Natural Language Processing(NLP)

using ML & DL(TF) in Python

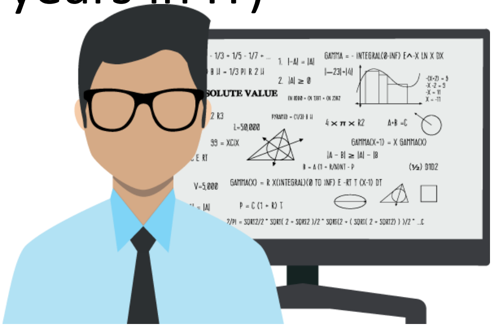
About Me

- Name: Shiv

- BSc and MSc in Mathematics from IIT Kharagpur, India

- PhD in Analytics

- Experience: 19 years (9 Years in Analytics and 10 years in IT)
- Current Role: Chief Data Scientist



Prerequisites

i. Knowledge of Python (Basic)

ii. Knowledge of basic Statistics

Take any course similar to following

AI, Basic Statistics, Basic Python, Basic R, ML (Overview)
https://www.udemy.com/ai-basic-statistics-basic-python-basic-r-ml-overview/?couponCode=AISPYRML

iii. Awareness of ML (Nice to have for Data Science professional)

Data science for AI and Machine learning using Python
https://www.udemy.com/data-scientist-for-ai-and-machine-learning-using-python/?couponCode=AIMLPY

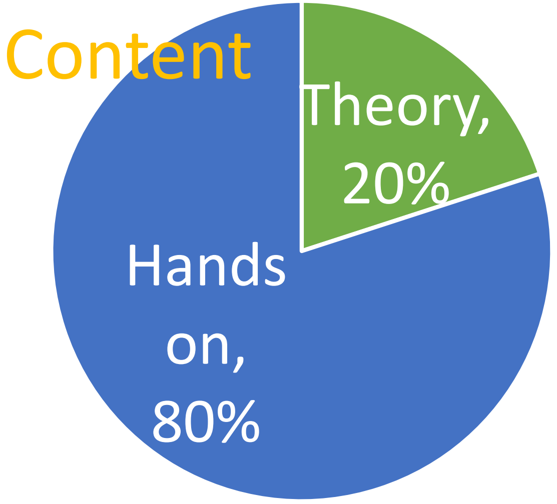
iv. Awareness of DL (Nice to have for Data Science professional)

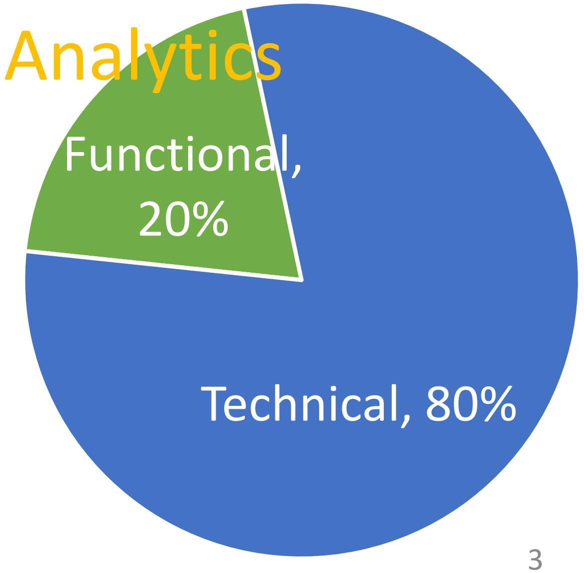
Deep Learning by TensorFlow (tf.keras) & Keras using Python
https://www.udemy.com/deep-learning-by-tensorflow-tfkeras-keras-using-python/?couponCode=DLPY010

v. Good Knowledge of any programming technology

vi. Ideal count of trainee 15 (classroom or webex)

Methodology


Content


Analytics

Knowledge Sharing Flow

Introduction

- **Text mining/analytics**
- What Is Natural Language Processing
- Why NLP is emerging
- Classic NLP vs Deep Learning NLP

- Why Deep Learning
  - Promise of Deep Learning
  - Types of Deep Learning Networks for NLP
  - Areas in NLP where DL methods are showing greatest success
- Applications of NLP

Basic

- String operations
- Regex
- String cleaning
  - Punctuation
  - Stop words
  - Search Engine words
  - Spelling Correction
  - Stemming
  - Lemmatization
  - Custom replace and removal
- Word Cloud

Hands on marked in green color

Knowledge Sharing Flow cont

Basic

- NLTK operations
  - Tokenizers
  - Part-of-speech tagging
  - Text Blob
  - Sentiment Analysis
  - Translation and Language Detection
  - Stemming and Lemmatization
  - Word Sense Disambiguation
  - BLEU Scores
- Stanford NLP

Intermediate

- Entity resolution
- Text to Features
  - One Hot Encoding
  - Count vectorizer
  - TF-IDF
- Word Embedding
- Word2vec
- GloVe: Global Vectors for Word Representation
- Custom Word Embedding by Word2vec

Hands on marked in green color

Knowledge Sharing Flow cont

Intermediate

- Word Sense Disambiguation
- Speech Recognition
  - Microphone
  - Audio Files
- Similarity between two strings
- Language Translation
- Computational Linguistics

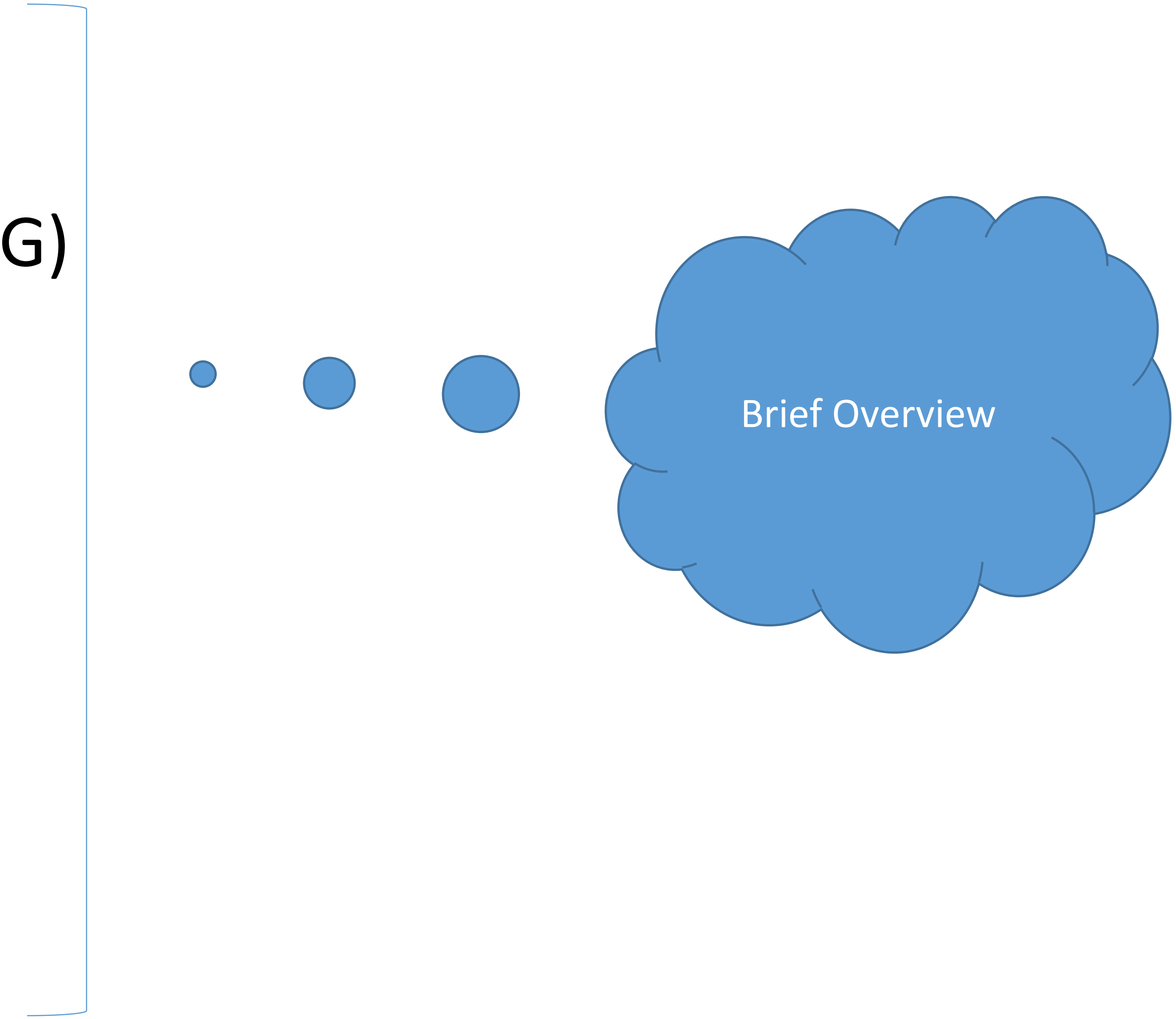
Hands on marked in green color

Advance

- Classifications using Machine learning
  - Random forest
  - Naive Bayes
  - Xgboost
- Classifications using Deep learning
  - With tf.keras on TF-IDF data
  - With tf.keras with inbuilt DL embedded layer
  - With tf.keras with Word Vector embedded layer (after taking average)
  - With tf.keras with Word Vector embedded layer (as it is in raw form)
- Sentiment analysis
- Clustering & Topic Modelling
- Search engine

Knowledge Sharing Flow cont

- Morphology
- Nltk corpus
- Natural language generation (NLG)
- Applications of NLP



Installations & Technology

- software_list.txt – Please follow the steps

- Anaconda (https://www.anaconda.com/download/) or any Python IDE

- Presentations and Code are uploaded in section (1 or 2)

Anaconda

- The open source version of Anaconda is an easy-to-install high performance Python with a package manager, environment manager and collection of 720+ open source packages
- conda install PACKAGENAME or pip install pydotplus
- Packages
  1. NumPy | numpy.org: N-dimensional array for numerical computation
  2. Pandas | pandas.pydata.org: Powerful Python data analysis toolkit
  3. SciPy | scipy.org: Collection of numerical algorithms and toolboxes, including signal processing and optimization
  4. MatPlotLib | matplotlib.org: Plotting library for Python
  5. Seaborn | stanford.edu/~mwaskom/software/seaborn/: Statistical data visualization
  6. Bokeh | bokeh.pydata.org: Interactive web visualization library
  7. SciKit-Learn | scikit-learn.org/stable: Python modules for machine learning and data mining
  8. NLTK | nltk.org: Natural language toolkit

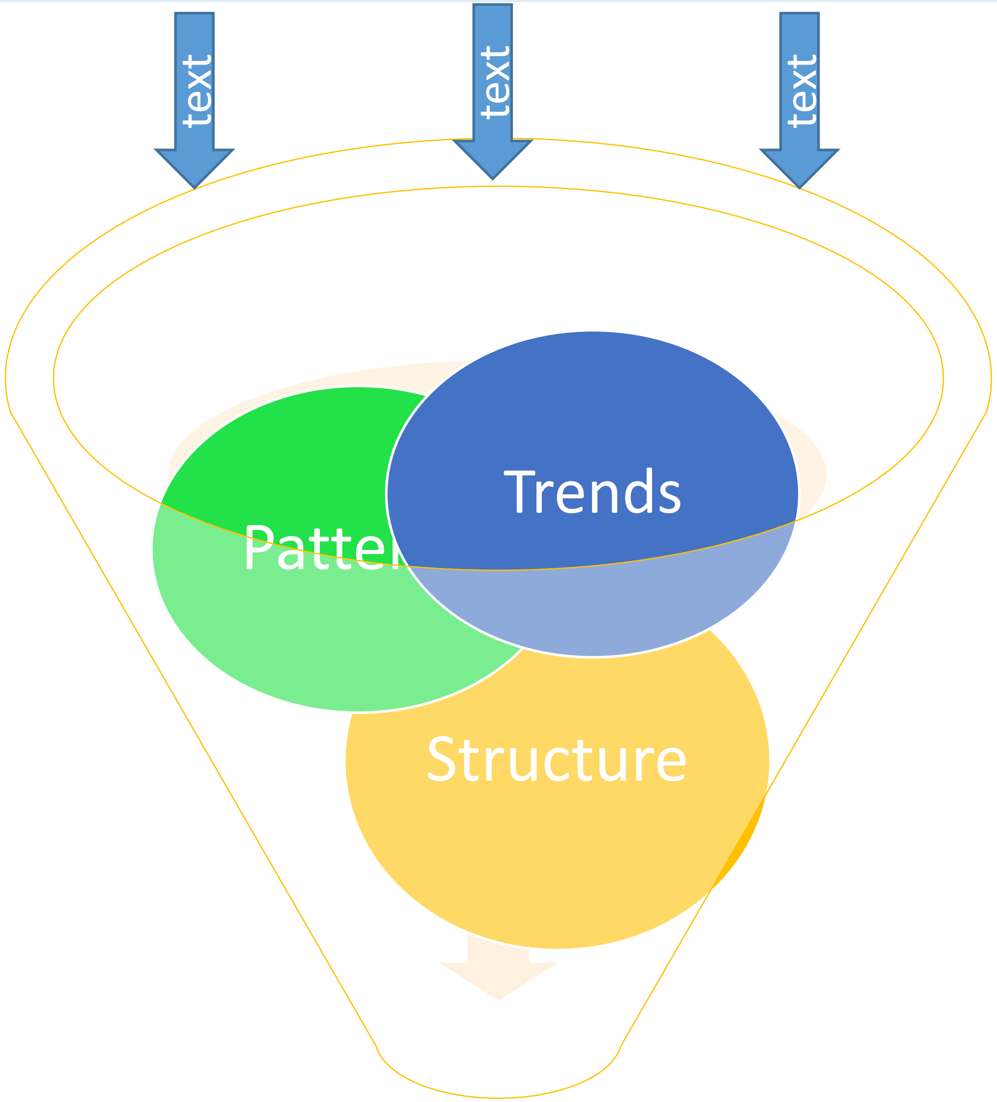
Various Libraries

- NLTK (Natural language toolkit)
- re
- StanfordNLP
- FastText (An NLP library by Facebook)
- ML Libraries
- DL Libraries
- sklearn

Text mining/analytics

https://en.wikipedia.org/wiki/Text_mining

- Text mining is the process of deriving high-quality information from text.
- High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning.
- Text mining usually involves the process of structuring the input text, deriving patterns within the structured data, and finally evaluation and interpretation of the output.
- 'High quality' in text mining usually refers to some combination of relevance, novelty, and interest.



