edureka!
a Veranda Enterprise | ILLINOIS TECH

POST GRADUATE PROGRAM IN GENERATIVE AI AND ML

Natural Language Processing

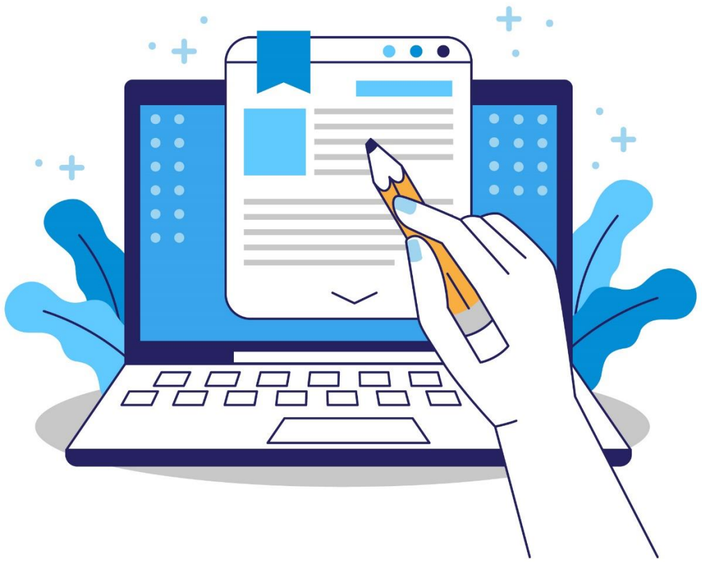
edureka!
a Veranda Enterprise

ILLINOIS TECH

Text Processing and Feature Engineering

Topics

- Stemming Techniques and Applications in NLP
- Lemmatization Methods and Real-World Relevance
- Part-of-Speech Tagging and Its Role in NLP
- Text Representation Techniques: BoW and TF-IDF
- Word Embeddings: Word2Vec, GloVe, and FastText
- Processing Code-Mixed and Multilingual Text
- Text Classification and Feature Selection Methods



Copyright © edureka and/or its affiliates. All rights reserved.

Learning Objectives

By the end of this lesson, you will be able to:

- Describe the principles of stemming, lemmatization, and part-of-speech tagging in NLP workflows.
- Demonstrate the use of traditional text representation techniques like Bag of Words and TF-IDF.
- Evaluate the effectiveness of word embedding models such as Word2Vec, GloVe, and FastText.
- Apply text classification techniques to multilingual data using appropriate feature selection methods.

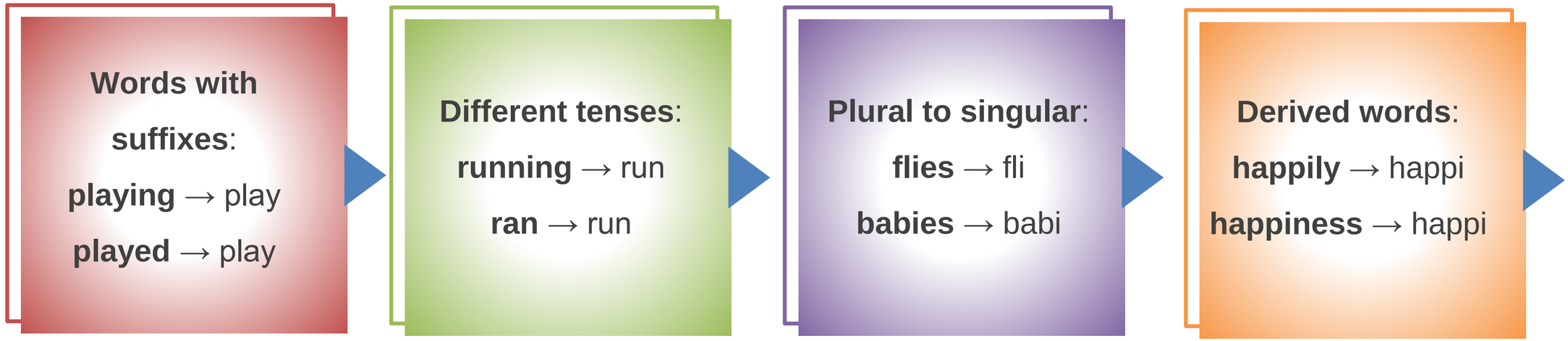
Copyright © edureka and/or its affiliates. All rights reserved.

Stemming

Copyright © edureka and/or its affiliates. All rights reserved.

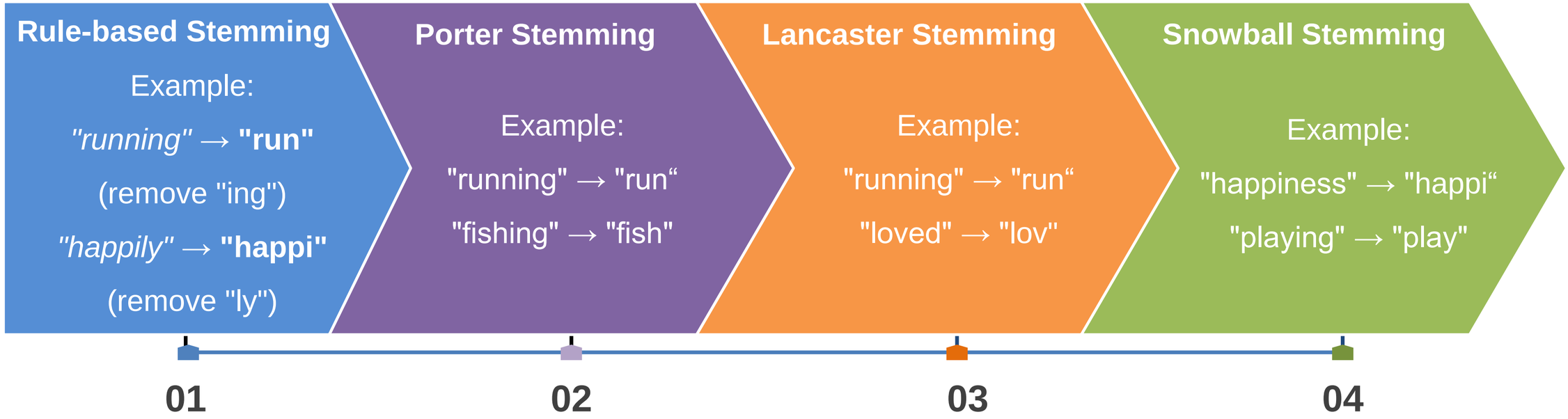
What is Stemming ?

Stemming is a simpler alternative to lemmatization, using handcrafted rules to strip word endings and reduce them to a common base form known as stems.



Copyright © edureka and/or its affiliates. All rights reserved.

