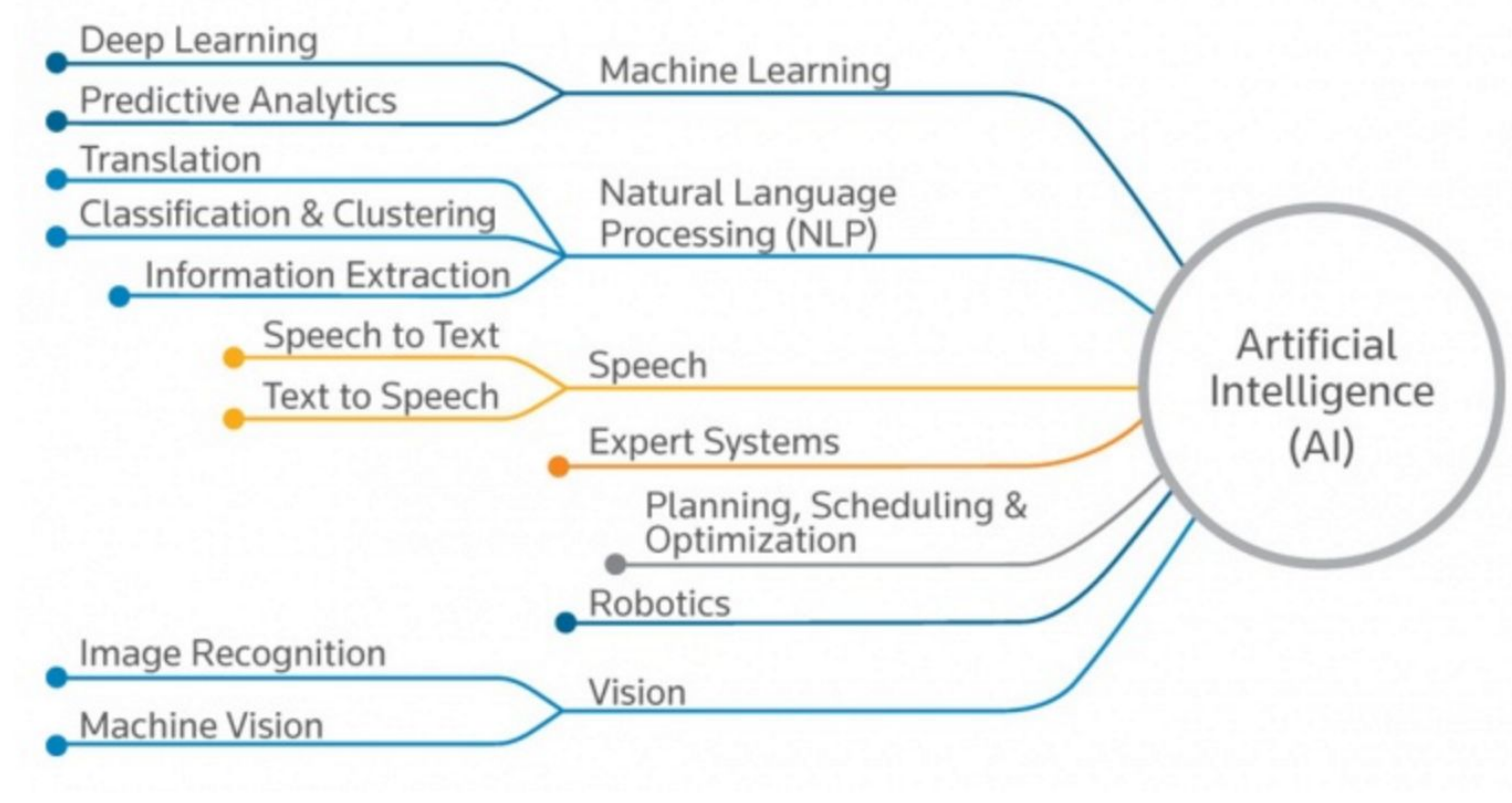


NLP Explained

What is NLP?

NLP is field of Artificial Intelligence that is focused on enabling computers to understand and process human languages, to get computers closer to human level understanding of language.

NLP is important because it helps resolve ambiguity in language and adds useful numeric structure to the data for many downstream applications, such as speech recognition or text analytics.



Source: Neota Logic

Why is NLP Difficult?

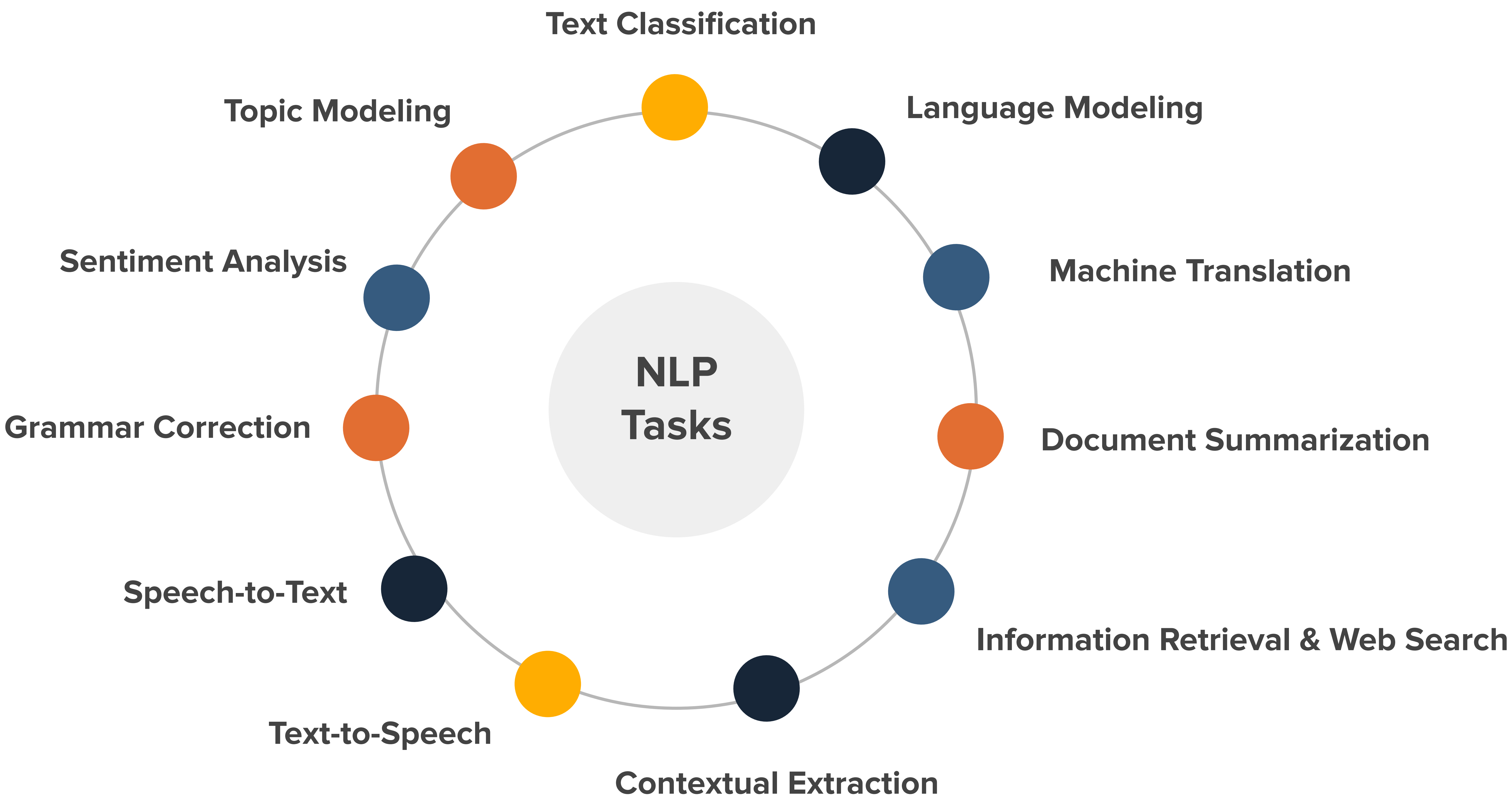
Natural language is highly ambiguous and inherently high dimensional/sparse. In human language, context and order are extremely important and linguistic patterns are not always typical. To add to the complexity, humans use slang, sarcasm and metaphors often. Below is a summary of Natural Language vs Computer Languages (programming).