High quality (relevance, novelty and interest)

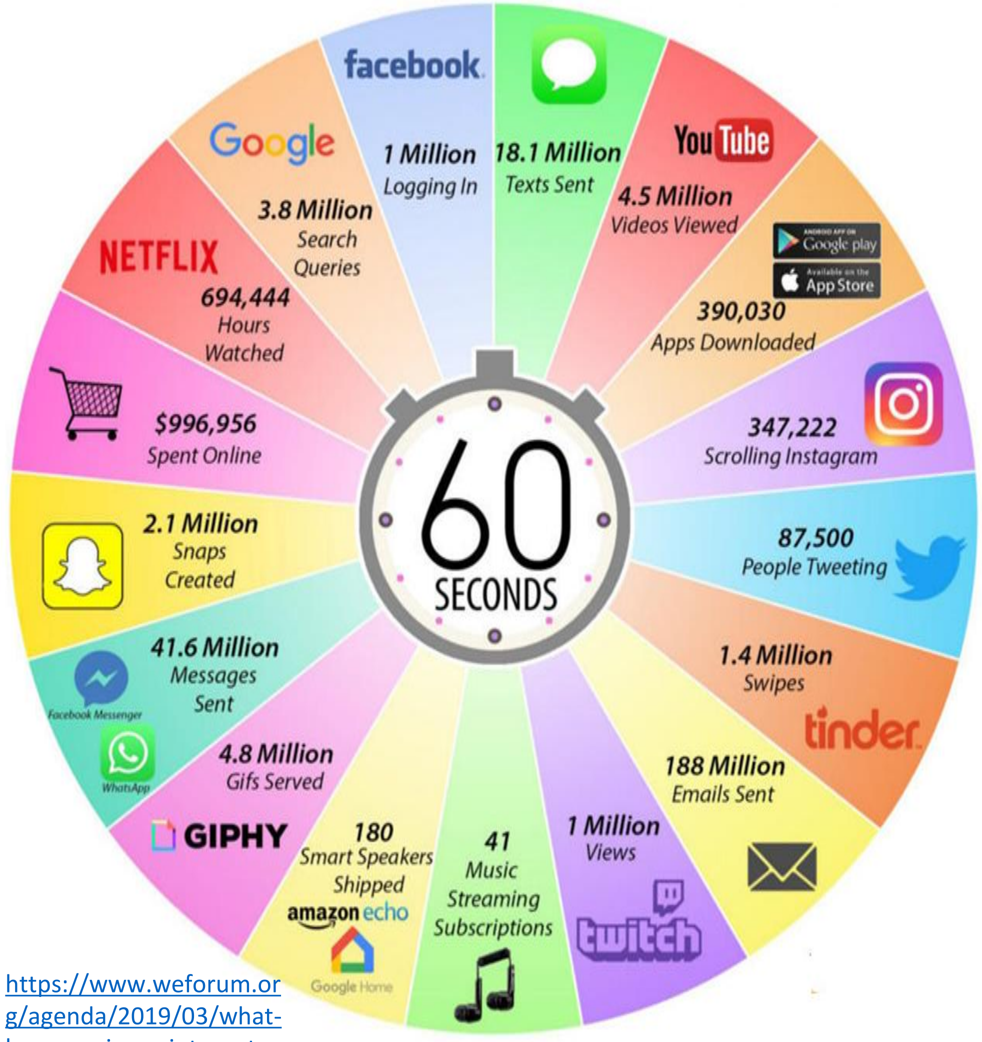
What Is Natural Language Processing

- It is broadly defined as the automatic manipulation of natural language, like speech and text, by software
- **Natural language** refers to the way we, humans, communicate with each other. Namely, speech and text.

- ***Challenge:*** Understanding NL require large amounts of knowledge about morphology, syntax, semantics and pragmatics as well as general knowledge about the world. Acquiring and encoding all of this knowledge is one of the fundamental impediments to developing effective and robust language systems. It uses machine learning methods with the promise of automatic the acquisition of this knowledge from annotated or unannotated language corpora.

Why NLP is emerging

DATA



https://www.weforum.org/agenda/2019/03/what-happens-in-an-internet-minute-in-2019

+

ALGORITHMS

- Logistic regression — 1958
- Hidden Markov Model — 1960
- Stochastic gradient descent — 1960
- Support Vector Machine — 1963
- k-nearest neighbors — 1967
- Artificial Neural Networks — 1975
- Expectation Maximization — 1977
- Decision tree — 1986
- Q-learning — 1989
- Random forest — 1995
- Deep Learning - 2006

+

INFRASTRUCTURE

=

Unstructured data is the information that doesn't reside in a traditional relational database

Areas in NLP where greatest success have been seen

Examples	Text Classification (Sentiment, Spam, Language identification, genre of a fictional story)	Speech Recognition
	Language Modeling (spelling, handwriting recognition, statistical machine translation, article headlines)	Caption Generation
	Machine Translation (text in one language to another text/audio)	Document Summarization (heading, abstract)
	Word Representation and Meaning	Question Answering

Applications of NLP

Basic	Intermediate	Advanced
String operation	Noun Phrase extraction	Topic modeling
Stop words	Text similarity	Text classification
Tokenization	Phonetic matching	Sentiment analysis
Stemming	Information extraction	Word sense disambiguation
Lemmantisation	NER – Entity Recognition	Speech recognition and speech to text
Spelling correction	Clustering Documents	Text to speech
Parts of speech tagging	Converting Text to Features	Next Word Prediction
	Language detection and translation	Summarization
	Computational linguistics	Q & A System

Basic

Hands on with ‘1.1.nlp_basic_string_operations.py’

Hands on with ‘1.2.nlp_basic_regex.py’

Hands on with ‘1.3.nlp_basic_nltk_test.py’

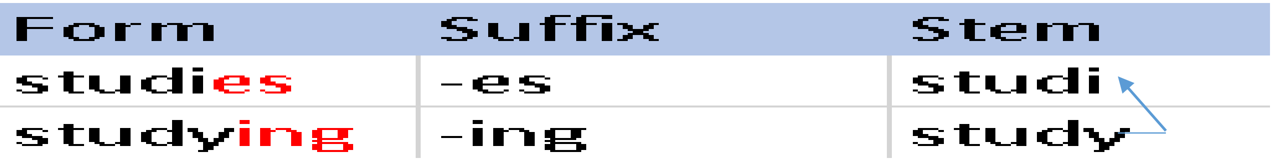
Hands on with ‘1.4.nlp_basic_nltk_operations.py’

Challenges of sentence tokenization

- Sentence tokenization varies from language to language.
- If there is small letter after a dot, then the sentence should not split after the dot.
  - He has completed his Ph.D. degree. He is happy.
  - The sentence tokenizer should split the sentence after degree, not after Ph.D.
- If there is a small letter after the dot, then the sentence should be split after the dot.
  - This is an apple.an apple is good for health.
  - The sentence tokenizer should split the sentence after apple.
- If there is an initial name in the sentence, then the sentence should not split after the initials:
  - Harry Potter was written by J.K. Rowling. It is an entertaining one.
  - The sentence should not split after J. It should ideally split after Rowling.

Stemming and Lemmatization

Stemming	Lemmatization
A process that chops off the ends of words to get stem	*Try* to use vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the *lemma* . It also depends on context (verb or a noun).
Process of converting word to its root form by deleting or replacing suffixes	Remove the inflection endings and convert the word into its base form like in dictionary
Stemming usually operates on single word without knowledge of the context	Lemmatization usually considers words and the context of the word in the sentence
In stemming, we do not consider POS Tags	In lemmatization, we consider POS tags
Stemming is used to group words with a similar basic meaning together	Lemmatization concept is used to make dictionary or WordNet kind of dictionary.



Form	Suffix	Stem
studies	-es	studi
studying	-ing	study

Form	Morphological information	Lemma
studies	Third person, singular number, present tense of the verb study	study
studying	Gerund of the verb study	study

Word-sense disambiguation (WSD)

- When a single word has multiple meaning

- For the machine, it is difficult to identify the correct meaning

- To solve this challenging issue we can use the RB (Computational Linguistics will be discussed later) or ML/DL.

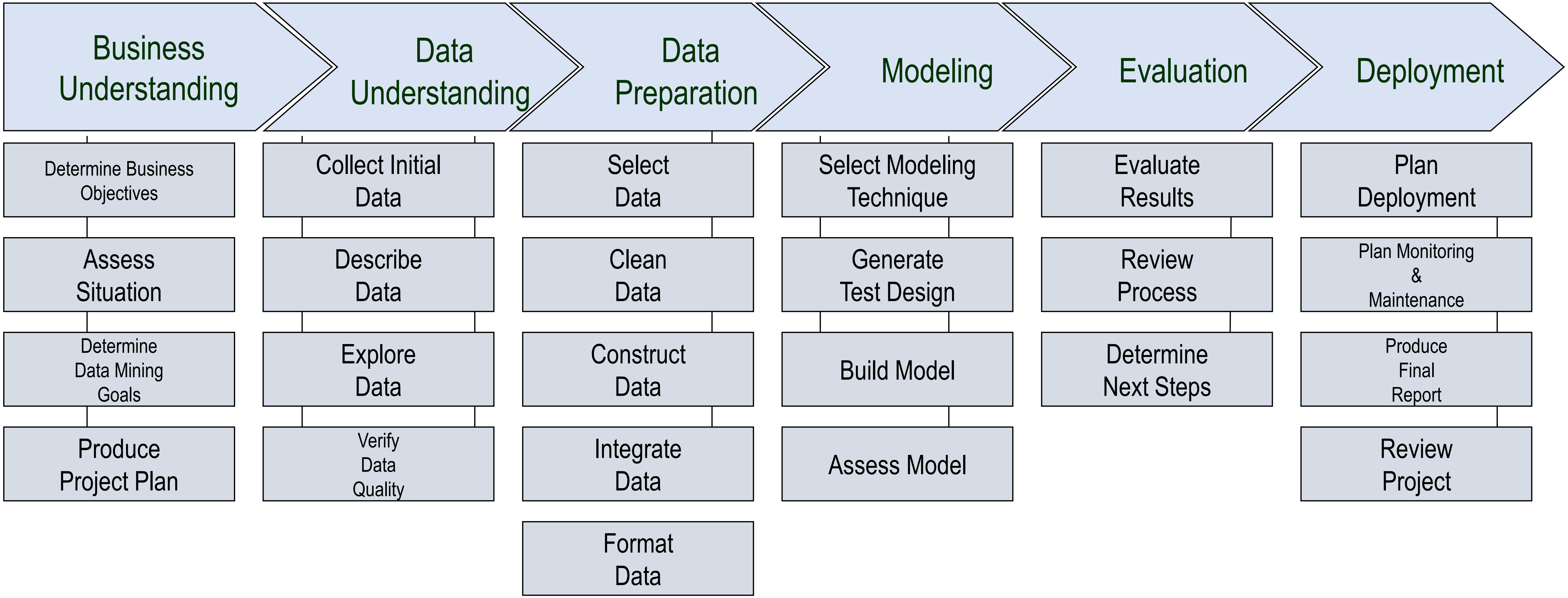
Hands on with ‘1.5.nlp_basic_string_cleaning.py’

Hands on with ‘1.6.nlp_basic_wordcloud.py’