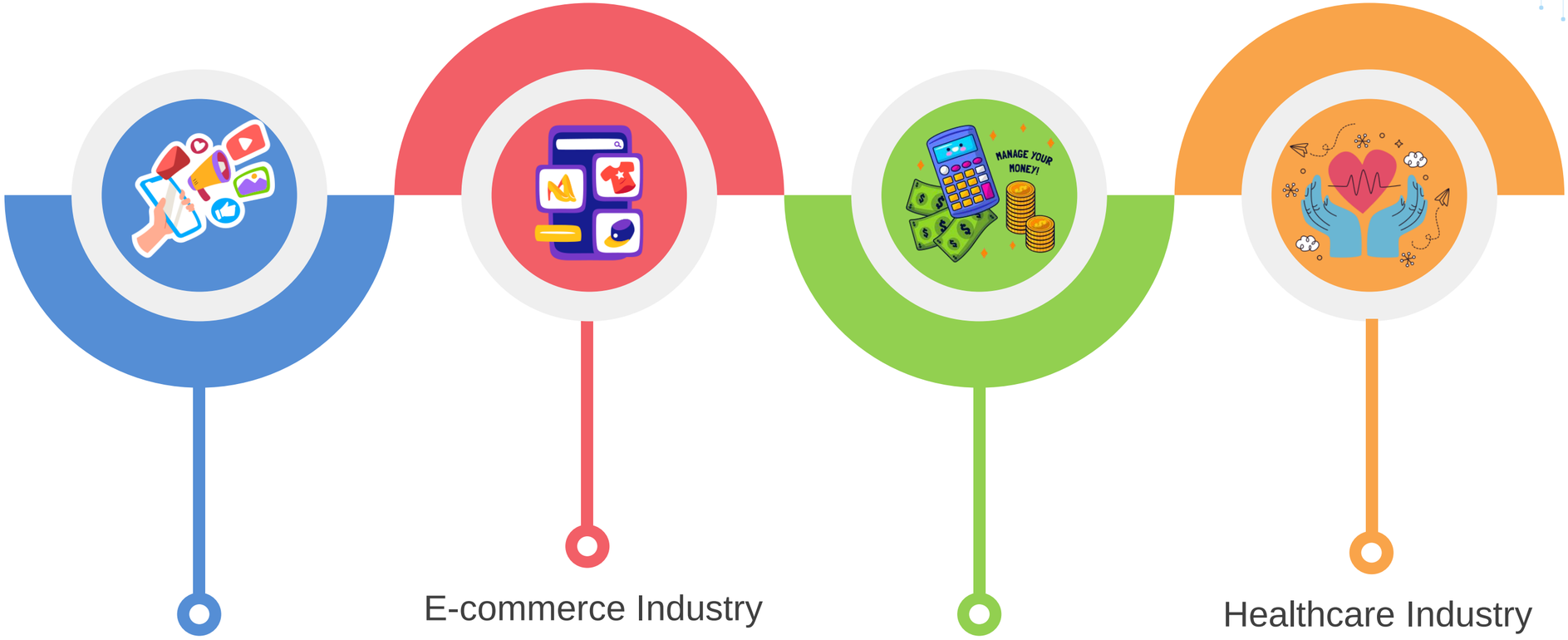
Types of Stemming



edureka! a Veranda ENTERPRISE

Copyright © edureka and/or its affiliates. All rights reserved.

Applications of Stemming



Social Media Analysis

E-commerce Industry

Finance Industry

Healthcare Industry

edureka! a Veranda ENTERPRISE

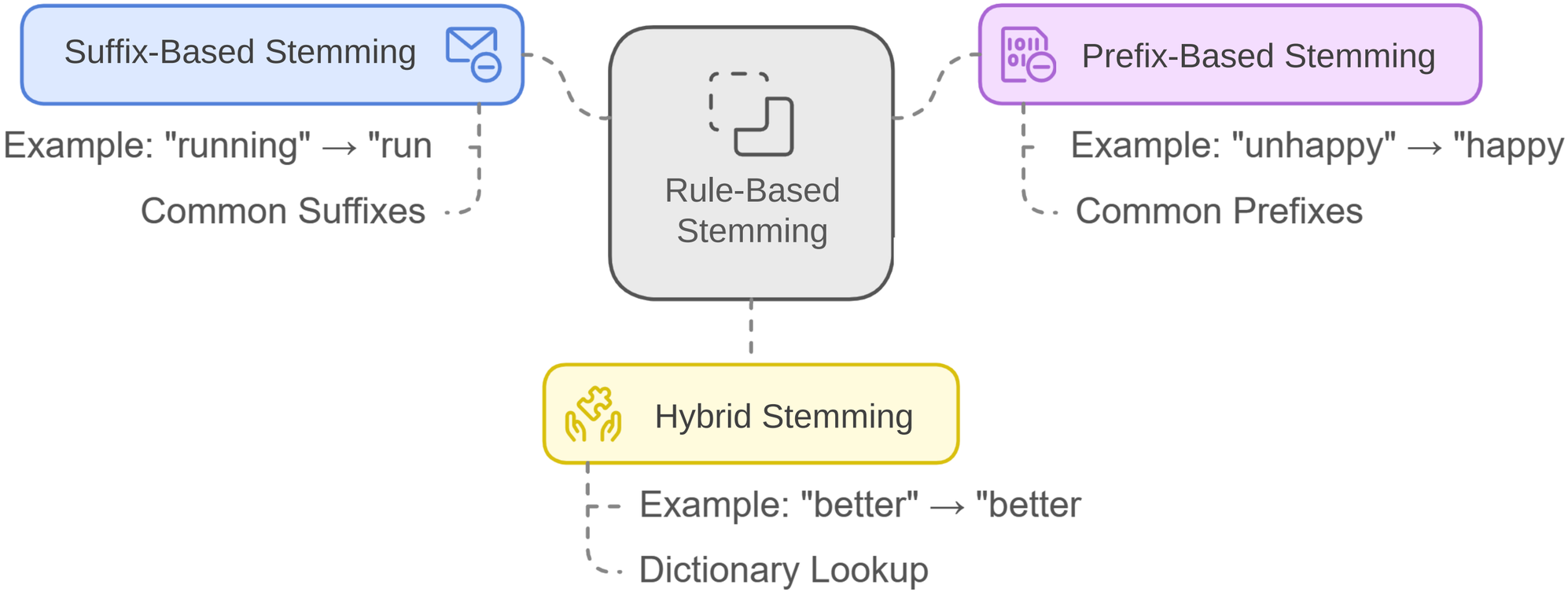
Copyright © edureka and/or its affiliates. All rights reserved.

What is Rule-Based Stemming ?

Rule-based stemming uses **predefined rules** to remove **suffixes or prefixes** from words to find their root (stem). These rules are often language-specific and focus on common affixes like **-ing, -ed, -s,** etc.

Copyright © edureka and/or its affiliates. All rights reserved.

Types of Rule-Based Stemming



edureka! a Veranda ENTERPRISE

Copyright © edureka and/or its affiliates. All rights reserved.

Limitations of Rule-Based Stemming

1. Under-stemming
2. Fixed Rules
3. Language Dependency
4. Limited Vocabulary Expansion
5. Context Ignorance
6. Over-stemming

edureka! a Veranda ENTERPRISE

Copyright © edureka and/or its affiliates. All rights reserved.