Parameter	Natural Language	Computer Languages
Ambiguous	They are ambiguous in nature.	They are designed to unambiguous.
Redundancy	Natural languages employ lots of redundancy.	Formal languages are less redundant.
Literalness	Natural languages are made of idiom & metaphor.	Formal languages mean exactly what they want to say.

Ambiguity: "John kissed his wife, and so did Sam"

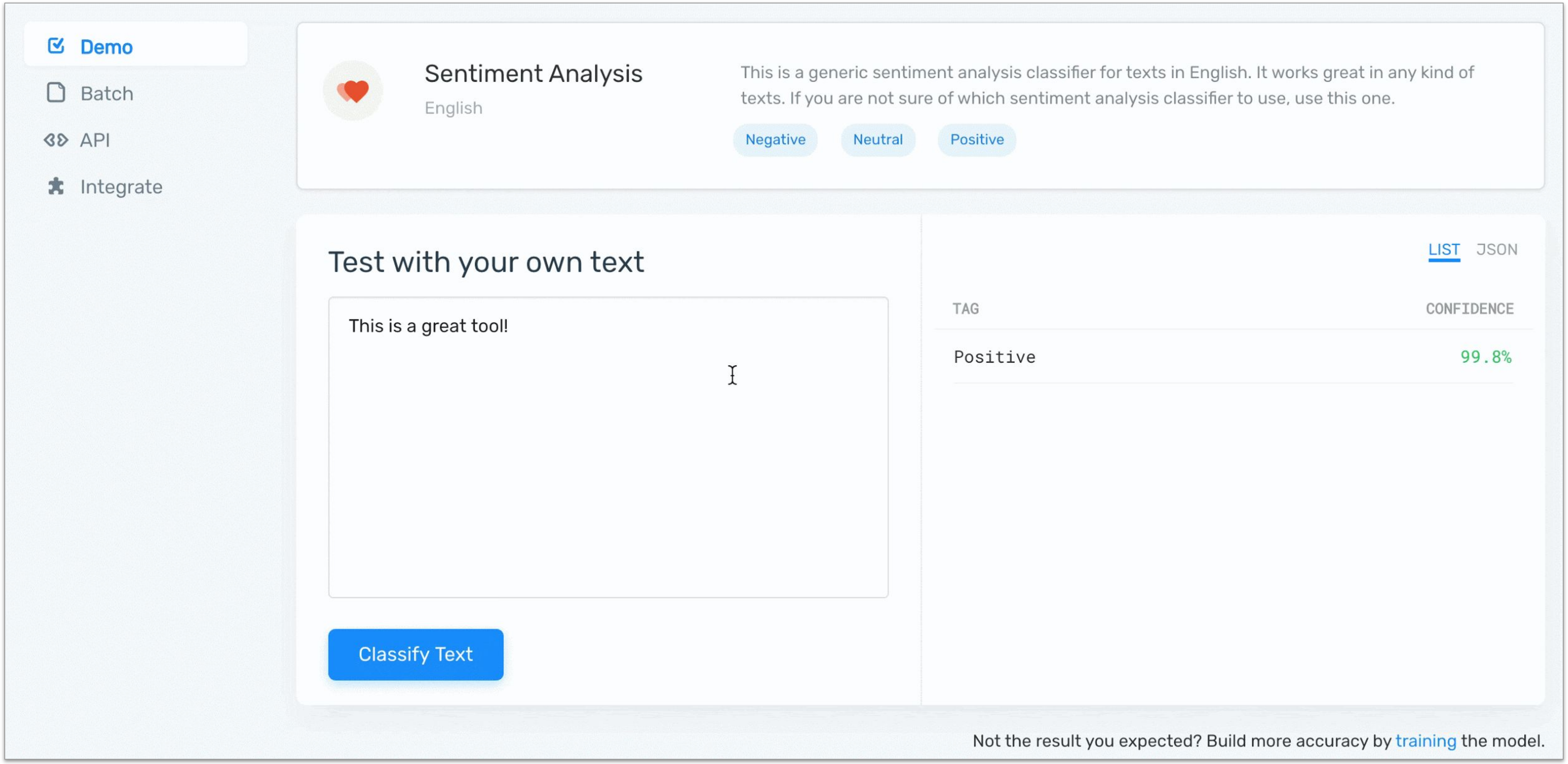
Source: Guru99

How Can NLP Be Applied?



Sentiment Analysis

Twitter Sentiment



Source: [MonkeyLearn]