Intermediate

Overall approach for NLP solutions

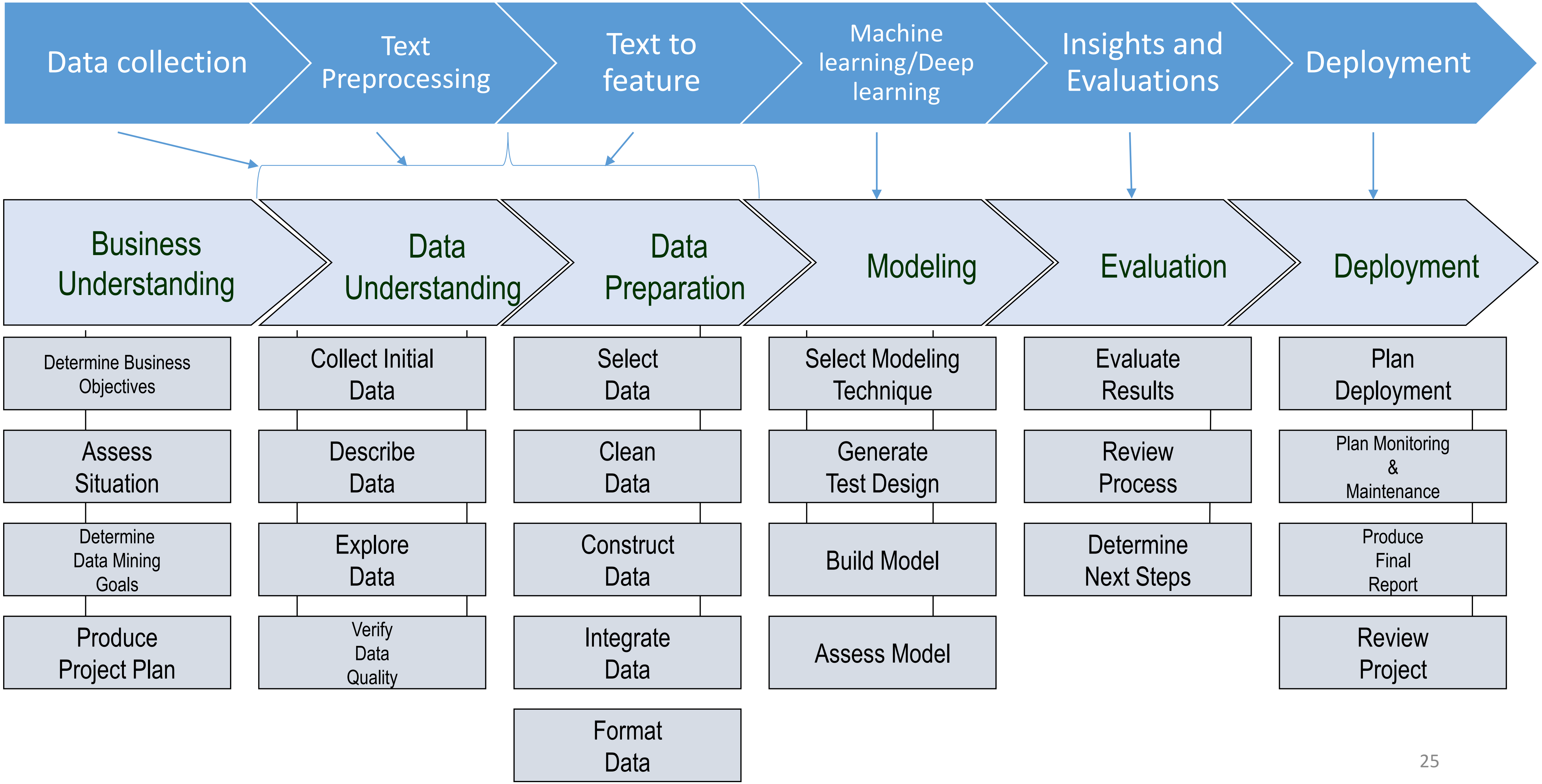
Analytics Methodology CRISP DM: Phases and Tasks



Business Understanding	Data Understanding	Data Preparation	Modeling	Evaluation	Deployment
Determine Business Objectives	Collect Initial Data	Select Data	Select Modeling Technique	Evaluate Results	Plan Deployment
Assess Situation	Describe Data	Clean Data	Generate Test Design	Review Process	Plan Monitoring & Maintenance
Determine Data Mining Goals	Explore Data	Construct Data	Build Model	Determine Next Steps	Produce Final Report
Produce Project Plan	Verify Data Quality	Integrate Data	Assess Model		Review Project
		Format Data			

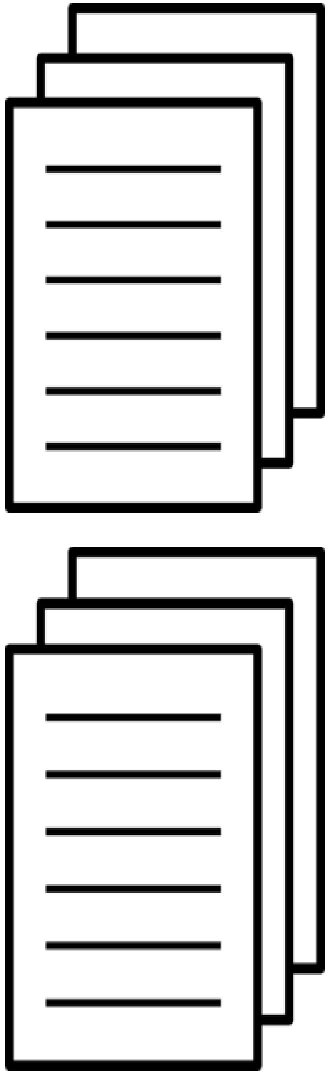
Cross Industry standard process

Overall approach for NLP



Text Analytics : Text analysis steps (Unstructured to Structure conversion)

Documents → Cleaning → Spelling Correction → Normalisation → Text to Features → Advance Analytics



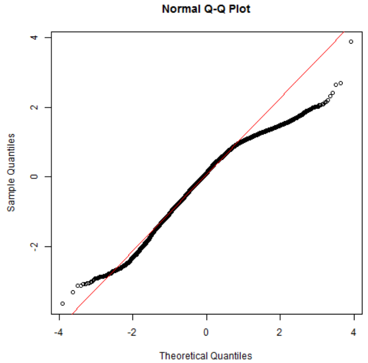
- Remove Whitespace
- Remove Punctuation
- Remove Stop Words
- Remove custom patterns
- Substitute words

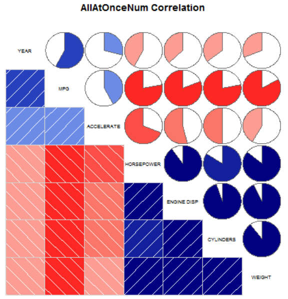
many more...

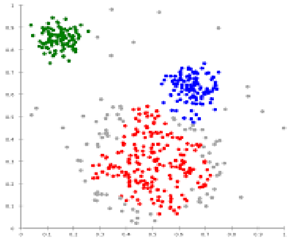
Stemming

Lemmantisation

2-3 Iterations

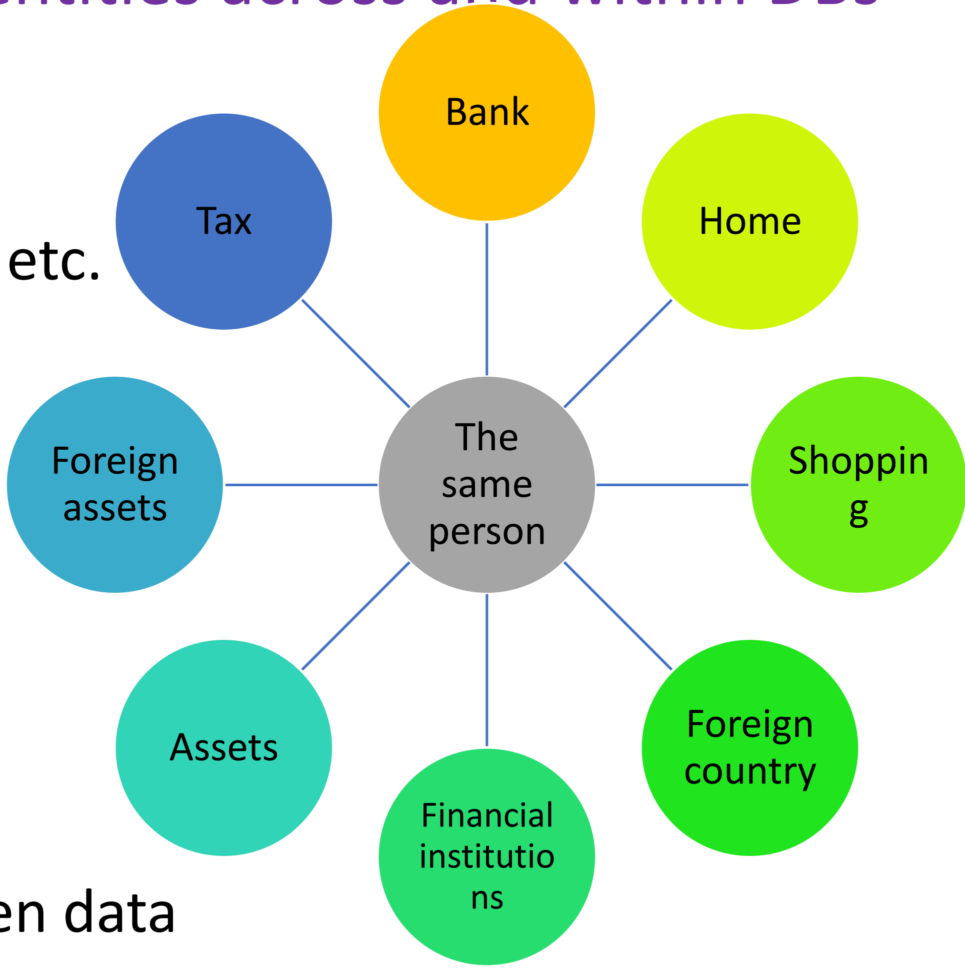






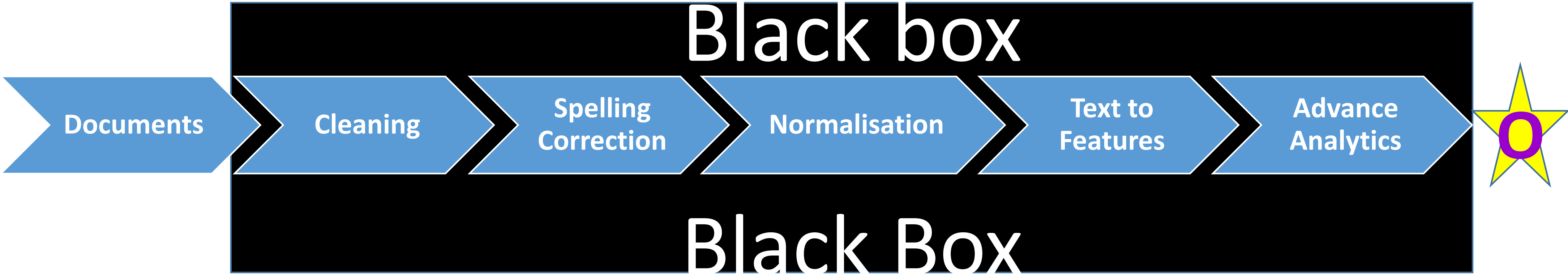
Entity resolution /Deduplication /Text Similarity

- Identification of records that correspond to same entities across and within DBs
- Definition:
  - Entity: A unique thing (a person, a business, a product)
  - Attributes: a name, an address, a shape, a title, a price, etc.
- Skill required: Domain expertise
- Resolution steps:
  - Deduplication: Remove duplicate
  - Record linkage: Records that reference the same entity
  - Normalization: Transform into a standard form
- Techniques:
  - User labels the data -> ML to train -> Predict with unseen data
  - User labels the data -> Measure distances (ex: Affine gap distance) -> Cluster minimum distance records

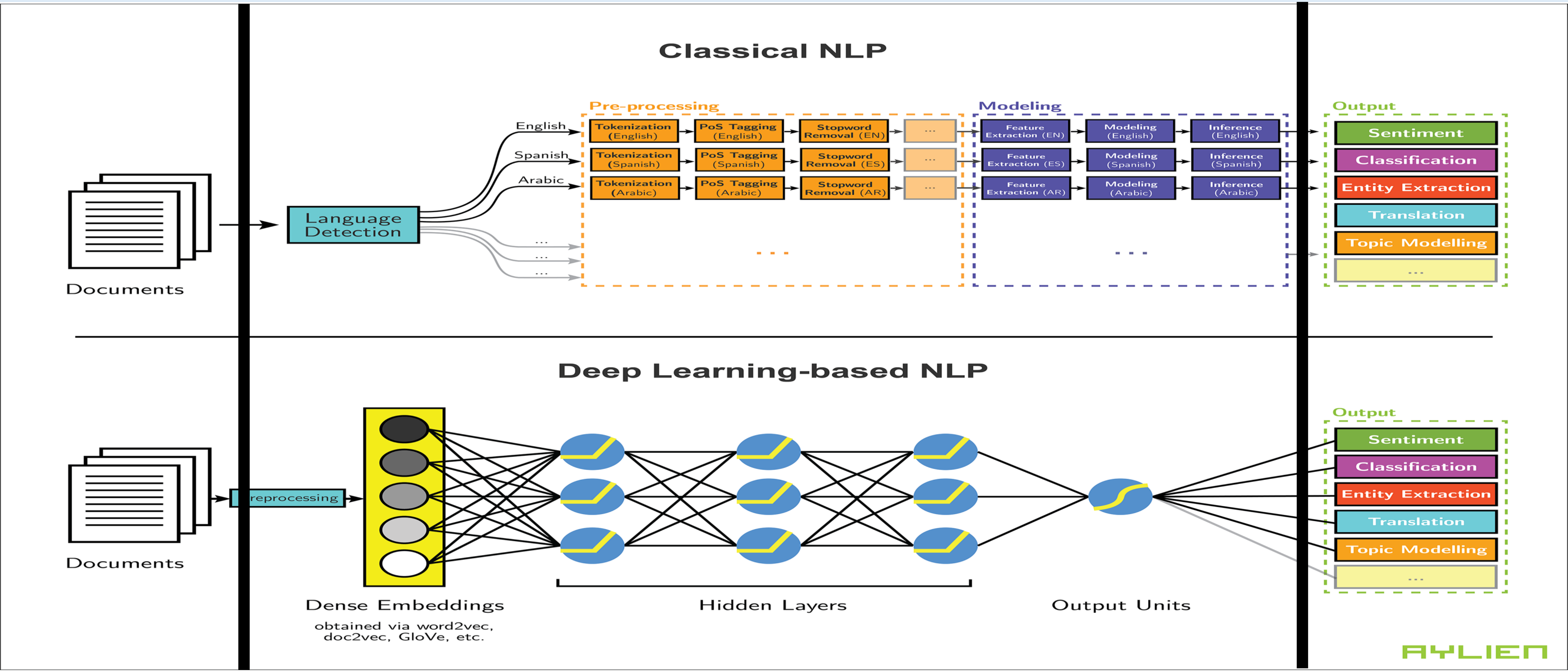


Hands on with ‘2.1.nlp_intermediate_entity_resolution.py’