Porter Stemmer

- The Porter Stemmer is one of the most widely used stemming algorithms.
- This algorithm applies a set of rules to identify and remove suffixes from words to extract their stems.
- For instance, the Porter algorithm transforms the word "leaping" into "leap" by stripping the "ing" suffix.

Copyright © edureka and/or its affiliates. All rights reserved.

Snowball Stemmer

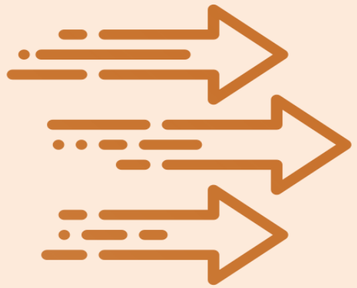
English: "hotels"

French: "*hôtels*"

Spanish: "*hoteles*"

German: "Hotels"

Snowball Stemmer



English: "hotel"

French: "*hôtel*"

Spanish: "*hotel*"

German: "Hotel"

Copyright © edureka and/or its affiliates. All rights reserved.

Why Snowball Stem. over Porter Stem. ?

Aspect	Porter Stemmer	Snowball Stemmer
Aggressiveness	Less aggressive in reducing words.	More aggressive in stemming words.
Fixes	Contains some inconsistencies and issues.	Addresses many issues present in Porter.
Working Differences	Slightly older and simpler rules.	Updated and more refined rules.
Example: 'fairly'	Stemmed to 'fairli'.	Stemmed to 'fair'.
Example: 'sportingly'	Stemmed to 'sportingli'.	Stemmed to 'sport'.
Accuracy	Can produce less accurate stems.	Provides more accurate and usable stems.

Copyright © edureka and/or its affiliates. All rights reserved.

Lancaster Stemmer

01

The Lancaster Stemmer is a rule-based stemming algorithm that is known for its **simplicity and high aggressiveness** in reducing words to their base form.

It uses a recursive approach with a predefined set of **stripping rules** to iteratively remove suffixes from words.

02

03

While this makes it computationally efficient, it often results in **over-stemming**, producing very short stems that may **lose meaning**.

Copyright © edureka and/or its affiliates. All rights reserved.

LS, KS, and CAS

Lovins Stemmer (LS)

The first stemming algorithm (1968), it removes suffixes in a **single step** using predefined rules, balancing speed and linguistic accuracy.

Krovetz Stemmer (KS)

Validates stemmed words **against a dictionary** to ensure meaningful stems, avoiding over-stemming for improved accuracy.

Context-Aware Stemming (CAS)

Uses machine learning or rules to stem words based on **sentence context**, prioritizing precision but requiring high computational resources.

Copyright © edureka and/or its affiliates. All rights reserved.

Lemmatization

Copyright © edureka and/or its affiliates. All rights reserved.

Defining Lemmatization

Lemmatization is the process of reducing words to their dictionary form (lemma) while considering their context and part of speech.

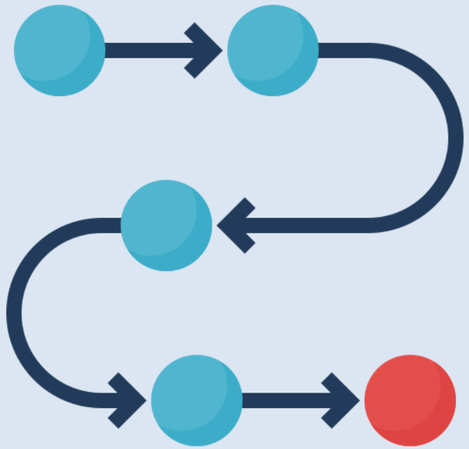
Copyright © edureka and/or its affiliates. All rights reserved.

Process of Lemmatization

01 Morphological Analysis

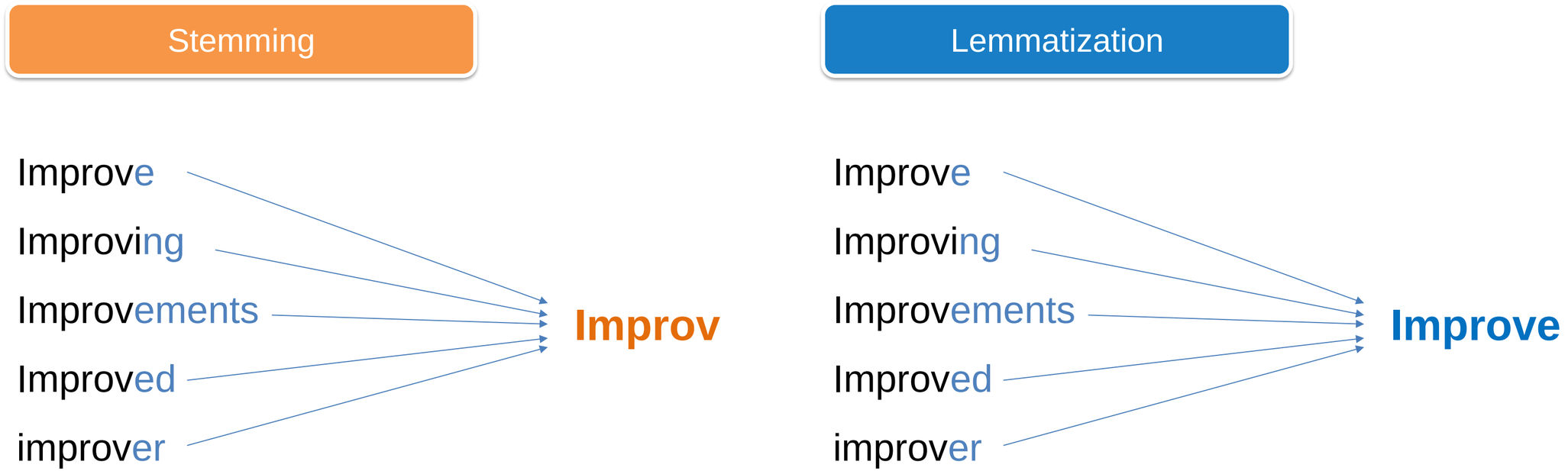
02 Part-of-Speech Tagging

03 Dictionary Lookup



Copyright © edureka and/or its affiliates. All rights reserved.