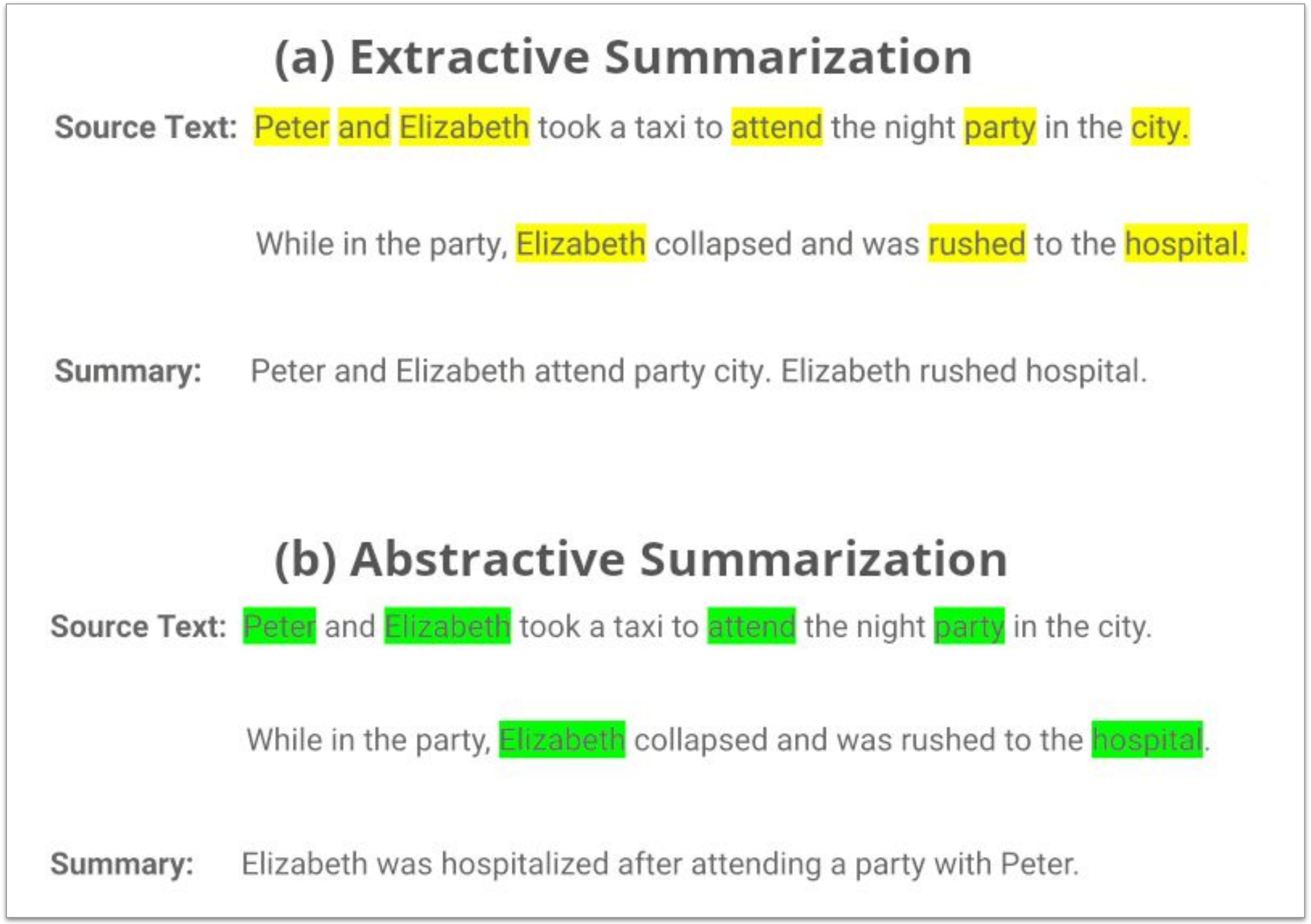
Text Summarization

Extractive Summarization: identify the important sentences or phrases from the original text and extract only those from the text

- Uses: Pull out important comments, keeping the raw version

Abstractive Summarization: generate new sentences from the original text to represent a summary

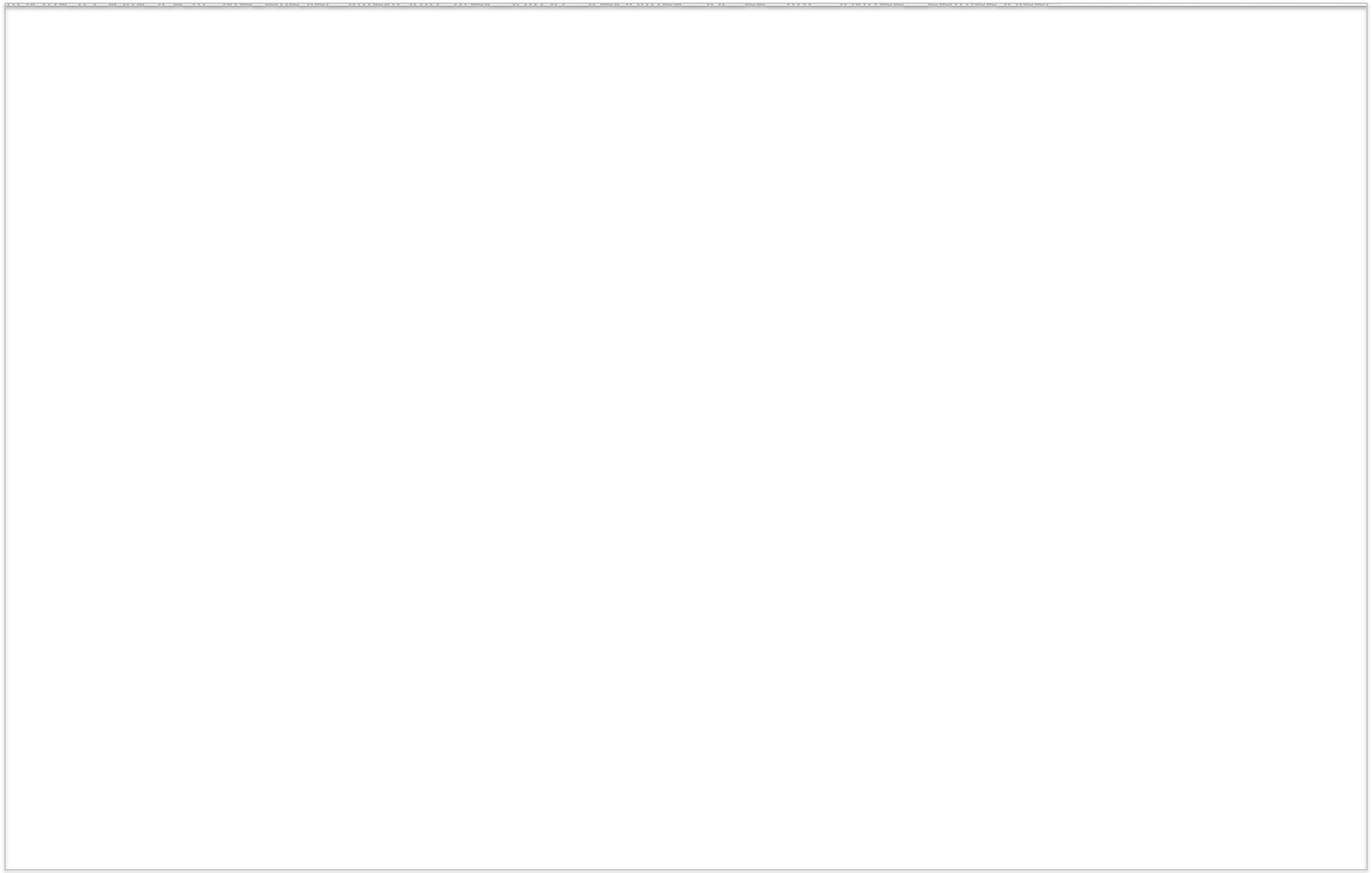
- Uses: Summarizing comments that consist in same topic class and sentiment