Text to Features



Text to Features cont



Black box

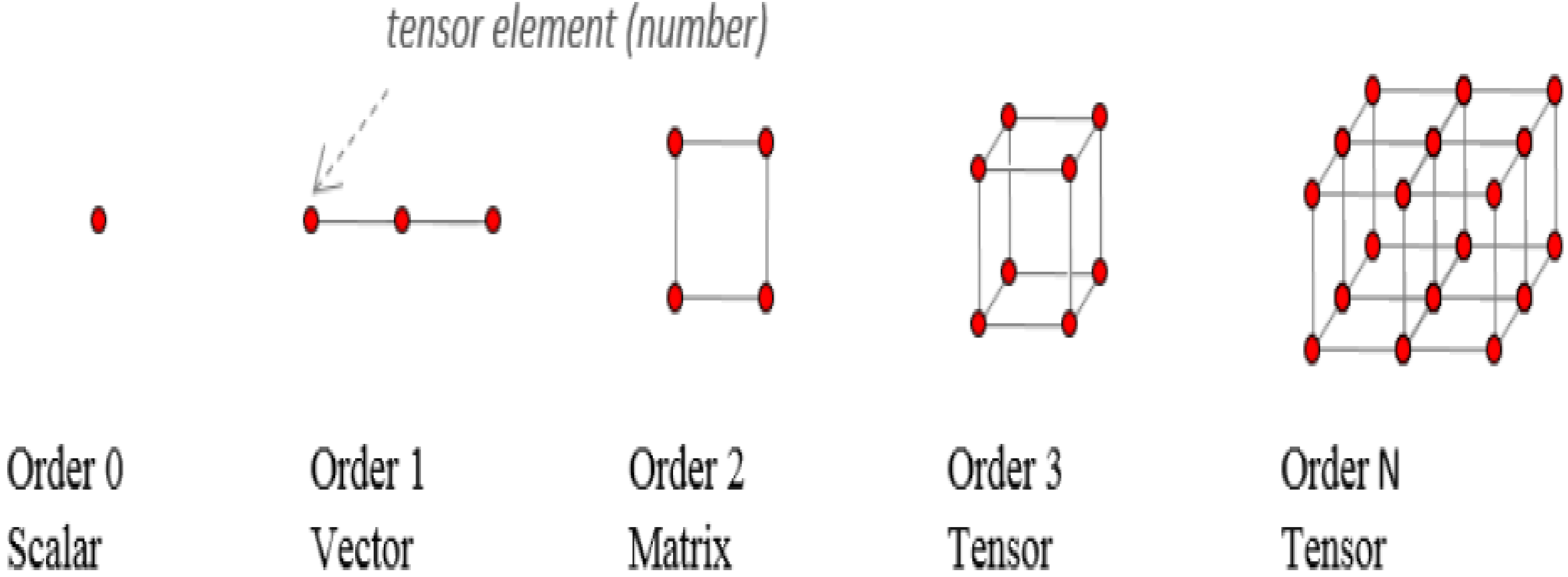
Text to Features - techniques

- Frequency based features
  - One Hot Encoding
  - Count vectorizer
  - TF-IDF
  - ……
- Prediction based features
  - Word embedding

Hands on with ‘2.2.nlp_intermediate_text_to_features.py’

Basic concepts of linear algebra

Tensors



Word Embedding

"You shall know the word by the company it keeps.“ - John Firth

- A type of word representation that allows words with similar meaning to have a similar representation.
- A class of approaches for representing words and documents using a dense vector representation.
- The position of a word within the vector space is learned from text and is based on the words that surround the word when it is used.
- The position of a word in the learned vector space is referred to as its embedding.
- Two popular examples of methods of learning word embedding from text include: Word2Vec and GloVe

Discuss why Word Embedding

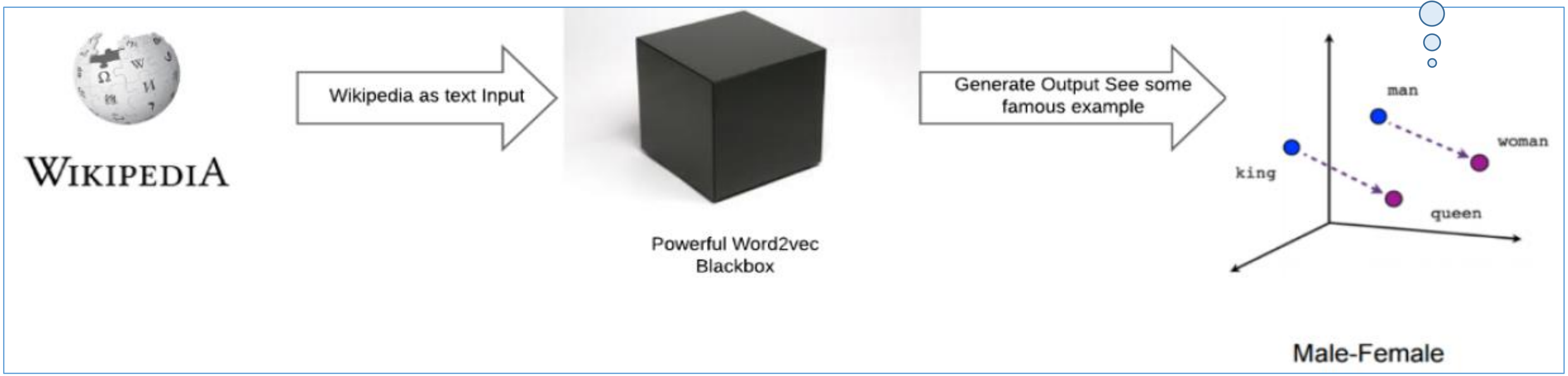
Dense vector formation

	Men	Women	King	Queen	Apple	Orange
Gender	1	-1	1	-1	0	0
Royal	0	0	1	1	0	0
Age	0	0	.7	.6	0	0
Food	0	0	0	0	1	1

Word2Vec (Developed by Tomas Mikolov, et al. at Google in 2013)

- It is a statistical method for efficiently learning a standalone word embedding from a text corpus by capturing semantic similarity.
- It generate the vector for a word that also carries the significance of similarity measure so that the contextual meaning of the word can understood
- Example:
  - (*"King"* - *"man-ness"*) + *"women-ness"* = *"Queen"*
  - 'Paris' - 'France' + 'Italy' *results in a that is very close to* 'Rome'

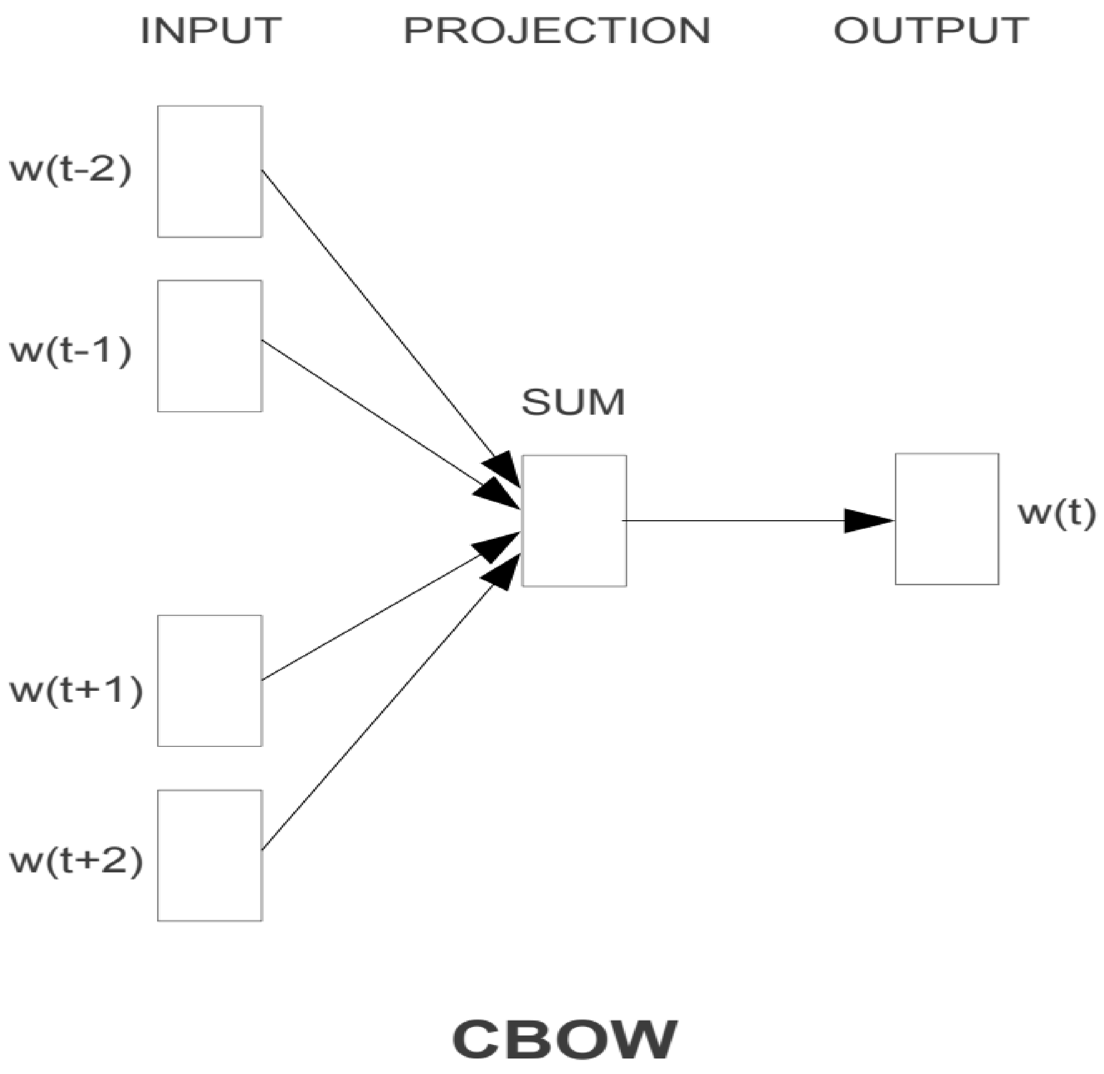
Analogy = *"king is to queen as man is to woman"*.

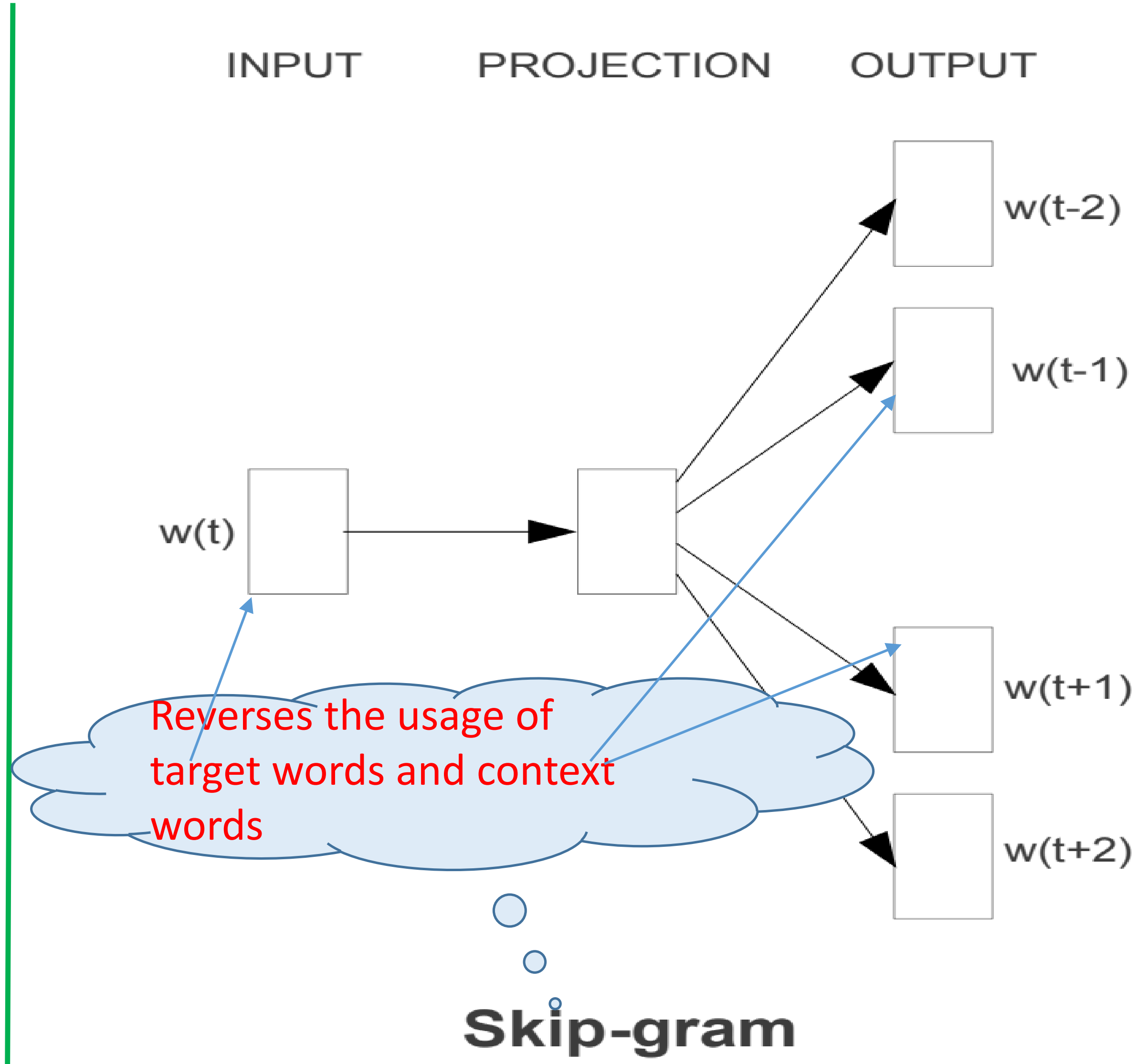

WIKIPEDIA
Wikipedia as text Input
Powerful Word2vec Blackbox
Generate Output See some famous example
man
woman
king
queen
Male-Female

Two different learning models to learn the word embedding : Continuous Bag-of-Words (CBOW) and Continuous Skip-Gram

The CBOW model learns the embedding by predicting the current word based on its context.