Lemmatization vs. Stemming



edureka! a Veranda ENTERPRISE

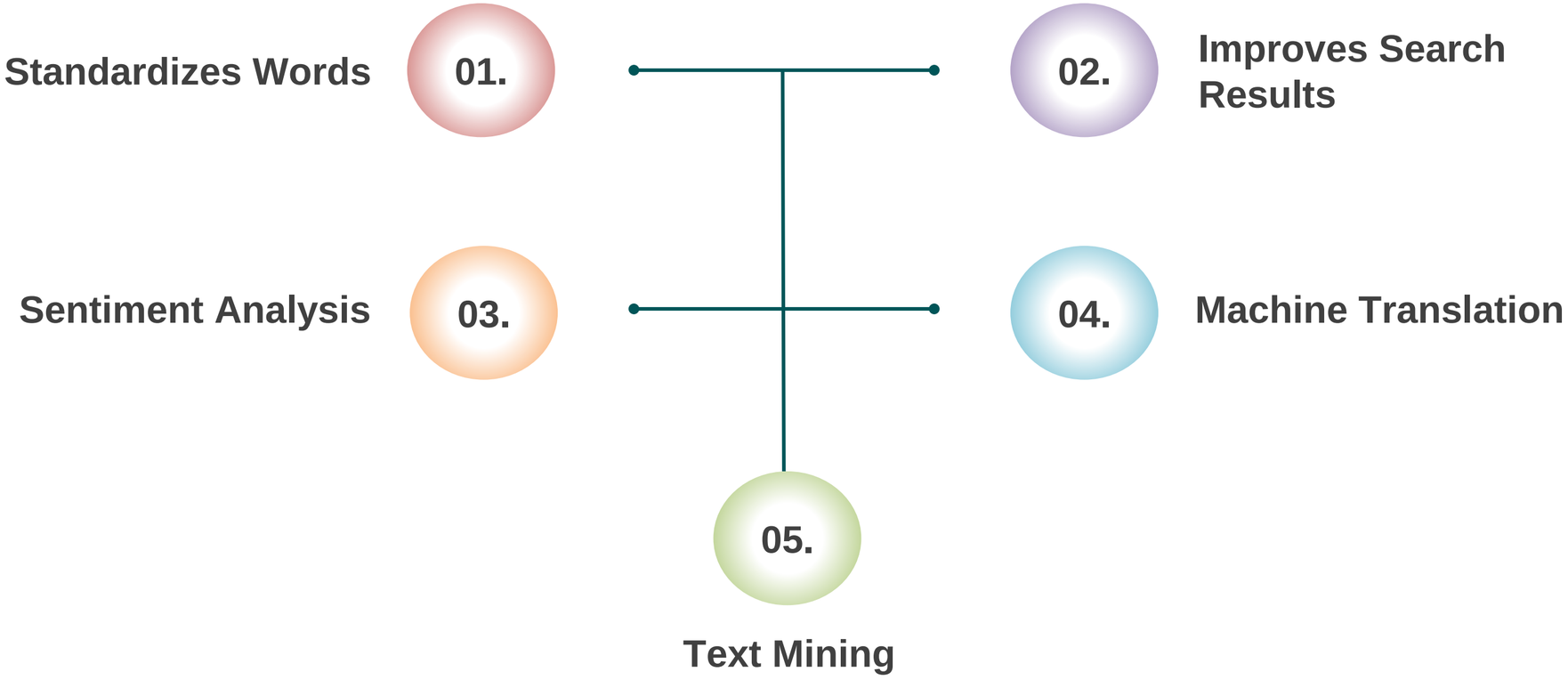
Copyright © edureka and/or its affiliates. All rights reserved.

Lemmatization vs. Stemming Example

Word	Stemming Output	Lemmatization Output	Context
Caring	Car	Care	Verb (action)
Studies	Studi	Study	Plural noun
Running	Run	Run	Verb (action)
Better	Bet	Good	Comparative adjective
Children	Children	Child	Irregular plural noun

Copyright © edureka and/or its affiliates. All rights reserved.

Why Lemmatization is Important?



edureka! a Veranda ENTERPRISE

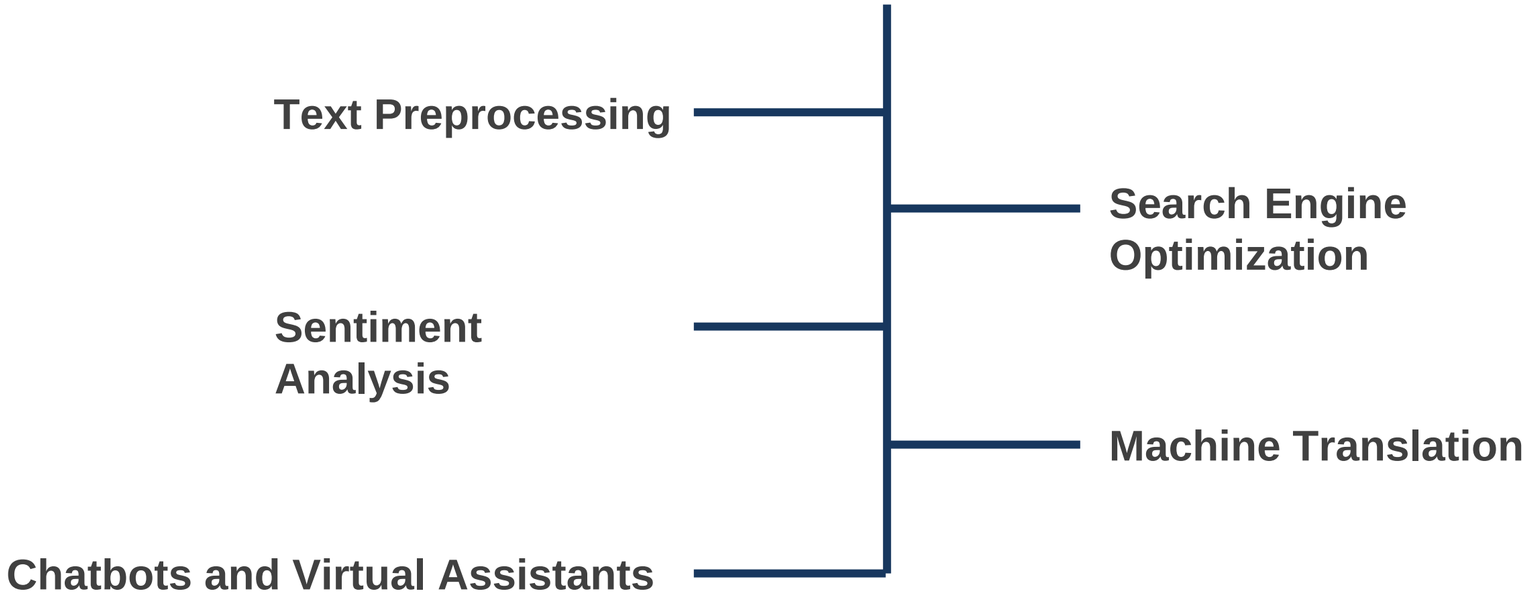
Copyright © edureka and/or its affiliates. All rights reserved.

Lemmatization

Copyright © edureka and/or its affiliates. All rights reserved.

Overview of Lemmatization Applications

Lemmatization has wide-ranging applications across various fields of Natural Language Processing (NLP), search engines, and machine learning.



Copyright © edureka and/or its affiliates. All rights reserved.

Text Preprocessing in NLP

Lemmatization is a text normalization technique to prepare data for processing.

Reduces inflected words to their dictionary form (lemma).

Simplifies unstructured text for tasks like classification, clustering, and analysis.

The children are playing happily. → **The child be play happy.**

Copyright © edureka and/or its affiliates. All rights reserved.

Search Engine Optimization

Lemmatization improves search engines by mapping words to their base forms.

User Queries: Variations in search terms are matched to a single lemma for better results.