Machine Translation

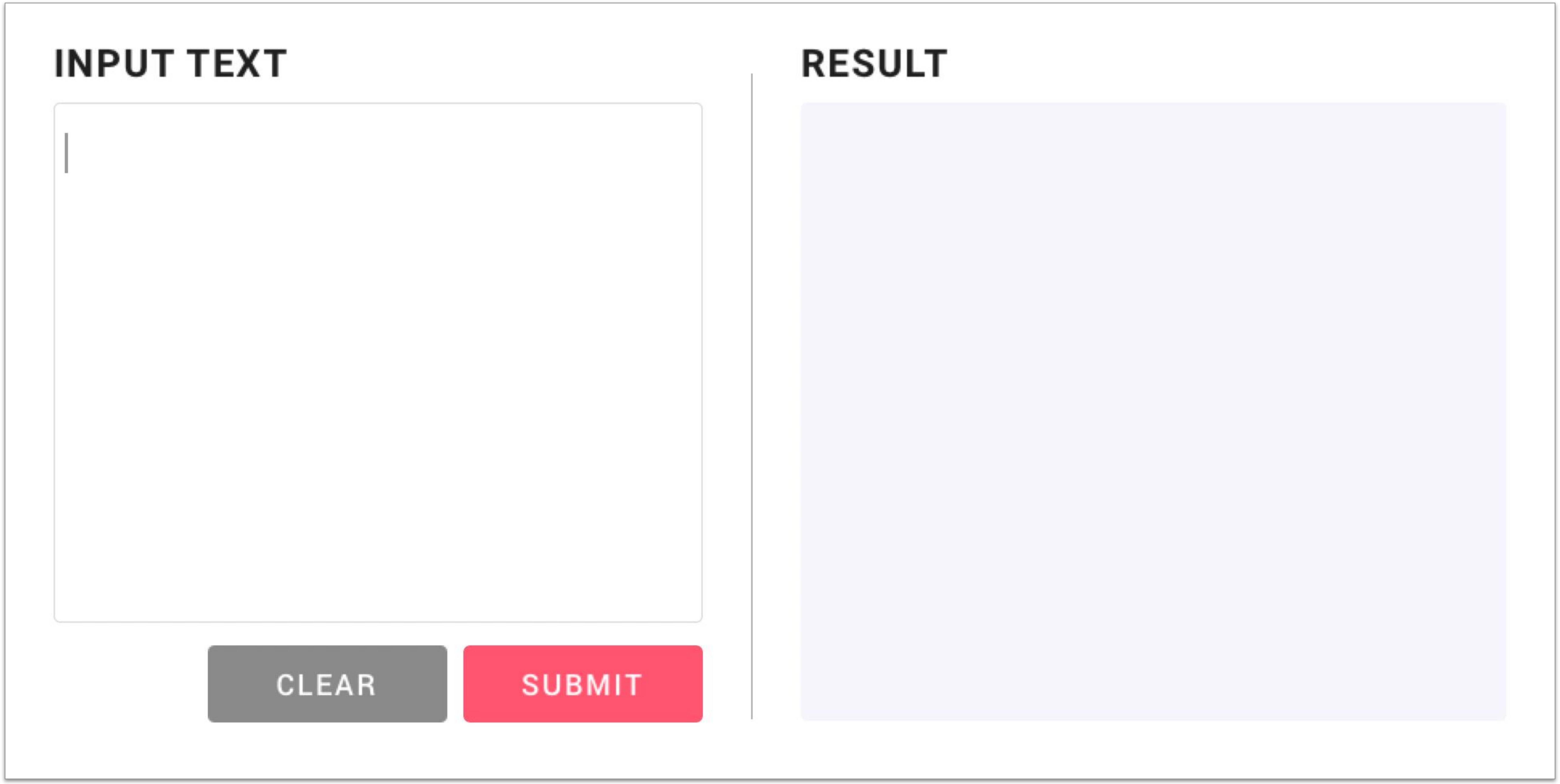
Source: [Google Translate]

Search via Keyword Extraction



Source: [Towards Data Science]