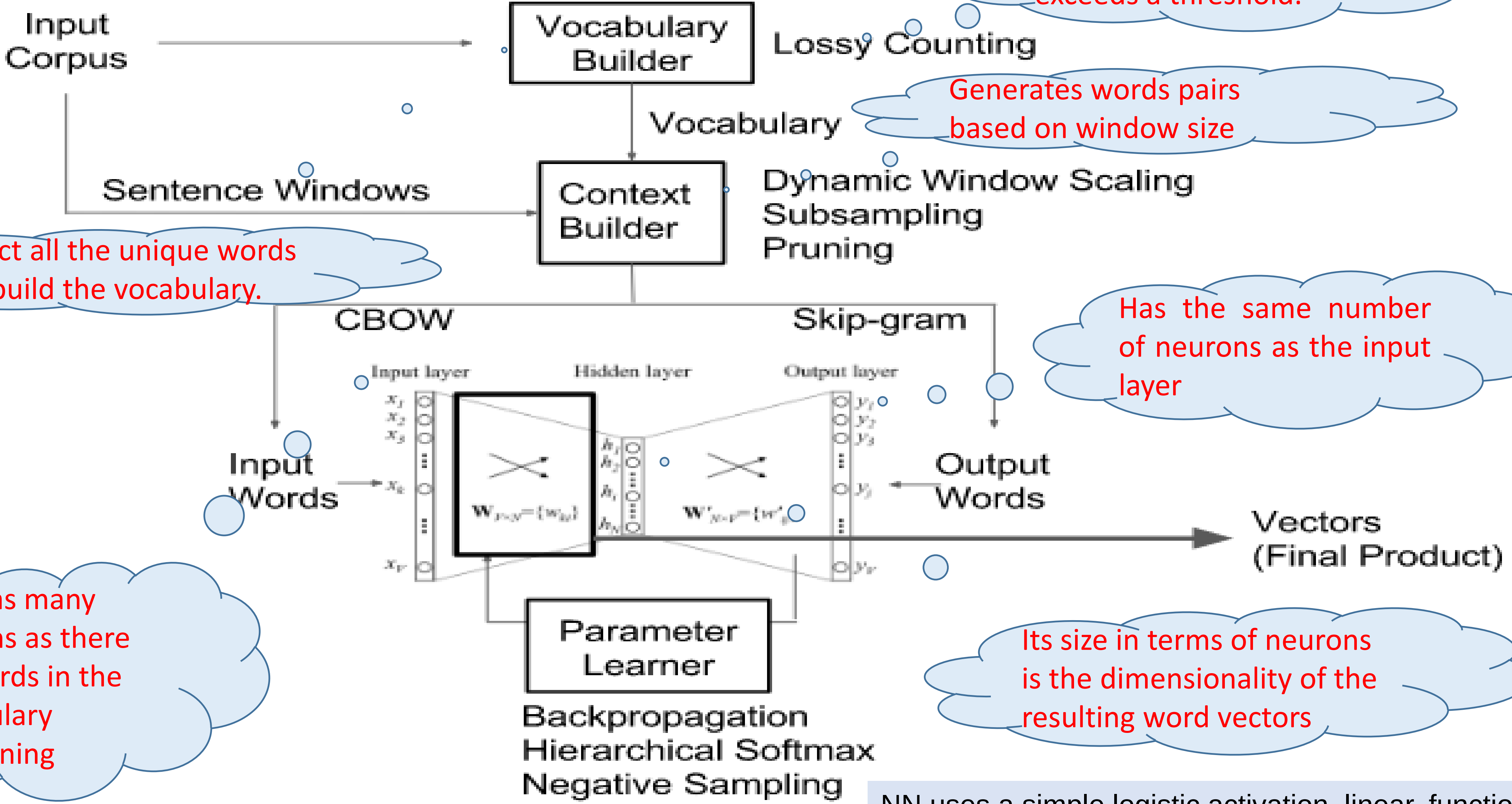
The continuous skip-gram model learns by predicting the surrounding words given a current word.





Reverses the usage of target words and context words

Word2vec Architecture



Count elements in a dataset whose frequency count exceeds a threshold.

Generates words pairs based on window size

Collect all the unique words and build the vocabulary.

Has the same number of neurons as the input layer

It has as many neurons as there are words in the vocabulary for training

Its size in terms of neurons is the dimensionality of the resulting word vectors

NN uses a simple logistic activation, linear, function

Applications of **Word2vec**

- Google uses word2vec and deep learning to improve their machine translation product - Google Machine translation, Google Speech recognition, and Google Vision applications.
- Dependency parser uses word2vec to generate better and accurate dependency relationship between words at the time of parsing.
- Name entity recognition (NER) can also use word2vec, as word2vec is very good at finding out similarity in NERs. All similar entities can come together and you will have better results.
- Sentiment analysis uses it to preserve semantic similarity in order to generate better sentiment results. Semantic similarity helps us to know which kind of phrases or words people use to express their opinions, and you can generate good insights and accuracy by using word2vec concepts in sentiment analysis.
- We can also build an application that predicts a person's profession by using their writing style.
- It helps in document classification with high accuracy and using simple statistics
- Word clustering is the fundamental product of word2vec. All words carrying a similar meaning are clustered together.

GloVe: Global Vectors for Word Representation

- The Global Vectors for Word Representation, or GloVe, algorithm is an extension to the word2vec method for efficiently learning word vectors, developed by Pennington, et al. at **Stanford**.
- GloVe, is a new global log-bilinear regression model for the unsupervised learning of word representations that outperforms other models on word analogy, word similarity, and named entity recognition tasks.
- Rather than using a window to define local context, GloVe constructs an explicit word-context or word co-occurrence matrix using statistics across the whole text corpus. The result is a learning model that may result in generally better word embedding.
- NLP practitioners seem to prefer GloVe at the moment based on results.

Two main difference between tf/ tf-idf and word embedding

- tf / tf-idf -> one number per word
- word embedding -> one vector per word

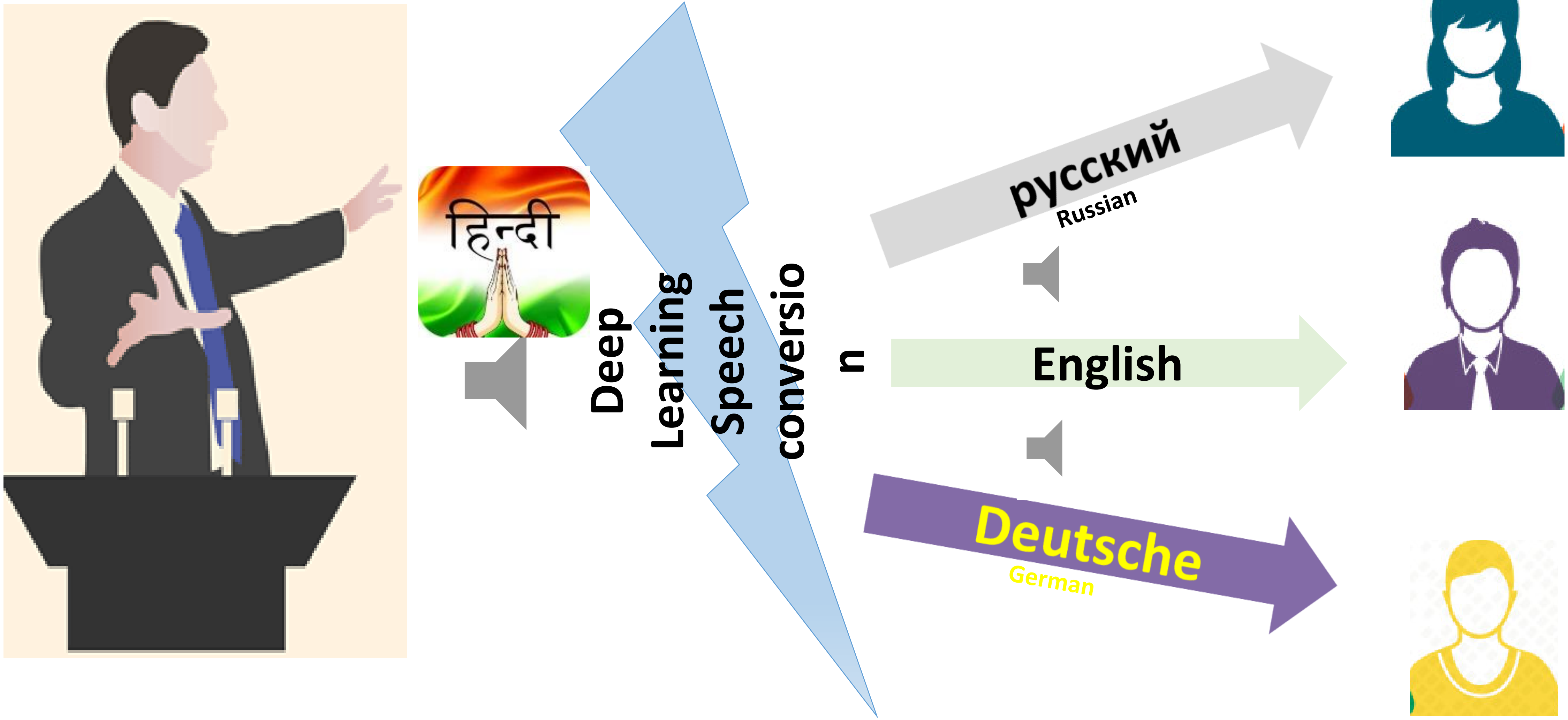
- tf / tf-idf -> useful for classification documents as a whole
- word embedding is good for identifying contextual content.

Hands on with ‘2.3.nlp_intermediate_word_embedding.py’

Hands on with ‘2.4.nlp_intermediate_operations.py’

Speech Recognition/Machine Translation

The use of word embedding over other text representations is one of the key methods that has led to breakthrough performance with deep neural networks on problems like machine translation.



Deep Learning Speech conversio n

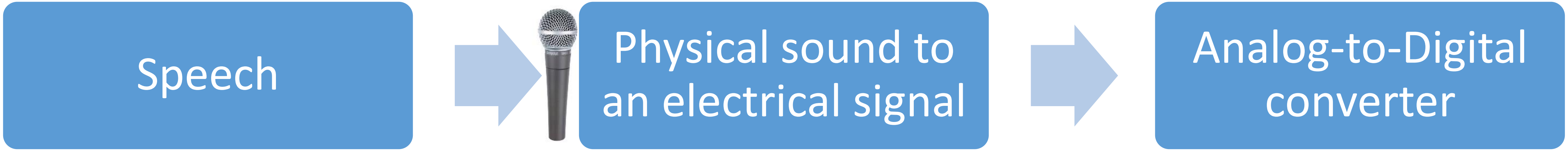
русский
Russian

English

Deutsche
German

Speech Recognition

- Took birth at Bell Labs in the early 1950s
- Early systems were limited to a single speaker and had limited vocabularies of about a dozen words.
- Modern speech recognition systems can recognize speech from multiple speakers and have enormous vocabularies in numerous languages.
- It rely on Hidden Markov Model (HMM) - for short enough timescale (say, ten milliseconds), can be reasonably approximated as a stationary process - statistical properties do not change over time.