Query: “Running shoes” » Searches for “run shoes,” “ran shoes.”

Result: Search engines retrieve results for all related terms.

Copyright © edureka and/or its affiliates. All rights reserved.

Sentiment Analysis

Lemmatization helps in sentiment analysis by reducing words to their base form for consistency.

Different forms of a word (e.g., happy, happier, happiness) are analyzed as a single entity.

Input Sentence	Lemmatized Words	Sentiment
"I loved the movie."	"I love the movie."	Positive
"He is running happily."	"He be run happy."	Positive

Copyright © edureka and/or its affiliates. All rights reserved.

Machine Translation

In machine translation, lemmatization simplifies word variations to improve translation.

Translates sentences by focusing on lemmas rather than inflected forms.

Input: *“She studies daily.”*

Lemmatized: *“She study daily.”*

Translation (to French):
“Elle étudie tous les jours.”

Copyright © edureka and/or its affiliates. All rights reserved.

Chatbots and Virtual Assistants

Lemmatization enhances chatbot responses by normalizing user input.

Helps identify user intent accurately by simplifying variations of words.

User Input: “I am running late.”

Lemmatized: “I be run late.”

Bot Response: “I understand. Do you need to reschedule?”

Copyright © edureka and/or its affiliates. All rights reserved.

Rule Based Lemmatization

Relies on predefined rules for word transformation.

Words are stripped of suffixes or prefixes based on grammatical rules.

Works well for inflectional morphology (tense, plural forms, etc.).

“ing” → Remove “ing” if preceded by a vowel (e.g., running → run).

“ed” → Replace with the root form (e.g., studied → study).

Copyright © edureka and/or its affiliates. All rights reserved.

Dictionary Based Lemmatization

Uses a predefined dictionary of words and their lemmas.

The input word is looked up in the dictionary to return its base form.

Input: “children”

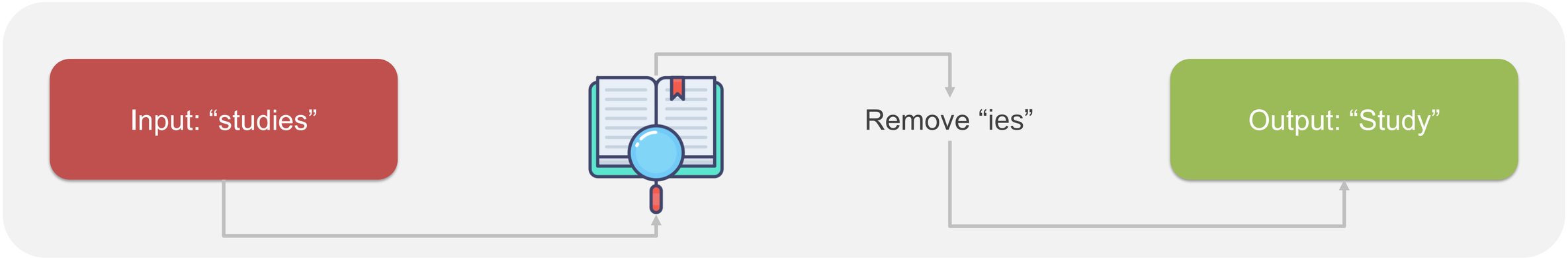
Dictionary Lookup: children → child

Output: child

Copyright © edureka and/or its affiliates. All rights reserved.

Hybrid Lemmatization

Combines rule-based and dictionary-based approaches for improved accuracy.



Copyright © edureka and/or its affiliates. All rights reserved.

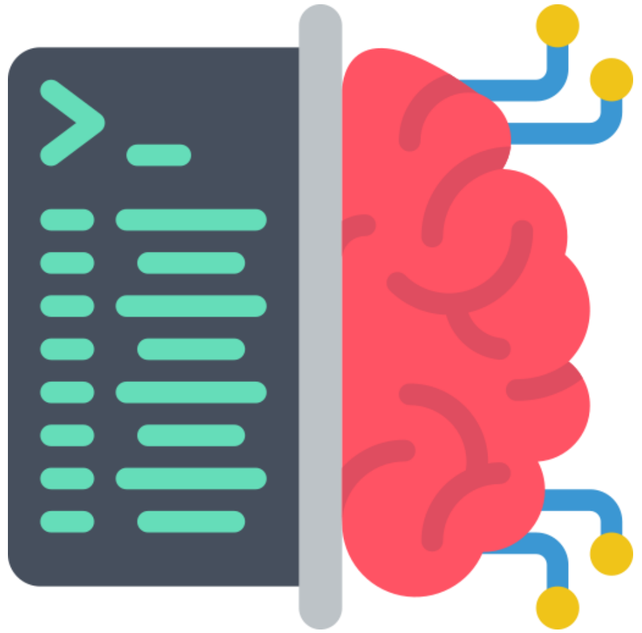
Machine Learning Based Lemmatization

Uses machine learning models to predict lemmas based on training data.

Trained on large text corpora with words and their corresponding lemmas.

Studies → study

went → go



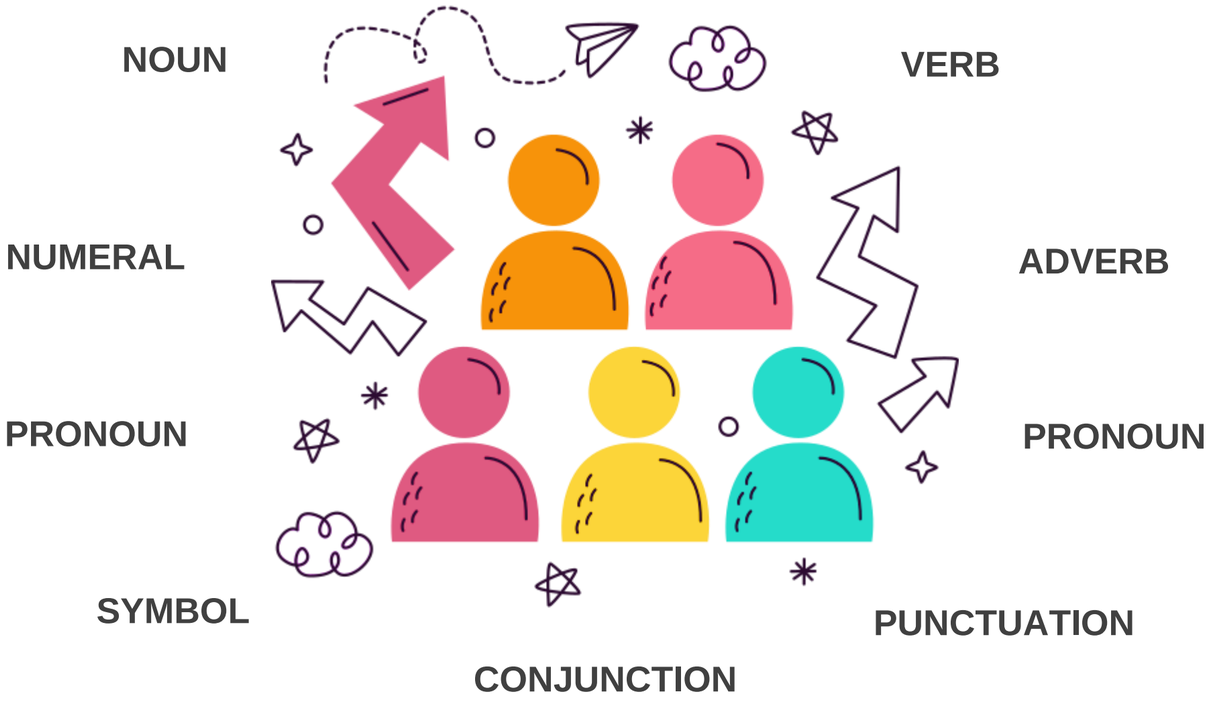
Copyright © edureka and/or its affiliates. All rights reserved.