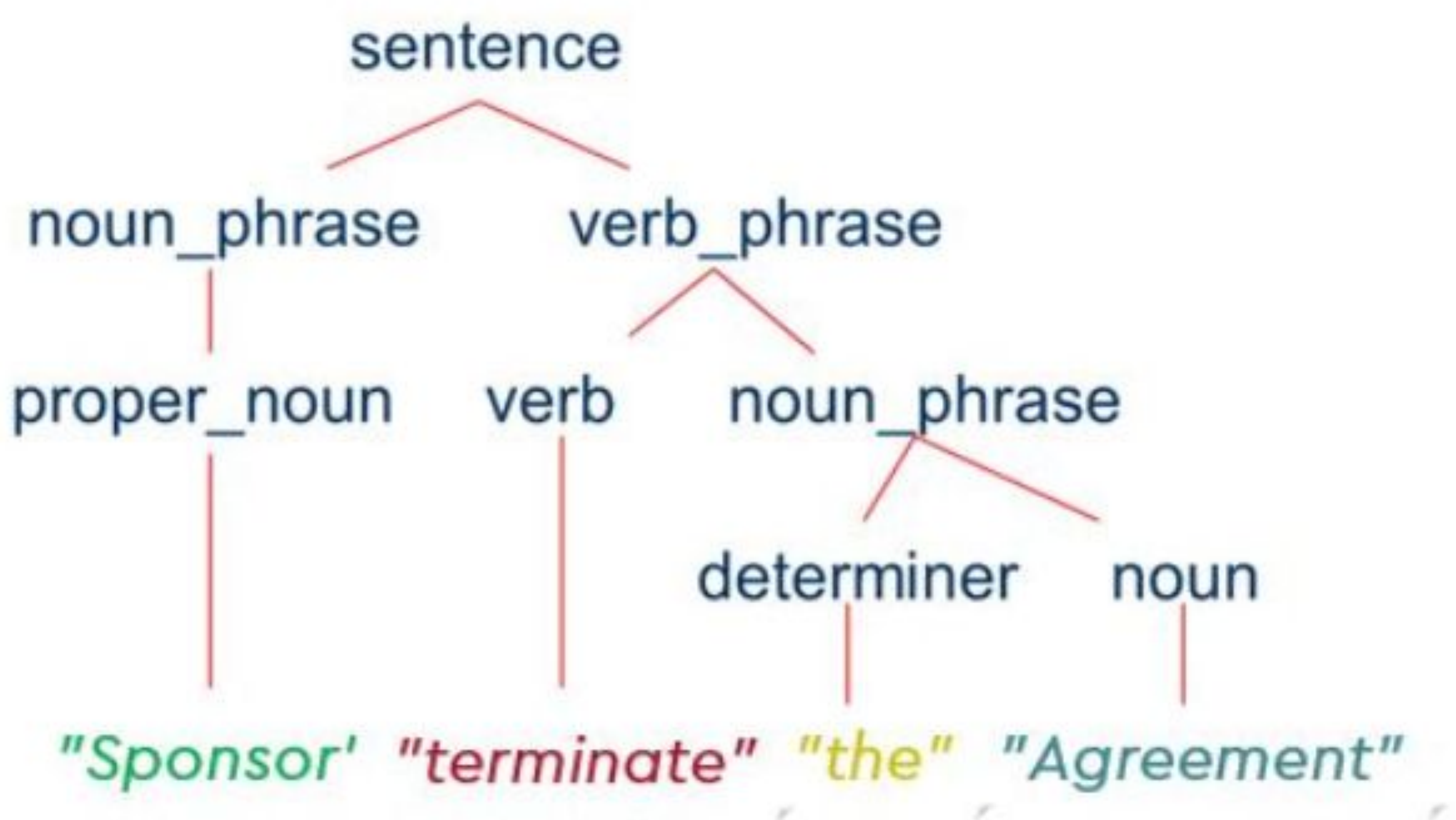
Text Classification



Source: Allganize

Document Understanding

Contracts



3 TERM
3.1 This Agreement shall commence on the Commencement Date and shall continue, unless terminated earlier in accordance with this Agreement, for the Term. On the expiry of the Term, this Agreement shall terminate automatically without notice.
4. SPONSORSHIP FEE
4.1 In consideration of the Rights granted to the Sponsor, the Sponsor shall pay Procurement Events Limited the Fees, in the instalments and on the dates set out in the Booking Form.
4.2 All amounts payable to Procurement Events Limited under this Agreement are to be paid in full without any discount, withholding, deduction, set off or abatement either: (a) within 30 days from the date of the invoice; or (b) prior to the date of the Event and/or Publication (as applicable)
4.3 All sums payable under this Agreement are exclusive of VAT, which shall be payable in addition within thirty (30) days of the date of an applicable VAT invoice.
4.4 Without prejudice to any other right or remedy of Procurement Events Limited, if the Sponsor fails to make any payment of any sums under this Agreement on the due date for payment then Procurement Events Limited may charge the Sponsor interest on the unpaid amount at the rate of 4% per year above the Bank of England base rate from the due date for payment until payment is received in full by Procurement Events Limited.
4.5 Without prejudice to any other right or remedy of Procurement Events Limited, if the Sponsor fails to make any payment of any sums under this Agreement on the due date for payment then Procurement Leaders Limited may at its discretion: (a) suspend delivery of the Rights; (b) cancel the Event and/or Publication; and/or (c) refuse to allow the Sponsor entry to the Venue.

Tag colors:

ACTION | ITEM | ORGANIZATION | LOCATION | TIME | MONEY

Industries to Apply NLP

- Legal
- Financial Services
- Retail
- Travel and Hospitality
- Healthcare
- Advertising/Marketing

Practical NLP Uses Cases

Legal

- Contract Review
- Legal Research and Electronic Discover
- Virtual Agents for Legal Advice
- Document Generation Automation

Retail

- Advertising/Marketing Strategy and Messaging Optimization
- Chatbots and Virtual Agents
- Customer Experience with Personalized Messaging
- Customer and Product Review Analysis

Healthcare

- Clinical Documentation
- Data Mining Research
- Computer Assisted Coding
- Personalized Patient Care with Virtual Assistants
- Clinical Trial Matching

Financial Services

- Sentiment Analysis on the Market (News/Twitter)
- Fraud Detection
- Spam Detection
- Sales and Marketing Campaign Management
- Creditworthiness Assessment

Travel and Hospitality

- Customer and Product Review Analysis
- Machine Translation for In-Flight Entertainment
- Social Media – Consumer Feedback and Interaction Analysis
- Customer Complaint Resolution

Advertising/Marketing

- SEO optrimization
- Customer-Specific Targeting
- Text Summarization for Market Trends
- Efficient Content Generation

Where Can NLP Go Wrong?

MIT Technology Review

Why Microsoft Accidentally Unleashed a Neo-Nazi Sexbot

It's not surprising that Microsoft's chatbot spewed racist invective, but here's how it could have been avoided.

by **Rachel Metz**

March 24, 2016

When Microsoft unleashed Tay, an artificially intelligent chatbot with the personality of a flippant 19-year-old, the company hoped that people would interact with her on social platforms like Twitter, Kik, and GroupMe. The idea was that by chatting with her you'd help her learn, while having some fun and aiding her creators in their AI research.

YIKES!

NLP Development

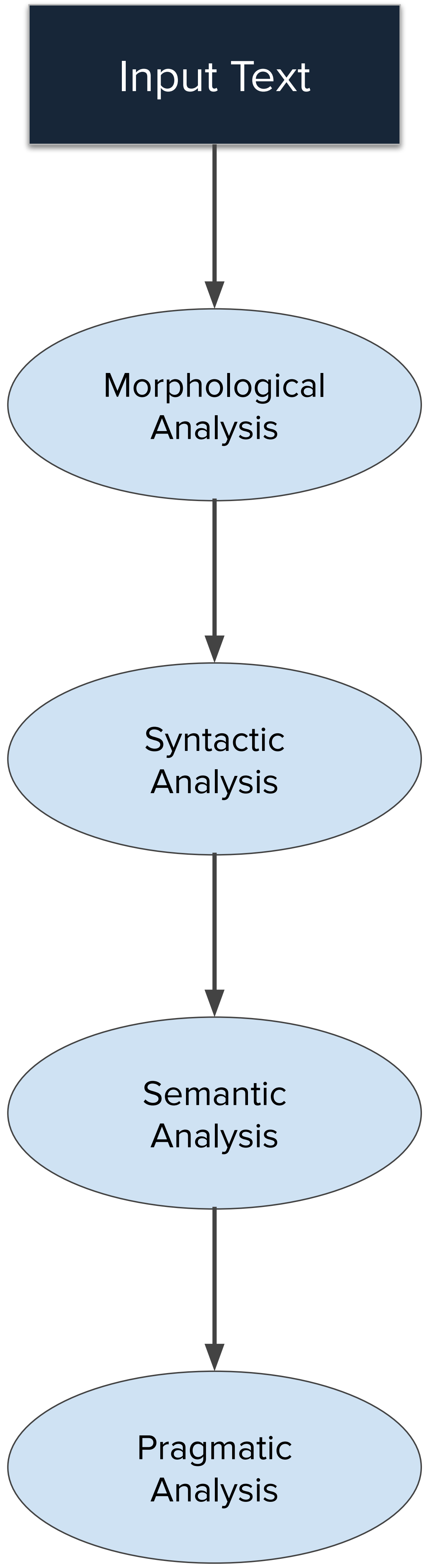
Basic Components of NLP

Morphological Analysis: Study of the structure and formation of words.

Syntactic Analysis: Focus on the orders of words which can affect a sentences meaning.

Semantic Analysis: Focus on the literal meaning of words, phrases, sentences. Based on dictionary meaning of words/phrases.

Pragmatic Analysis: Focuses on the entire conversation to derive context and uncover the intended meaning. Also, can integrate discourse to chain together the sentences for meaning based on context.