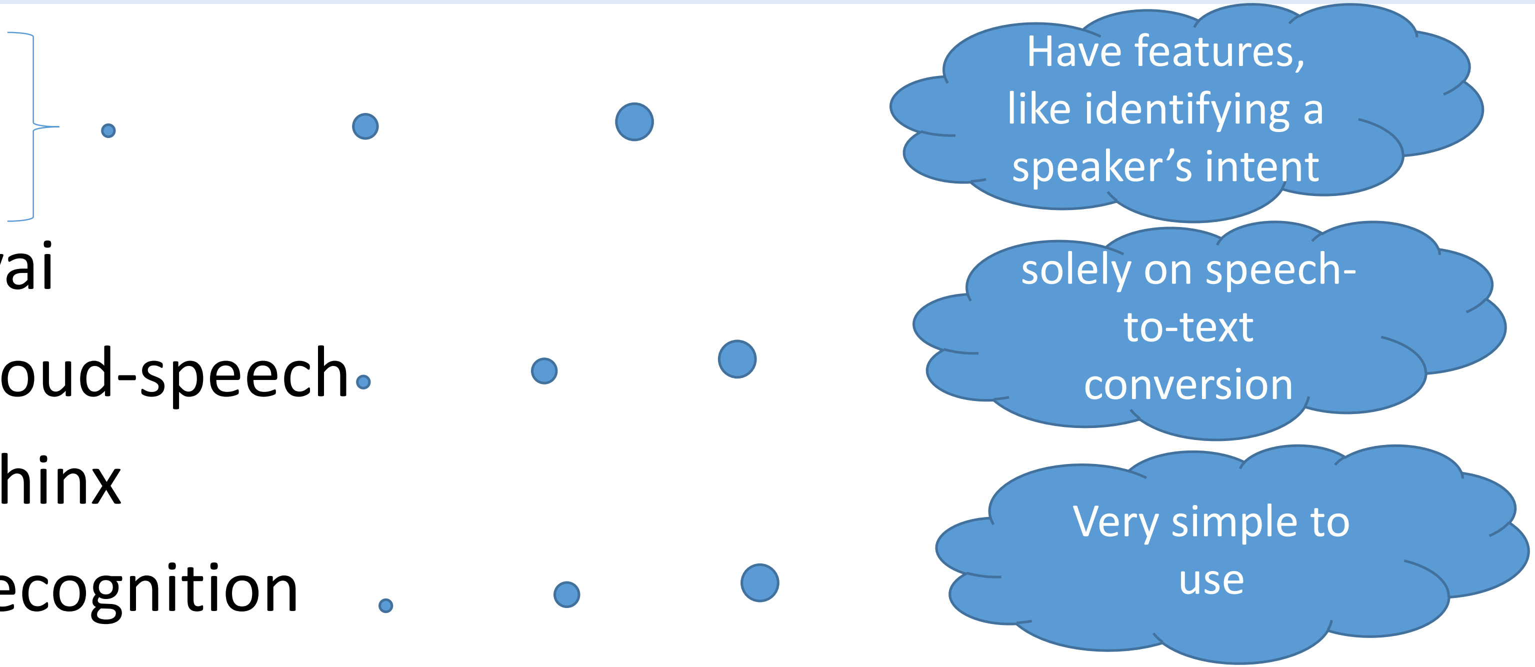
Speech

Physical sound to an electrical signal

Analog-to-Digital converter

Libraries (https://pypi.org/project/SpeechRecognition/)

- Apiai
- wit
- assemblyai
- google-cloud-speech
- pocketsphinx
- SpeechRecognition
- watson-developer-cloud



Have features, like identifying a speaker’s intent

solely on speech-to-text conversion

Very simple to use

Computational Linguistics

It is an interdisciplinary field concerned with the statistical or rule-based modeling of natural language from a computational perspective, as well as the study of appropriate computational approaches to linguistic questions.

https://en.wikipedia.org/wiki/Computational_linguistics

Applications of Computational Linguistics

- Machine Translation
- Text Summarization
- Natural Language Generation
- Natural Language Understanding
- Speech to Text

Types of ambiguity

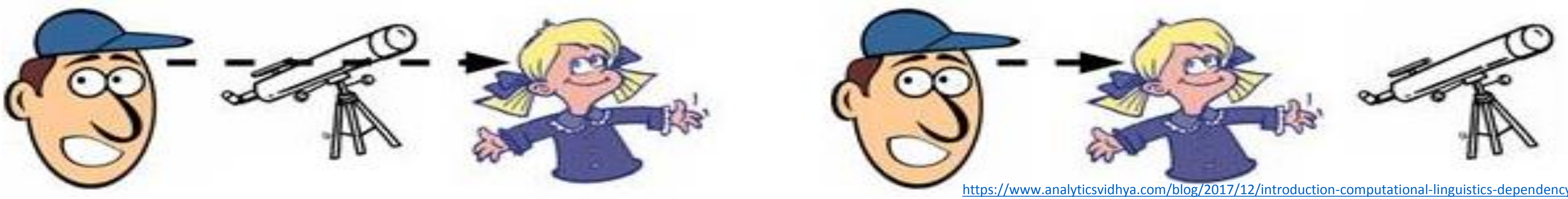
- **Lexical ambiguity** is word-level ambiguity. A single word can have ambiguous meaning in terms of its internal structure and its syntactic class.

  - Look
    - Look at the stars. Here, look is a *verb*.
    - The person gave him a warm look. Here, look is a noun.
  - Silver
    - She won three silver medals. Here, silver is a noun.
    - She made silver speech. Here, silver is a adjective.
    - His stress had silvered his hair. Here, silvered is a verb.

  - By using accurate POS tagger tools, this kind of ambiguity can be resolved
  - WordNet sense has various sense available for a word when the words take specific POS tag. This also helps to handle ambiguity

Types of ambiguity cont

- **Syntactic ambiguity** is for sequences of words that are grammatically structured. There are different ways of interpreting sequences of words, and each structure has a different interpretation. In syntactic ambiguity, syntax is unclear, not the word-level meaning.
  - The man saw the girl with the telescope. Here, the ambiguity is because it is not clear whether the man sees the girl, who has a telescope, or the man sees the girl by using telescope. This ambiguity is called prepositional phrase (PP) ambiguity.



https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/

  - To handle this ambiguity, we need to use statistical approaches and get the most likelihood ratio. We need to take co-occurrences between the verb and the preposition in one hand, and reposition and the noun on other hand, and then calculate the log-likelihood ratio by using following equation:
  - If F(v,p,n) < 0, then we need to attach the preposition to the noun, and if F(v,p,n) >0, then we need to attach preposition to the verb.

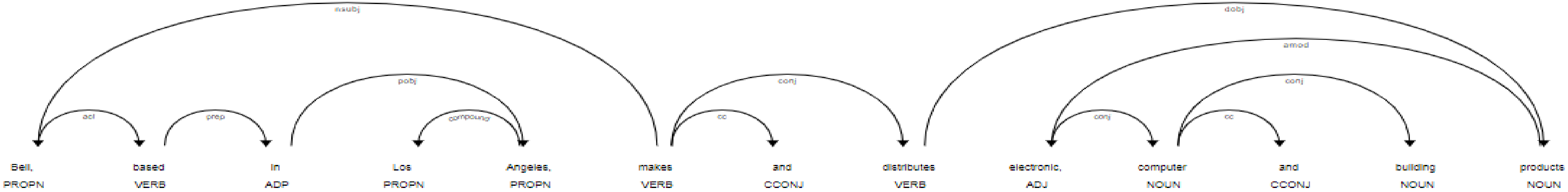
$$F(v,n,p) = log\frac{p(p/v)}{p(p/n)}$$

Types of ambiguity cont

- **Semantic ambiguity**: It occurs when the meaning of the words themselves can be misinterpreted.
- ABC head seeks arms
- Here, the word head either means chief or body part, and in the same way, arms can be interpreted as weapons or as body parts
- Handling semantic ambiguity with high accuracy is an open research area. Nowadays, the word2vec representation technique is very useful for handling semantic ambiguity.
- **Pragmatic ambiguity**: It occurs when the context of a phrase gives it multiple different interpretations.
- Give it to that girl. This could mean any number of things.
- Now let's take a large context: I have chocolate and a packet of biscuits. Give it to that girl. Here, it is not clear whether it refers to chocolate or the packet of biscuits.
- Handling this kind of ambiguity is still an open area of research.

How Computational Linguistics helps

- Structural aspects of the text refer to the organization of tokens in a sentence and the how the contexts among them are interrelated.
- This organization is often depicted by the word-to-word grammar relationships which are also known as dependencies.
- Dependency is the notion that syntactic units (words) are connected to each other by directed links which describe the relationship possessed by the connected words.
- Dependency Extraction: For words in sentence, Computational Linguistics helps to digs further deeper into the relationships and links among them.



Source: https://nlp.stanford.edu/software/dependencies_manual.pdf

Bell, based in Los Angeles, makes and distributes electronic, computer and building products

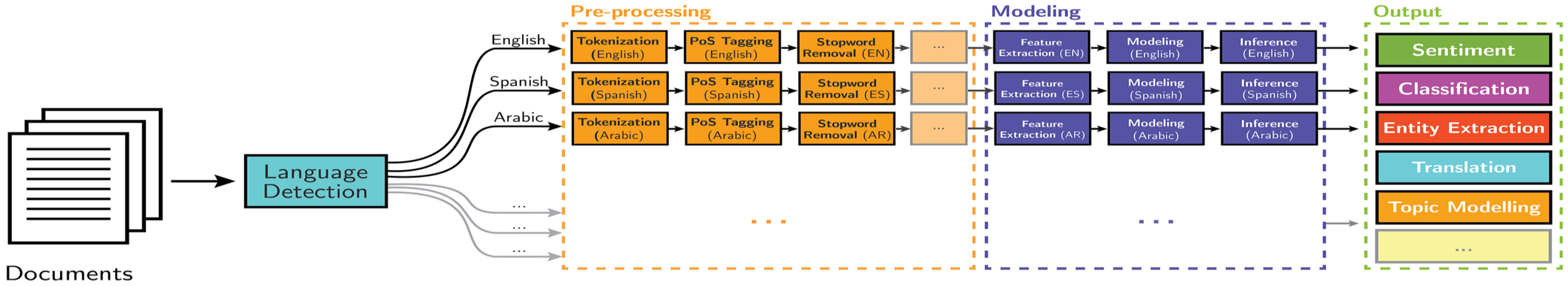
Computational Linguistics: Dependency Extraction

- python libraries : NLTK, Spacy or Stanford-CoreNLP
- Applications of Dependency Trees
  - Named Entity Recognition
    - https://nlp.stanford.edu/pubs/joint-parse-ner.pdf
    - https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14741/14133
  - Question Answering
    - http://cogcomp.org/papers/PunyakanokRoYi04.pdf

Hands on with ‘2.5.nlp_intermediate_computational_linguistics.py’

Advance

Classical NLP



Documents

Language Detection

English

Spanish

Arabic

...

...

...

Pre-processing

Tokenization (English)

PoS Tagging (English)

Stopword Removal (EN)

...

Tokenization (Spanish)

PoS Tagging (Spanish)

Stopword Removal (ES)

...

Tokenization (Arabic)

PoS Tagging (Arabic)

Stopword Removal (AR)

...

...

Modeling

Feature Extraction (EN)

Modeling (English)

Inference (English)

Feature Extraction (ES)

Modeling (Spanish)

Inference (Spanish)

Feature Extraction (AR)

Modeling (Arabic)

Inference (Arabic)

...

Output

Sentiment

Classification

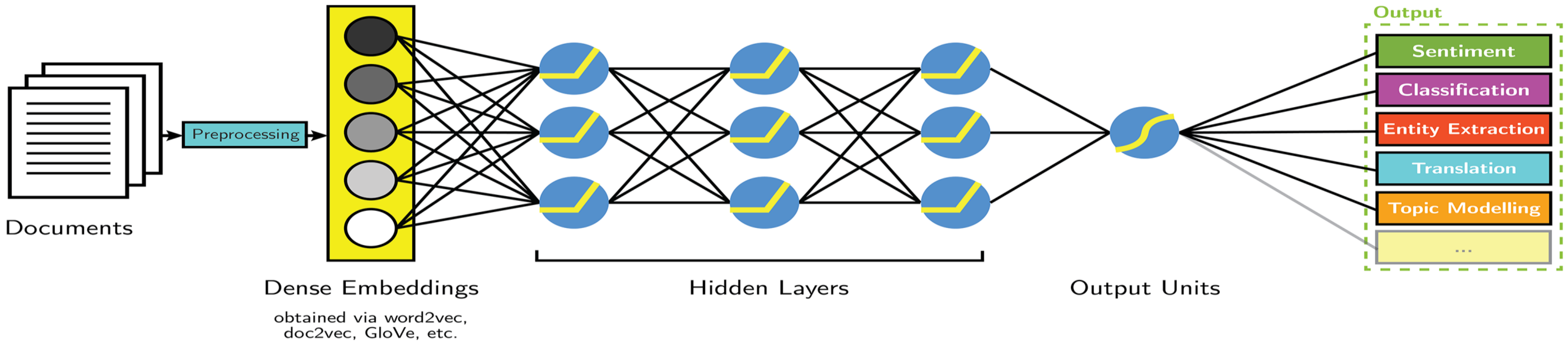
Entity Extraction

Translation

Topic Modelling

...

Deep Learning-based NLP



Documents

Preprocessing

Output

Sentiment

Classification

Entity Extraction

Translation

Topic Modelling

...

Dense Embeddings

obtained via word2vec, doc2vec, GloVe, etc.

Hidden Layers

Output Units

Source: http://blog.aylien.com/leveraging-deep-learning-for-multilingual/

Promise of Deep Learning

1. **The Promise of Drop-in Replacement Models**. The Deep Learning methods can be dropped into existing natural language systems as replacement models that can achieve commensurate or better performance.
2. **The Promise of New NLP Models**. The Deep Learning methods offer the opportunity of new modeling approaches to challenging natural language problems like sequence-to-sequence prediction.
3. **The Promise of Feature Learning**. The Deep Learning methods can learn the features from natural language required by the model, rather than requiring that the features be specified and extracted by an expert.
4. **The Promise of Continued Improvement**. The Deep Learning’s performance in natural language processing is based on real results and that the improvements appear to be continuing and perhaps speeding up.
5. **The Promise of End-to-End Models**. The large end-to-end deep learning models can be fit on natural language problems offering a more general and better-performing approach.

Types of Deep Learning Networks for NLP

- Deep Learning is a large field of study, and not all of it is relevant to natural language processing.
- It is easy to get bogged down in specific optimization methods or extensions to model types intended to lift performance.
- From a high-level, there are few methods from deep learning that deserve the most attention for application in natural language processing. They are:
    - Embedding Layers.
    - Multilayer Perceptions (MLP).
    - Convolutional Neural Networks (CNNs).
    - Recurrent Neural Networks (RNNs).