Part-of-Speech (POS) Tagging

Copyright © edureka and/or its affiliates. All rights reserved.

What is POS ?

Part-of-Speech (POS) tagging is a fundamental technique in natural language processing that assigns grammatical labels—such as noun, verb, adjective, adverb, or pronoun—to individual words within a sentence.



Copyright © edureka and/or its affiliates. All rights reserved.

How POS Tagging Works?

For the sentence: *"The cat sleeps on the mat."*

POS tags could be:

The → **Determiner (DT)**

cat → **Noun (NN)**

sleeps → **Verb (VBZ)**

on → **Preposition (IN)**

the → **Determiner (DT)**

mat → **Noun (NN)**

Copyright © edureka and/or its affiliates. All rights reserved.

Text Representation: Bag of Words (BoW), TF-IDF

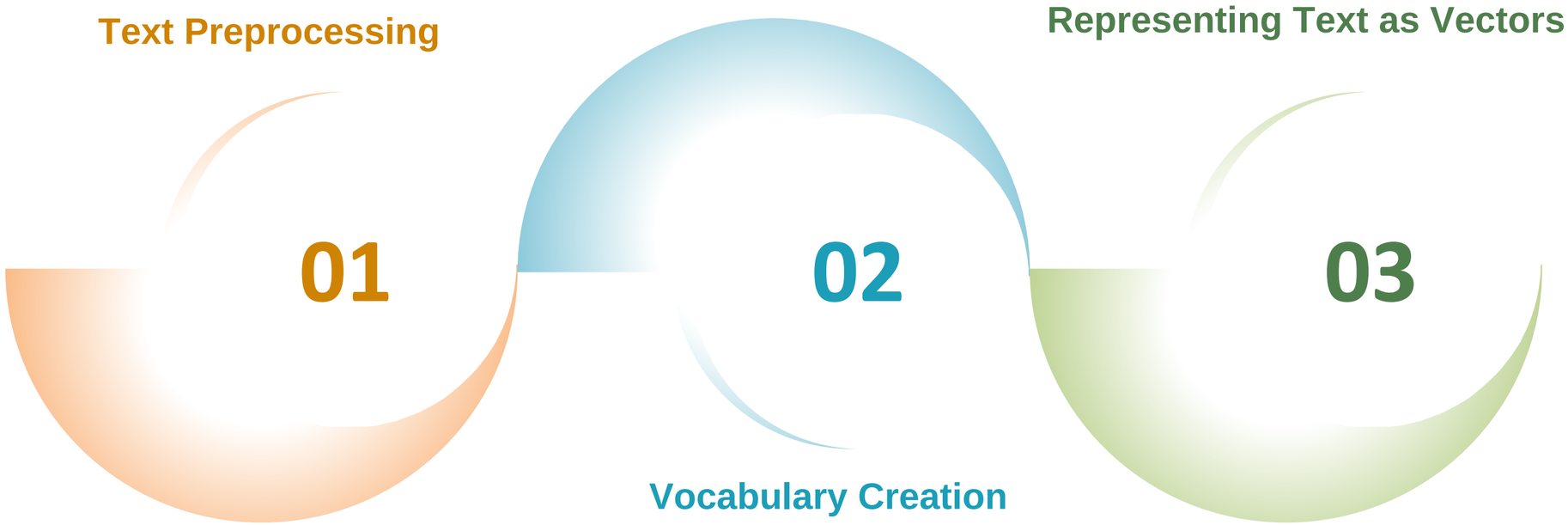
Copyright © edureka and/or its affiliates. All rights reserved.

Introduction to Bag of Words - BoW

Bag of Words (BoW) is an NLP technique that represents text by converting it into a set of unique words, ignoring grammar and word order, and counting word occurrences.

Copyright © edureka and/or its affiliates. All rights reserved.

Steps in Bag of Words - BoW



edureka! a Veranda ENTERPRISE

Copyright © edureka and/or its affiliates. All rights reserved.

Representation of Bag of Words

Suppose we have the following two sentences in a document:

- "The cat sat on the mat."
- "The dog ate the cat."

["the", "cat", "sat", "on", "mat", "dog", "ate"]

Word	The	Cat	Sat	On	Mat	Dog	Ate
Sentence 1	2	1	1	1	1	0	0
Sentence 2	2	1	0	0	0	1	1

Copyright © edureka and/or its affiliates. All rights reserved.

Types of BoW Representation

Binary BoW

Words are represented as binary values: 1 if present, 0 otherwise.