Implementing NLP

Data Preprocessing and Cleansing

Feature Extraction

Model Development and Evaluation

Tools of the Trade

Python
- spaCy
- NLTK (TextBlob for a nice interface)
- Gensim

R
- tm (text mining)
- OpenNLP
- RWeka

Java
- Stanford Core NLP

Cloud Giants
- Google’s NLP API, AutoML NLP
- Amazon’s Comprehend
- IBM Watson

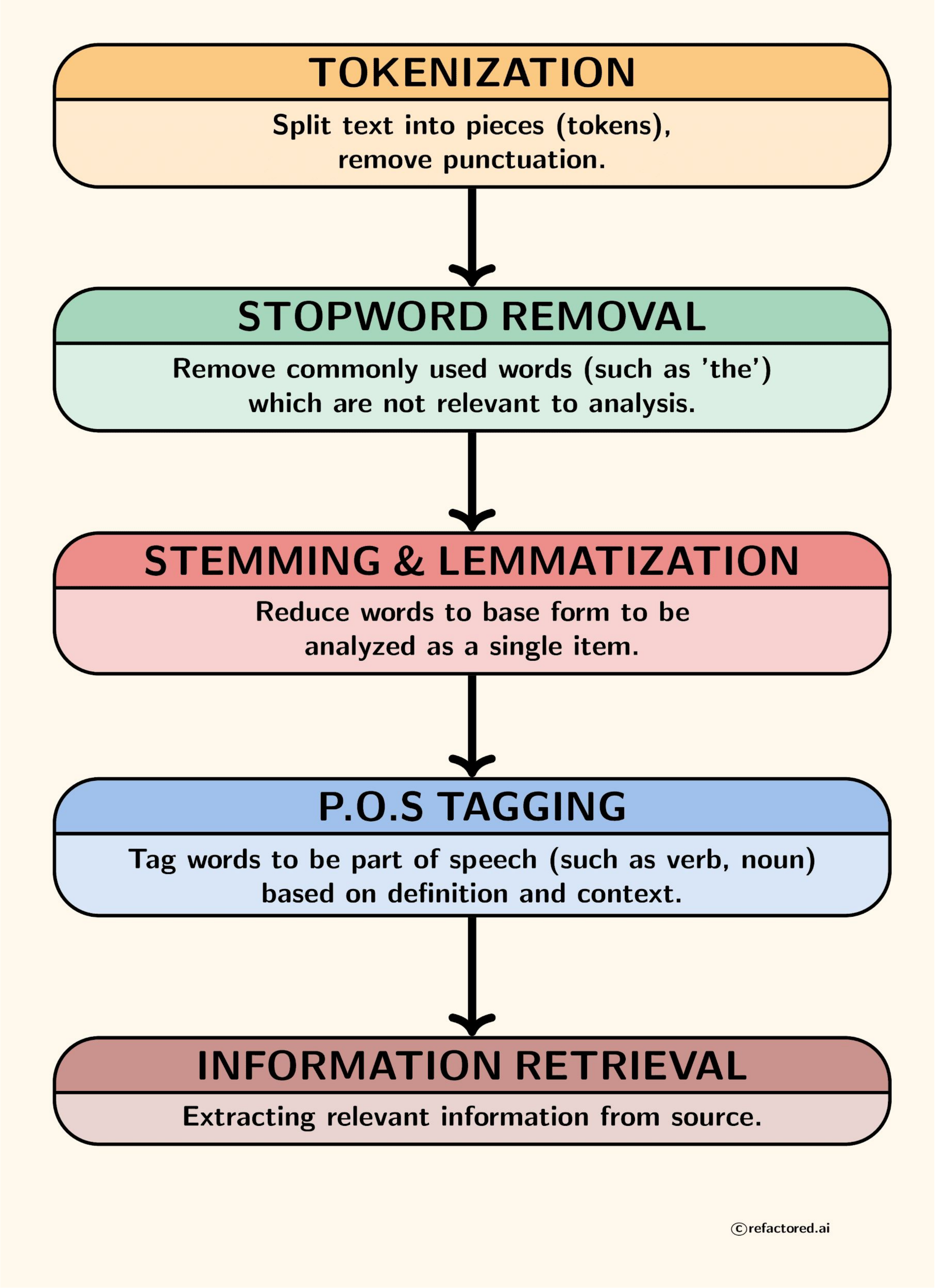
Deep Learning Frameworks
- Tensorflow
- Pytorch

Data Preprocessing & Cleansing

Why Preprocessing in NLP?

- Text data is unstructured and highly ambiguous
- Cleaning this data means removing less useful parts of text
- Normalization consists linguistic reduction thru Stemming, Lemmatization and other form of standardization
- The data should be in the form that model can understand
- Machine learning algorithms needs numbers as input. We are preparing for that to take place.

Preprocessing & Cleansing Methods



Text Cleansing: Remove punctuation, downcase all words, and remove possessive pronouns

Tokenization: Splitting sentences from paragraphs and words from the sentences. Can be done via uni-gram, bi-grams, tri-grams, n-gram.

Stopword Removal: Any piece of text which is not relevant to context of data can be output can be specified as noise and removed. Examples: “the”, “a”, “myself”, etc.

Stemming: Aims to identify stem form of the word and use it in lieu of the word itself (rule based stripping of suffixes from word).

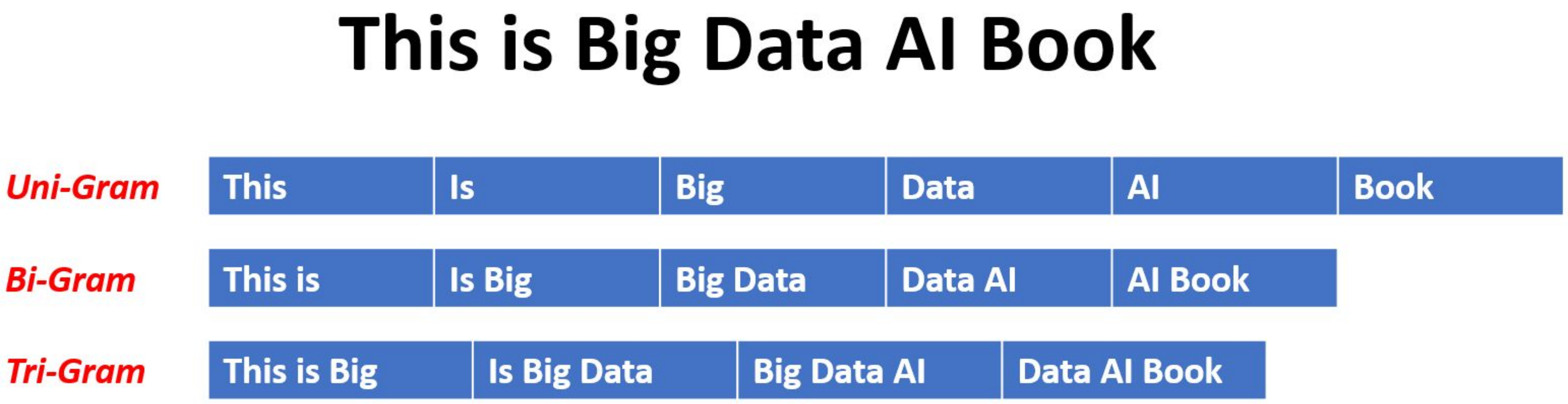
Lemmatization: Process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form. This is a form of morphological analysis.

POS Tagging: The primary target of Part-of-Speech tagging is to identify the grammatical group of a given word. In this stage, we look for relationship within the sentence and assign a corresponding tag to the word.

Preparing the Data

Tokenizing the text to the word or phrase level allows Data Scientists to begin creating features for a Machine Learning model. Tokenization can be down at a defined level of “n”. Below are some details regarding tokenization:

- n-gram (uni-gram, bi-gram, tri-gram… and so on)
- higher the n-gram, higher the computing cost
- unigram, bigram and trigram usually works best
- Example:



Feature Extraction

Why Feature Engineering in NLP?