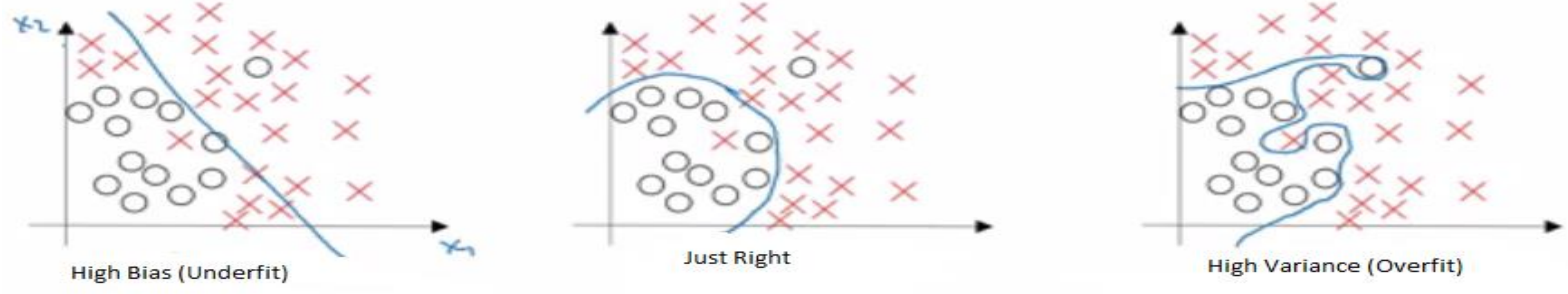
For details, please ask
in DL session

Hands on with ‘3.1.nlp_advance_classifications_ml.py’

Hands on with ‘3.2.nlp_advance_classifications_dl.py’

How to know models are good enough: Bias vs Variance

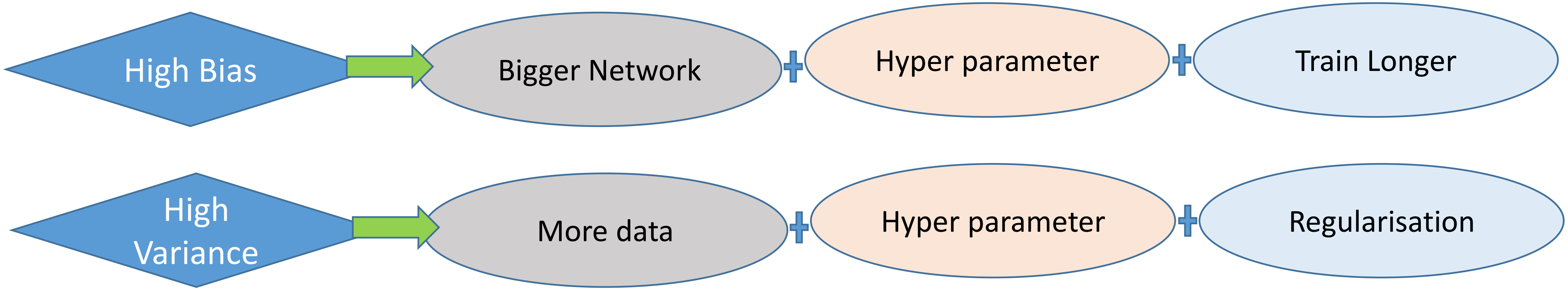
Definition



Error symptom

	High Bias	High Variance	High Bias & High Variance	Low Bias & Low Variance
Train	16%	2%	16%	1%
Dev/Test	17%	12%	30%	1.5%

Solution



High Bias → Bigger Network + Hyper parameter + Train Longer

High Variance → More data + Hyper parameter + Regularisation

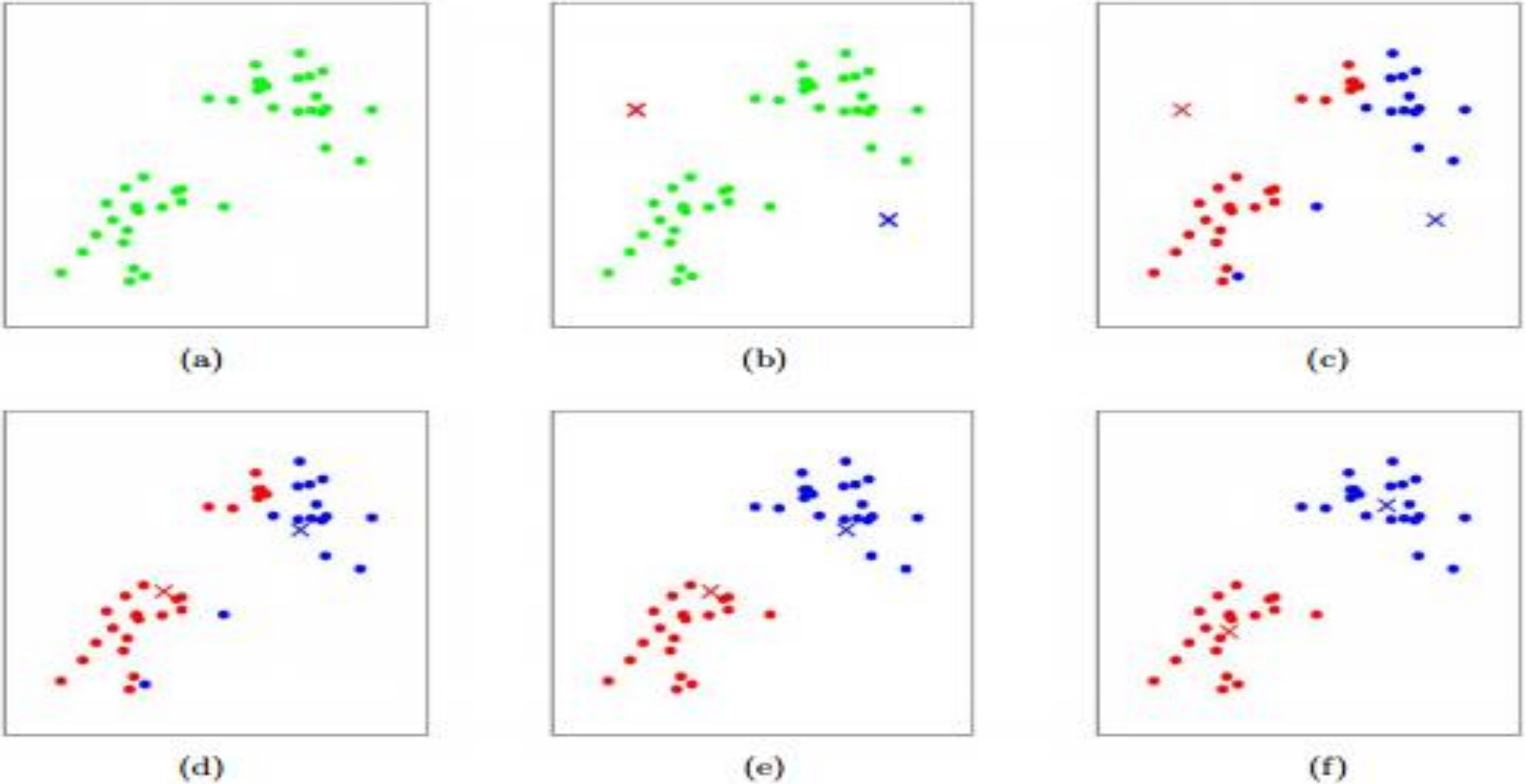
Sentiment analysis

- It is a natural language processing problem where text is understood and the underlying intent is predicted.
- It is all about evaluating the reviews of your customers and categorizing them into positive, negative, and neutral categories

Image link is at reference

Hands on with ‘3.3.nlp_advance_sentiments.py’

K-means clustering



(a)

(b)

(c)

(d)

(e)

(f)

Hands on with ‘3.4.nlp_advance_clustering.py’

Topic modeling using Latent Dirichlet Allocation (LDA)

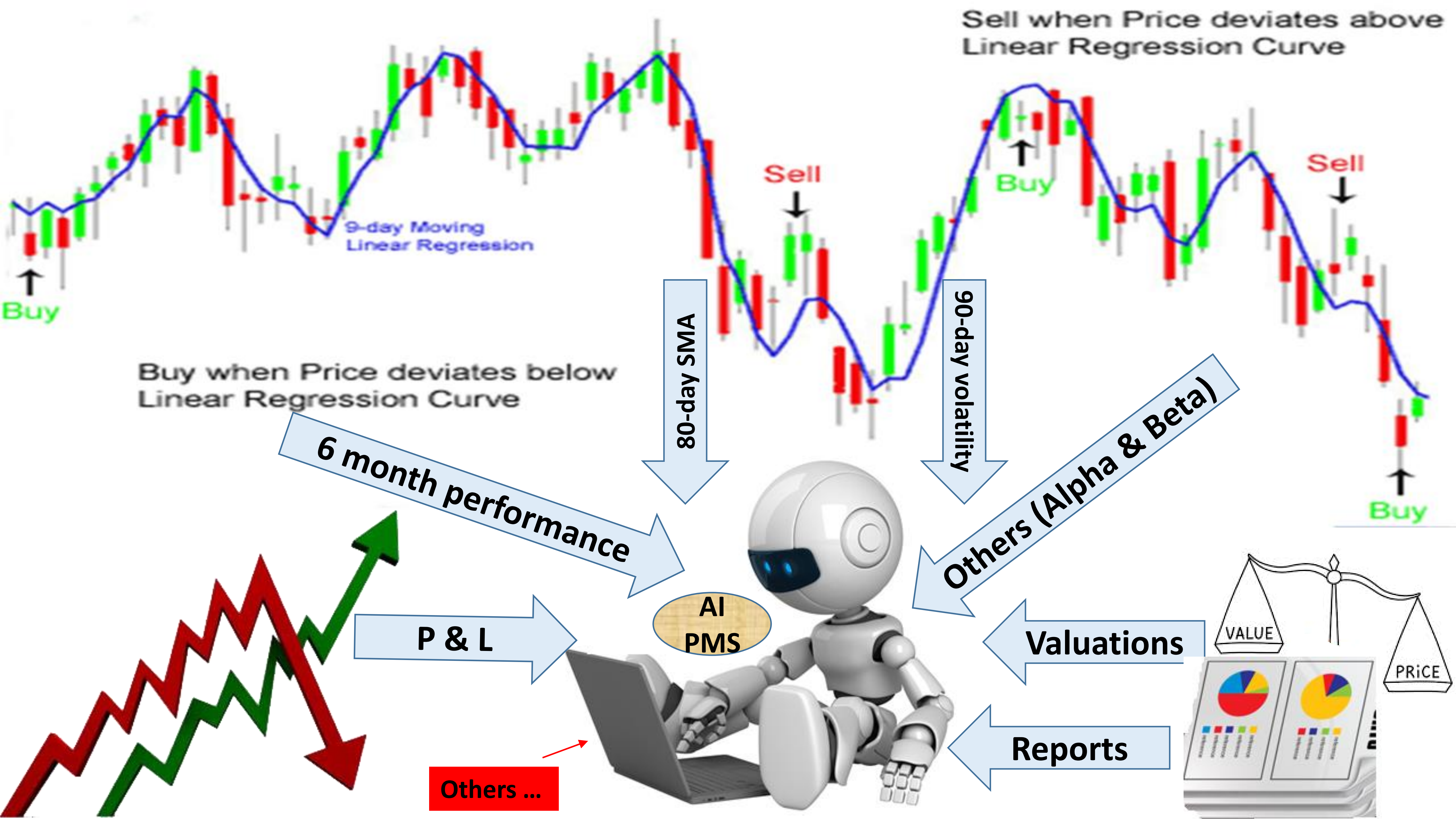
K-Means	Latent Dirichlet Allocation (LDA)
Both are unsupervised techniques. Need the parameter K for the number of clusters and the number of topics	
Hard clustering: partition the N documents in K disjoint clusters	Soft clustering: Document to a mixture of topics. Therefore each document is characterized by one or more topics (e.g. 60% to Topic A, 30% to topic B and 10% to topic E).
	Can give more realistic results than k-means for topic assignment
	Used for Document clustering, Organizing large blocks of textual data, Information retrieval from unstructured text and Feature selection

Hands on with ‘3.5.nlp_advance_topic_modeling.py’

Hands on with ‘3.6.nlp_advance_search_engine.py’

Also known as "Semantic search"

Miscellaneous



6 month performance

80-day SMA

90-day volatility

Others (Alpha & Beta)