Count-Based BoW

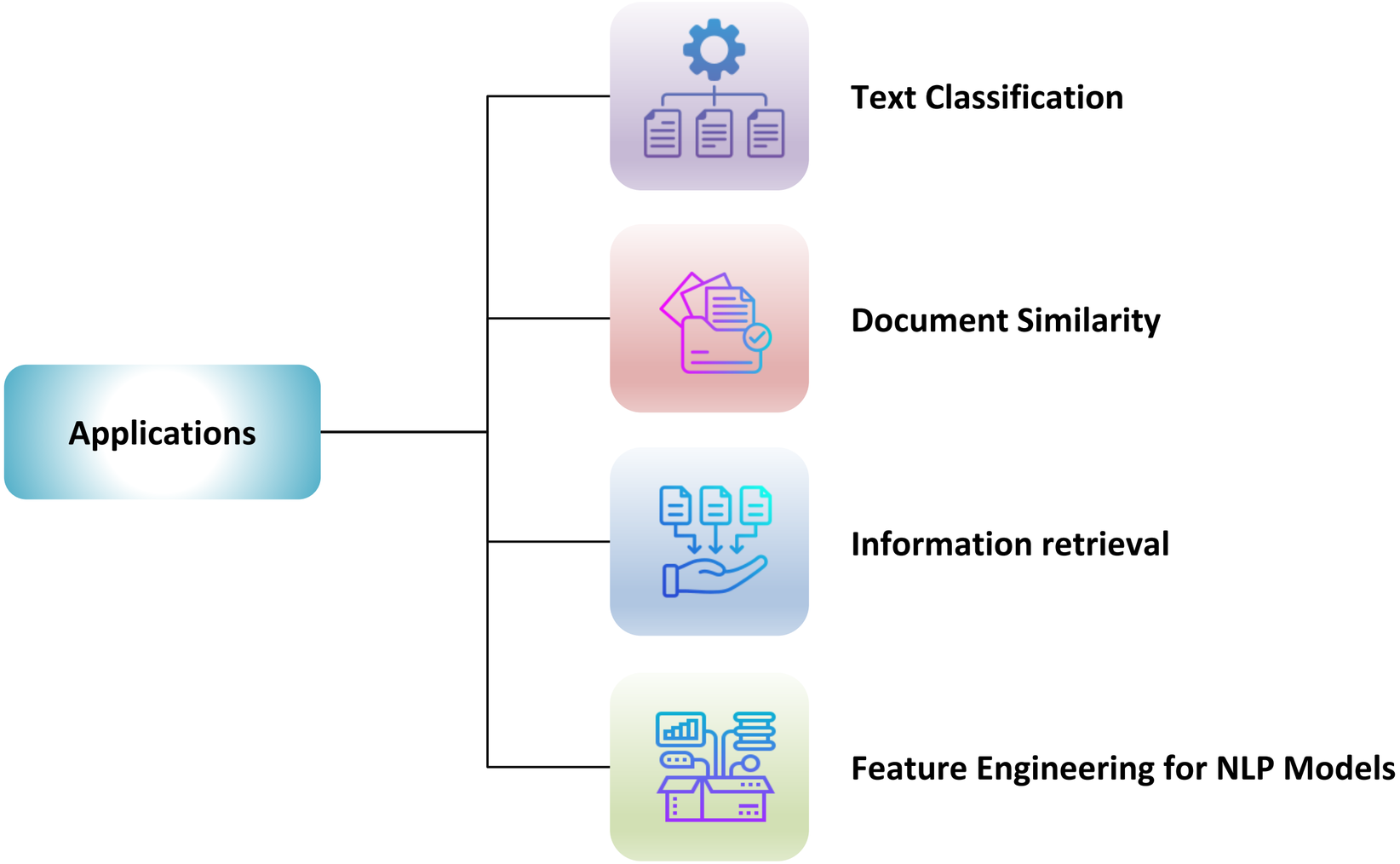
Words are represented by their raw occurrence counts.

Weighted BoW

Assigns weights to words based on their importance

Copyright © edureka and/or its affiliates. All rights reserved.

Application of BoW



edureka! a Veranda ENTERPRISE

Copyright © edureka and/or its affiliates. All rights reserved.

Calculating TF-IDF

Document 1: "The quick brown fox jumped over the lazy dog.“

Document 2: "The fast fox jumped over the quick dog.“

Document 3: "The brown dog barked loudly."

"fox" appears once in Document 1, which has a total of 9 words.TF for "fox" in Document 1 = 1/9 = 0.111.

"fox" appears in Document 1 and Document 2 (2 out of 3 documents).IDF for "fox" = log(3/2)log(3/2) ≈ 0.176.

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t)$$

TF-IDF for "fox" in Document 1 = TF × IDF = 0.111 × 0.176 ≈ 0.0195.

Copyright © edureka and/or its affiliates. All rights reserved.

Key Steps in Computing TF-IDF

Preprocess the Text

Tokenize, remove stop words, lowercase, and apply stemming/ lemmatization.

Calculate TF for Each Term

Compute term frequency for all words in the document.

Compute IDF for Each Term

Use the entire corpus to calculate how rare each word is.

Multiply TF by IDF

Generate the final TF-IDF score for each word in the document.

Copyright © edureka and/or its affiliates. All rights reserved.

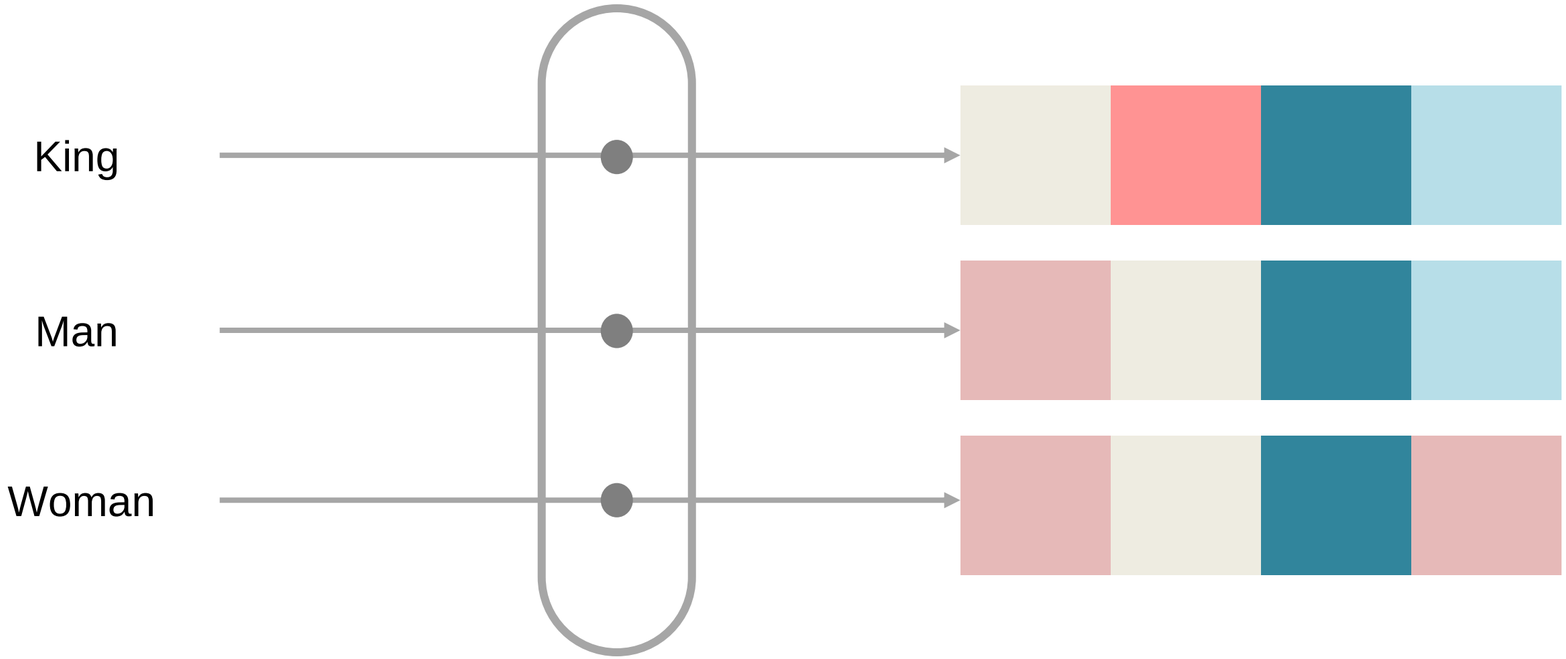
Word Embeddings: Word2Vec, GloVe, FastText

edureka! a Veranda ENTERPRISE

Copyright © edureka and/or its affiliates. All rights reserved.