- High quality data ➔ better ML model ➔ better quality predictions
- Data needs be in the form that machine/model can understand (not unstructured text fields)
- Since Machine learning algorithms cannot work with raw text directly, we need to convert the text into vectors of numbers (Vector Representation or Text Representation).
- The feature engineering process is typically called *feature extraction* in the NLP world

Vector Representation Methods

Options for Feature Extraction Techniques:

Vector Representation (sometime called Text Representation and often you'll see used interchangeably with Word Embeddings) is the process of converting text into numerical representation, where words or phrases from the vocabulary are mapped to vectors of real numbers. This is a necessary step to make text data mathematically computable and to provide inputs into Machine Learning models. Below are 4 common/simplistic techniques to complete this task:

- Count Vectors
- TF-IDF Vectors
- Continuous Bag of Words (CBOW)
- Word2Vec

Feature Extraction Comparison

Technique	Description	Type	Complexity
Count Vectors	Simply a count of word or term frequencies.	Frequency Based (Deterministic)	Easiest
TF-IDF	Term frequency–inverse document frequency. A statistical measure used to evaluate the importance of the word to a document in the collection/corpus.	Frequency Based (Deterministic)	Easy
Continuous Bag of Words	CBOW predicts the probability of a word given a context. A context is based on the surrounding words.	Prediction Based, Neural Network (Word2Vec)	Difficult
Skip Gram	This is the complete opposite of CBOW. Skip-Gram attempts to predict its surrounding words based on the given word.	Prediction Based, Neural Network (Word2Vec)	Difficult

Count Vectors

Example

```
I like this movie, it's funny.
I hate this movie.
This was awesome! I like it.
Nice one. I love it.
```

	awesome	funny	hate	it	like	love	movie	nice	one	this	was
0	0	1	0	1	1	0	1	0	0	1	0
1	0	0	1	0	0	0	1	0	0	1	0
2	1	0	0	1	1	0	0	0	0	1	1
3	0	0	0	1	0	1	0	1	1	0	0

TF-IDF Vectors

Equation

$$TFIDF(term) = TF(term) * \mathrm{IDF}(term)$$

- **Term Frequency (TF)**: a scoring of the frequency of the word in the current document.

$$TF(term) = \frac{Number\ of\ times\ \mathbf{term}\ appears\ in\ a\ document}{Total\ number\ of\ items\ in\ the\ document}$$

Term Frequency Formula

- **Inverse Term Frequency (ITF)**: a scoring of how rare the word is across documents.

$$IDF(term) = \log\left(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ with\ \mathbf{term}\ in\ it}\right)$$

Inverse Document Frequency Formula

Example

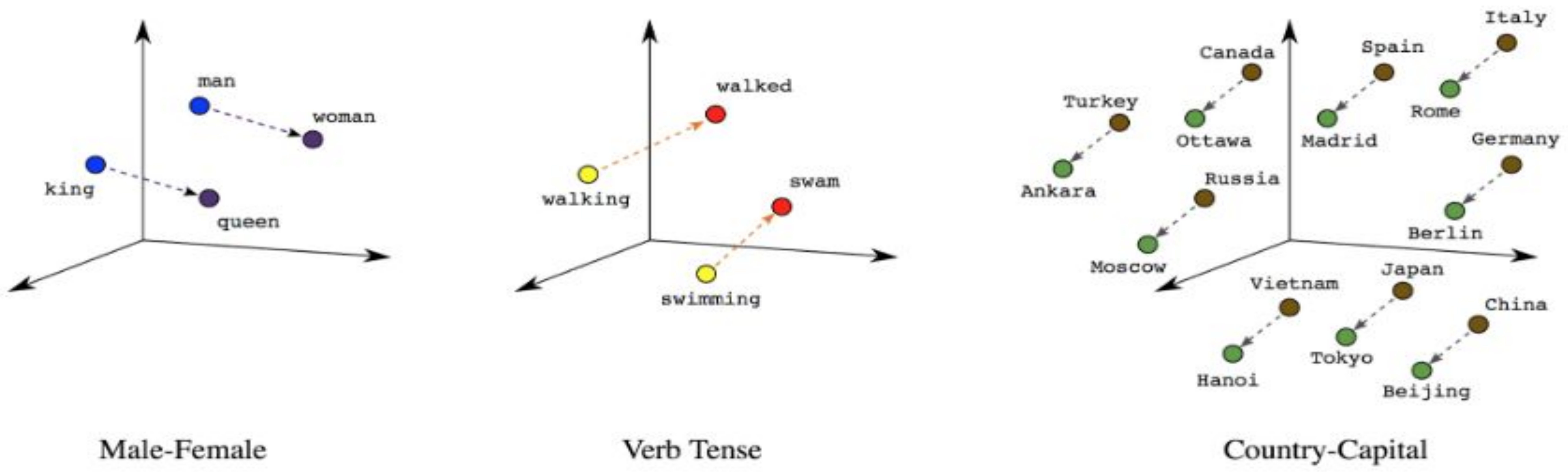
```
I like this movie, it's funny.
I hate this movie.
This was awesome! I like it.
Nice one. I love it.
```

	awesome	funny	hate	it	like	love	movie	nice	one	this	was
0	0.000000	0.571848	0.000000	0.365003	0.450852	0.000000	0.450852	0.000000	0.000000	0.365003	0.000000
1	0.000000	0.000000	0.702035	0.000000	0.000000	0.000000	0.553492	0.000000	0.000000	0.448100	0.000000
2	0.539445	0.000000	0.000000	0.344321	0.425305	0.000000	0.000000	0.000000	0.000000	0.344321	0.539445
3	0.000000	0.000000	0.000000	0.345783	0.000000	0.541736	0.000000	0.541736	0.541736	0.000000	0.000000