AI
PMS

P & L

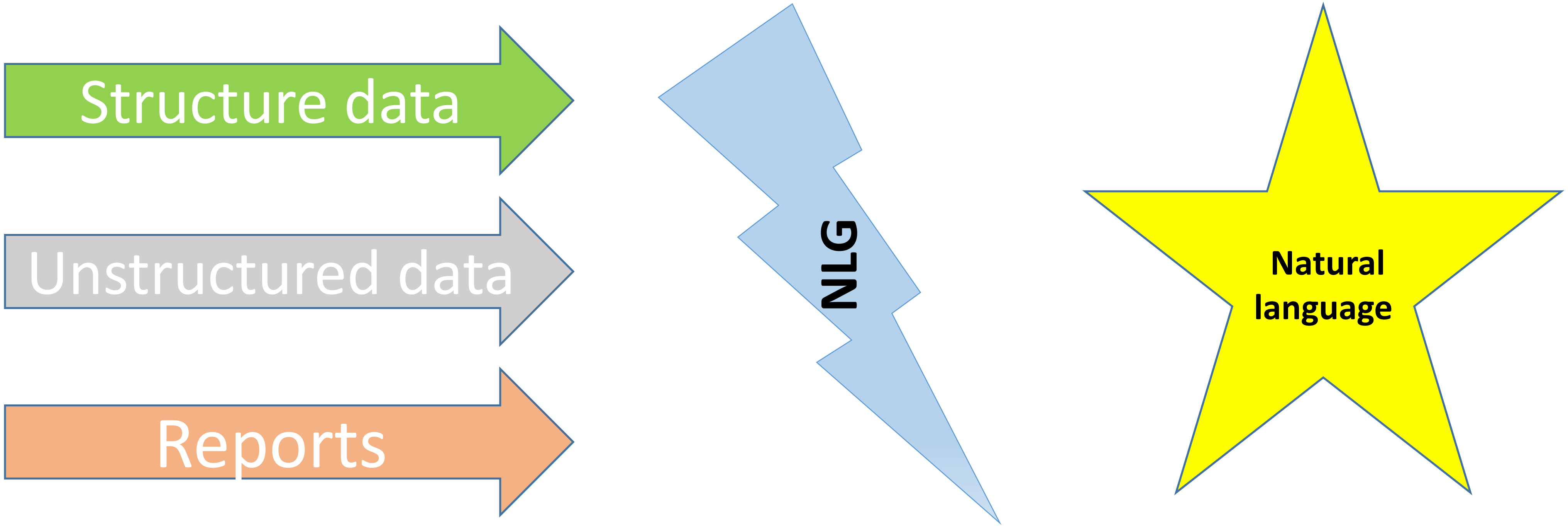
Valuations

Reports

Others …

Few more NLP applications

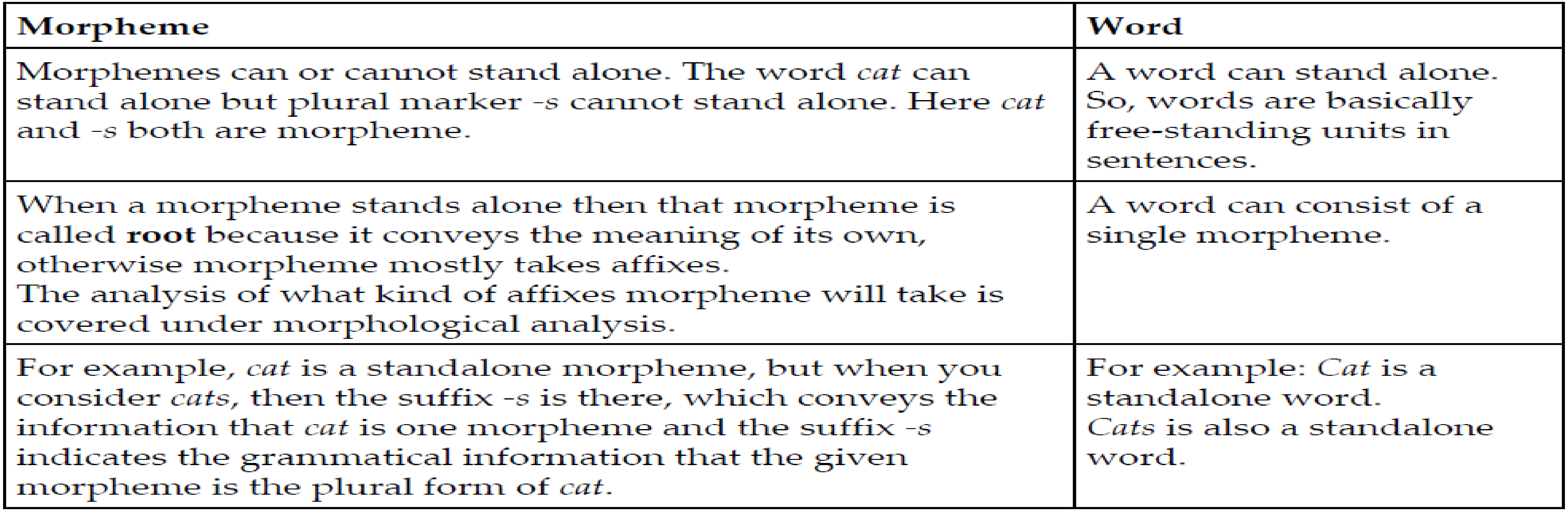
Natural language generation (NLG)



- NLG is considered the second component of NLP.
- NLG is defined as the process of generating NL by a machine as output.
- The output of the machine should be in a logical manner, meaning, whatever NL is generated by the machine should be logical.
- In order to generate logical output, many NLG systems use basic facts or knowledge-based representation.

Morphology

- Morphology is branch of linguistics that studies how words can be structured and formed.
- In linguistics, a morpheme is the smallest meaningful unit of a given language. The important part of morphology is morphemes, which are the basic unit of morphology. Let's take an example. The word ‘boy’ consists of single morpheme whereas ‘boys’ consists of two morphemes; one is boy and the other morpheme -s



Morpheme	**Word**
Morphemes can or cannot stand alone. The word *cat* can stand alone but plural marker *-s* cannot stand alone. Here *cat* and *-s* both are morpheme.	A word can stand alone. So, words are basically free-standing units in sentences.
When a morpheme stands alone then that morpheme is called **root** because it conveys the meaning of its own, otherwise morpheme mostly takes affixes. The analysis of what kind of affixes morpheme will take is covered under morphological analysis.	A word can consist of a single morpheme.
For example, *cat* is a standalone morpheme, but when you consider *cats*, then the suffix *-s* is there, which conveys the information that *cat* is one morpheme and the suffix *-s* indicates the grammatical information that the given morpheme is the plural form of *cat*.	For example: *Cat* is a standalone word. *Cats* is also a standalone word.

Reference

- http://hpe-cct.github.io/programmingGuide/img/diagram1.png
- https://www.packtpub.com/graphics/9781787121423/graphics/B06923_06_07.png
- https://s3.amazonaws.com/aylien-main/misc/blog/images/nlp-language-dependence-small.png
- https://www.analyticsvidhya.com/blog/2017/10/essential-nlp-guide-data-scientists-top-10-nlp-tasks/?utm_source=feedburner&utm_medium=email&utm_campaign=Feed%3A+AnalyticsVidhya+%28Analytics+Vidhya%29
- https://cdn.ttgtmedia.com/visuals/ComputerWeekly/Hero%20Images/Evolution-of-technology-Fotolia.jpg
- Book “Natural Language Processing Recipes_ Unlocking Text Data with Machine Learning and Deep Learning using Python (2019, Apress)” by Akshay Kulkarni, Adarsha Shivananda
- Book “Python Natural Language Processing-Packt Publishing (2017)” by Jalaj Thanaki
- https://docs.python.org/2/library/re.html
- https://www.shoutmeloud.com/seo-stop-words
- http://nltk.org/nltk_data/
- https://github.com/nltk/nltk_data
- https://nlp.stanford.edu/
- http://feedproxy.google.com/~r/AnalyticsVidhya/~3/r-TzzESKAbQ/?utm_source=feedburner&utm_medium=email
- https://recordlinkage.readthedocs.io/en/latest/ref-datasets.html
- https://recordlinkage.readthedocs.io/en/latest/notebooks/link_two_dataframes.html

Reference

- https://stackoverflow.com/questions/52141785/sort-dict-by-values-in-python-3-6
- http://nlp.stanford.edu/data
- https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit?usp=sharing
- https://stackoverflow.com/questions/53815402/value-of-alpha-in-gensim-word-embedding-word2vec-and-fasttext-models
- https://allennlp.org/elmo
- https://www.analyticsvidhya.com/blog/2019/03/learn-to-use-elmo-to-extract-features-from-text/?utm_source=feedburner&utm_medium=email&utm_campaign=Feed%3A+AnalyticsVidhya+%28Analytics+Vidhya%29
- https://pypi.org/project/SpeechRecognition/
- https://cloud.google.com/speech-to-text/docs/languages
- https://realpython.com/python-speech-recognition/
- https://github.com/realpython/python-speech-recognition/blob/master/audio_files/harvard.wav
- http://www.voiptroubleshooter.com/open_speech/american.html
- http://www.voiptroubleshooter.com/open_speech/india.html
- https://en.wikipedia.org/wiki/Levenshtein_distance
- https://en.wikipedia.org/wiki/Hamming_distance
- https://xgboost.readthedocs.io/en/latest/parameter.html
- https://en.wikipedia.org/wiki/Text_mining
- https://www.analyticsvidhya.com/blog/2017/12/introduction-computational-linguistics-dependency-trees/
- https://towardsdatascience.com/a-practitioners-guide-to-natural-language-processing-part-i-processing-understanding-text-9f4abfd13e72
- https://spacy.io/usage/linguistic-features
- http://nlpprogress.com/english/relationship_extraction.html

Few important terms

- **syntactic**. syn·tac·tic. adjective. The **definition** of **syntactic** is relating to the rules of language. An example of something **syntactic** is a sentence that uses the correct form of a verb; **syntactic** sentence.
- **semantic** for English Language Learners. : of or relating to the meanings of words and phrases. : of or relating to **semantics**.
- A large collection of text data is called corpus (corpora in plural)
- **Linguistics** branch focuses on how NL can be analyzed using various scientific techniques. So, the Linguistics branch does scientific analysis of the form, meaning, and context.
- All linguistics analysis can be implemented with the help of computer science techniques. We can use the analysis and feed elements of analysis in a machine learning algorithm to build an NLP application. Here, the machine learning algorithm is a part of **Computer Science**, and the analysis of language is **Linguistics**.
- **Computational linguistics** is a field that helps you to understand both computer science and linguistics approaches together.

Various Applications in the industry

- **n-gram**: In plagiarism tool, use n-gram to extract the patterns that are copied.
- **n-gram**: Computational biology has been using n-grams to identify various DNA patterns in order to recognize any unusual DNA pattern; based on this, biologists decide what kind of genetic disease a person may have
- **TF-IDF**: Text summarization, TF-IDF provides the most important feature for generating a summary for the document.
- **TF-IDF**: Variations of the TF-IDF weighting scheme are often used by search engines to find out the scoring and ranking of a document's relevance for a given user query.
- **TF-IDF**: Document classification use this technique along with BOW.
- **Lossy counting**: Apart from word2vec, the lossy counting algorithm is used in network traffic measurements and analysis of web server logs.

- The part of a word that an affix is attached to is called as **stem**. The word tie is **root** whereas Untie is **stem**.
- **Lexical analysis**: It is defined as the process of breaking down a text into words, phrases, and other meaningful elements. Lexical analysis is based on word-level analysis. In this kind of analysis, we also focus on the meaning of the words, phrases, and other elements, such as symbols.
- **Tokens** are defined as the meaningful elements that are generated by using techniques of lexical analysis.
- **Semantic analysis** focuses on larger chunks and can be performed at the phrase level, sentence level, paragraph level, and sometimes at the document level as well.
- A **part of speech** is a category of words or lexical items that have similar grammatical properties. Words belonging to the same **part of speech (POS)** category have similar behavior within the grammatical structure of sentences. In **English**, POS categories are verb, noun, adjective, adverb, pronoun, preposition, conjunction, interjection, and sometimes numeral, article, or determiner.
- **Markov assumption**: Consider only the last one, two or three words to compute the probability for the upcoming word prediction

P(the | its water is so transparent that) = P(the | that) **or** P(the | transparent that)

- Ontology (onto- from the Greek - "being; that which is", and logy i.e. "logical discourse"): It is the philosophical study of the nature of being, becoming, existence, or reality, as well as the basic categories of being and their relations. Traditionally listed as a part of the major branch of philosophy known as metaphysics, ontology often deals with questions concerning what entities exist or may be said to exist and how such entities may be grouped, related within a hierarchy, and subdivided according to similarities and differences. A very simple definition of ontology is that it is the examination of what is meant, in context, by the word 'thing'.
- Lossy counting: It helps to find which profiles are accessed the most often. With lossy counting, you periodically remove very low count elements from the table. The most-frequently accessed profiles would almost never have low counts anyway, and if they did, they wouldn't be likely to stay there for long. The algorithm basically involves grouping the inputs into blocks or chunks and counting within each chunk. Then you reduce the count for each element by one, dropping any elements whose counts drop to zero. The most-frequently hit profiles will get on your count and stay there. Any profiles that aren't hit very often will drop to zero in a few blocks and you won't have to track them any more. Note that the final results are order-dependent, giving heavier weight to the counts processed last. In some cases, this makes perfect sense and is an upside rather than a downside. (If you want to know basically which profiles are the most popular now, you want to weigh accesses today more than accesses last month.) There are a large number of refinements to the algorithm. But the basic idea is this -- to find the heavy hitters without having to track every element, periodically purge your counts of any elements that don't seem likely to be heavy hitters based on the data so far.

nltk has four types of corpora

- Isolate corpus: This type of corpus is a collection of text or natural language. Examples of this kind of corpus are gutenberg, webtext, and so on.
- Categorized corpus: This type of corpus is a collection of texts that are grouped into different types of categories. An example of this kind of corpus is the brown corpus, which contains data for different categories such as news, hobbies, humor, and so on.
- Overlapping corpus: This type of corpus is a collection of texts that are categorized, but the categories overlap with each other. An example of this kind of corpus is the reuters corpus, which contains data that is categorized, but the defined categories overlap with each other. More explicitly, I want to define the example of the reuters corpus. For example, if you consider different types of coconuts as one category, you can see subcategories of coconut-oil, and you also have cotton oil. So, in the reuters corpus, the various data categories are overlapped.
- Temporal corpus: This type of corpus is a collection of the usages of natural language over a period of time. An example of this kind of corpus is the inaugural address corpus. Suppose you recode the usage of a language in any city of India in 1950. Then you repeat the same activity to see the usage of the language in that particular city in 1980 and then again in 2017. You will have recorded the various data attributes regarding how people used the language and what the changes over a period of time were.