Word Embeddings

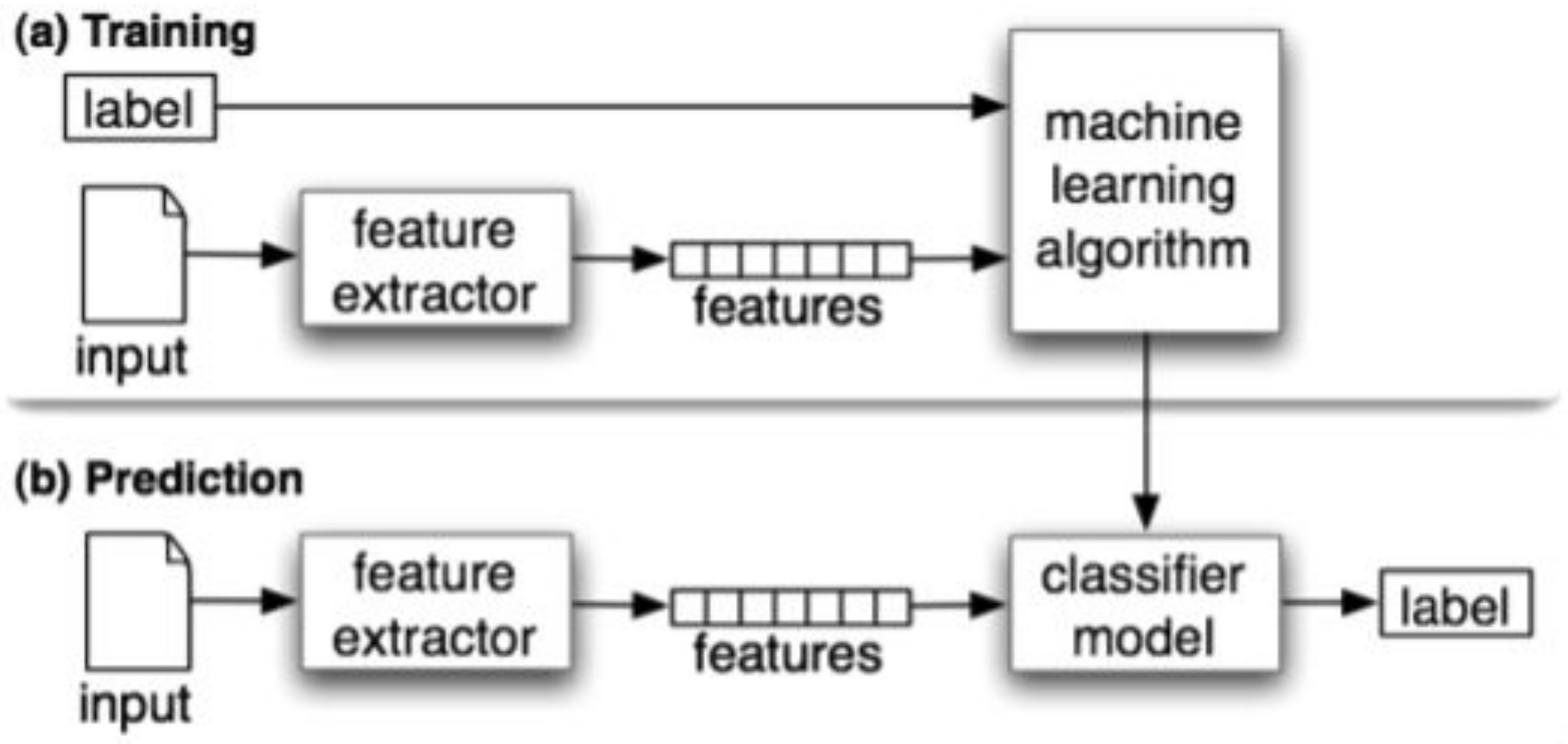
Words have meaning(s) associated with them and as a result it can be represented with word tokens in a dense vector space where location and distance between words indicates how similar they semantically are. Word2Vec is a combination of both the CBOW (continuous bag of words) and the Skip-gram model.

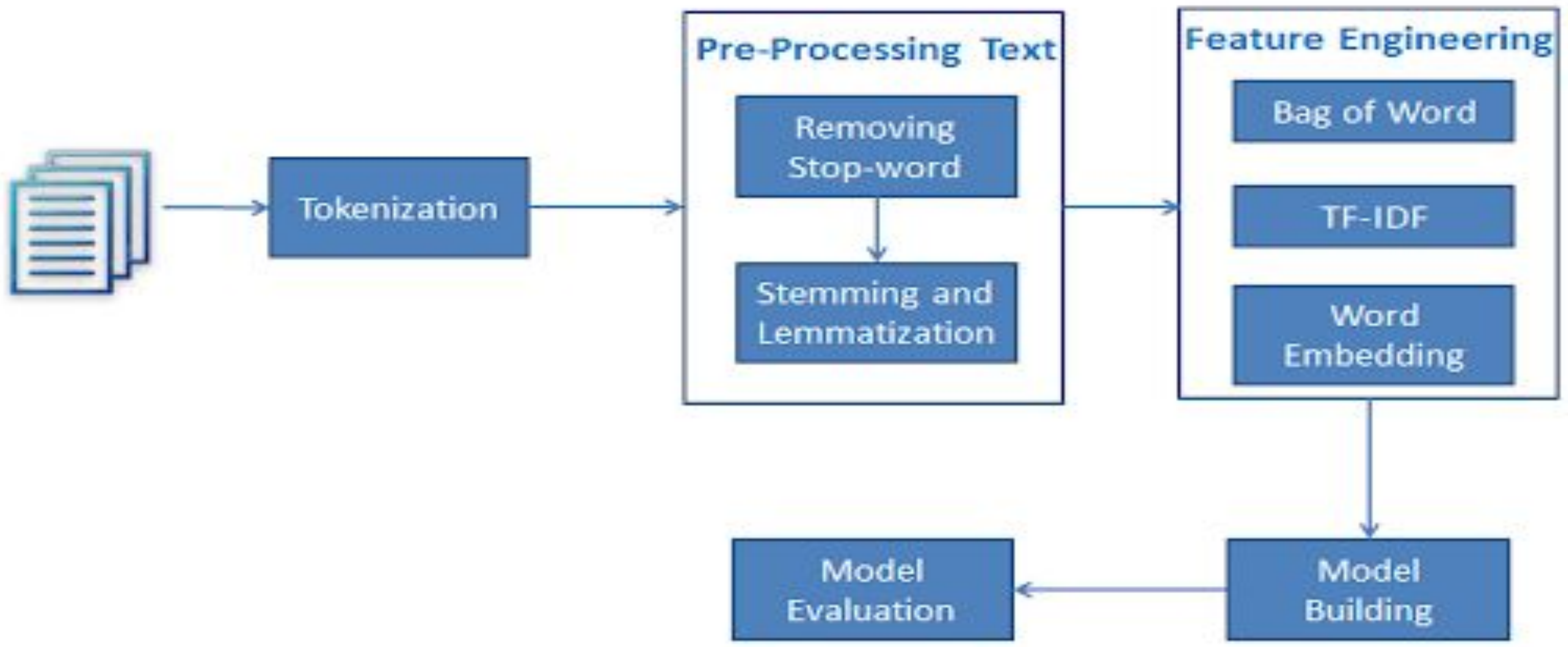
Fortunately Google has trained a **word2vec** (by google) model that takes a text corpus as input and produces the word vectors as output. It basically translates words into complex vector representation.



Model Development

Model Architecture





Algorithms

For text classification (demo example), the most commonly used algorithms are:

- Naive Bayes Classifier
- Support Vector Machine <- **Demo Algorithm**
- Tree based algorithms(Random Forest & XGBoost)
- Logistic Regression (with word2vec and doc2vec)
- Neural Networks

Resources

Demo Details

GitHub Repository: https://github.com/bpoconnor3/data-science-development-series
Raw Dataset: https://www.kaggle.com/c/learn-ai-bbc/data

The content in this demo is intended show the basics of Natural Language Processing (NLP). In the demo notebook, we will walk through the following NLP project tasks:

- Exploratory Data Analysis (EDA)
- Data Preprocessing & Cleansing
- Feature Extraction
- ML Model Development (Supervised Machine Learning Model)
- Model Evaluation and Interpretation

Resources

NLP Datasets:

- BBC News Classification: Over 2,000 Classified News Articles
- IMDB: 50k Movie Reviews
- Fake News: Unreliable News Classification

Courses:

- Udemy
- DataCamp

Notebook Repository:

- The Super Duper NLP Repo

Books:

- Hands-On Machine Learning with Scikit-Learn and TensorFlow
- Natural Language Processing with Python

Entry Level Blogs:

- Document Classification
- Text Preprocessing

Questions!