Introduction to Word2Vec

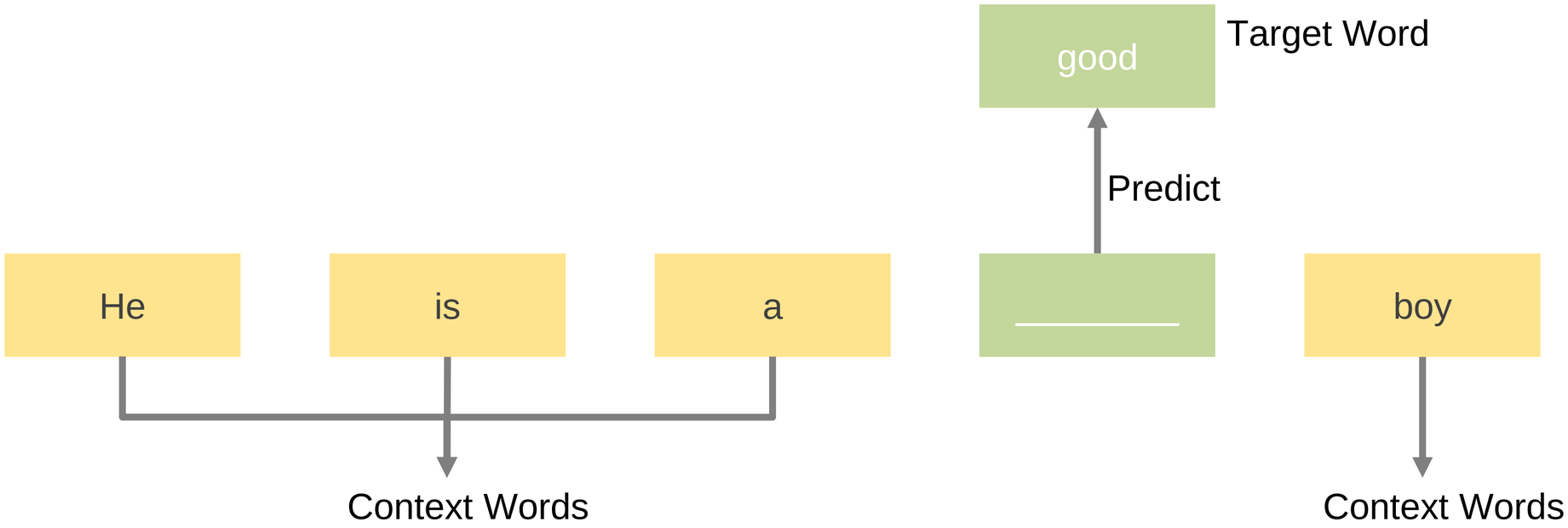
Word2Vec generates word embeddings that capture semantic relationships



Copyright © edureka and/or its affiliates. All rights reserved.

CBOW Model

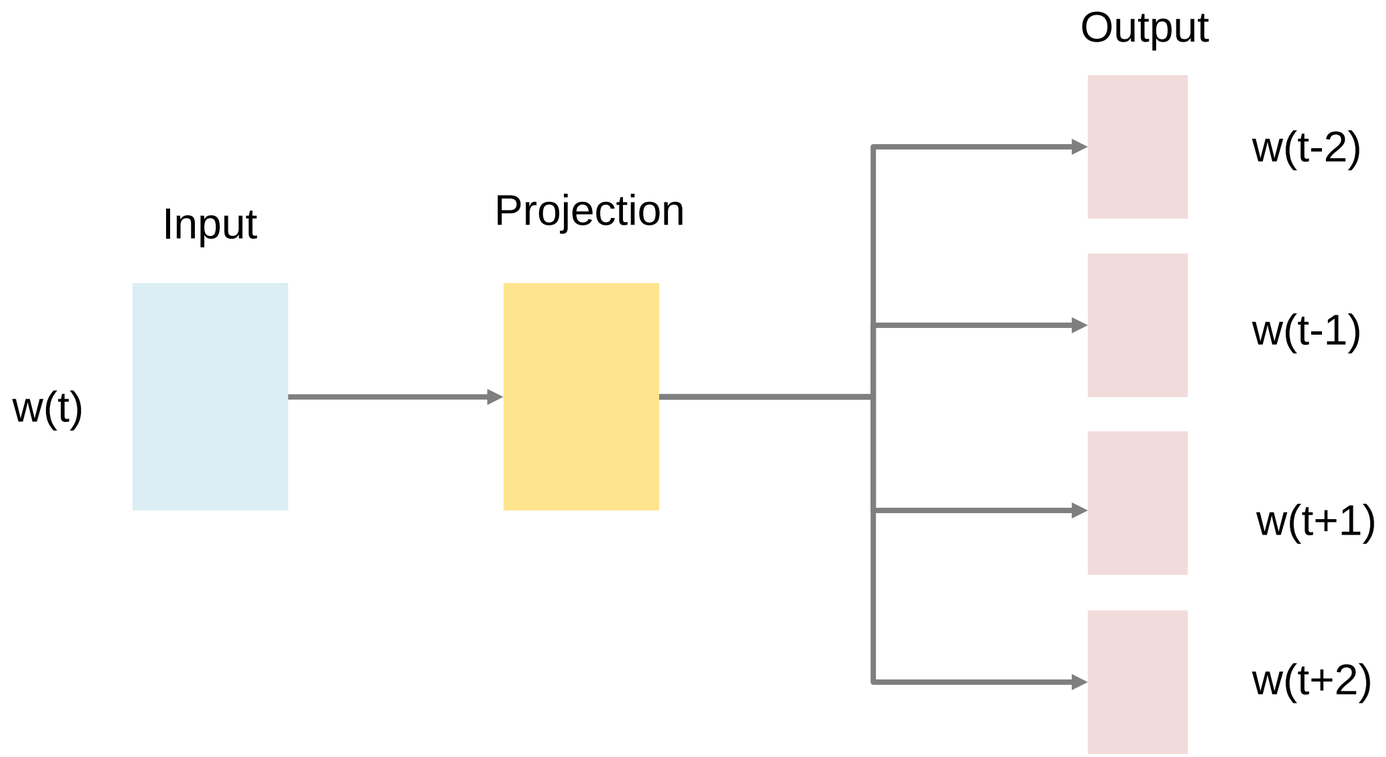
Continuous Bag of Words (CBOW) is a neural network model that learns word embeddings by predicting a target word based on its context words



Copyright © edureka and/or its affiliates. All rights reserved.

Skip-Gram Model

Skip-gram is a popular algorithm used in natural language processing (NLP) to create word embeddings



Copyright © edureka and/or its affiliates. All rights reserved.

Negative Sampling and Hierarchical Softmax

Problem: High computational cost of Softmax for large vocabularies.

Negative Sampling

Samples a few "negative" examples (random words) to reduce computations.

Hierarchical Softmax

Uses a binary tree structure for faster computation of probabilities.

Copyright © edureka and/or its affiliates. All rights reserved.

Introduction to GloVe

Global Vectors for Word Representation (GloVe), is an unsupervised learning algorithm that generates vector representations, or embeddings, of words.

Limitations

Focuses only on local context (neighboring words).

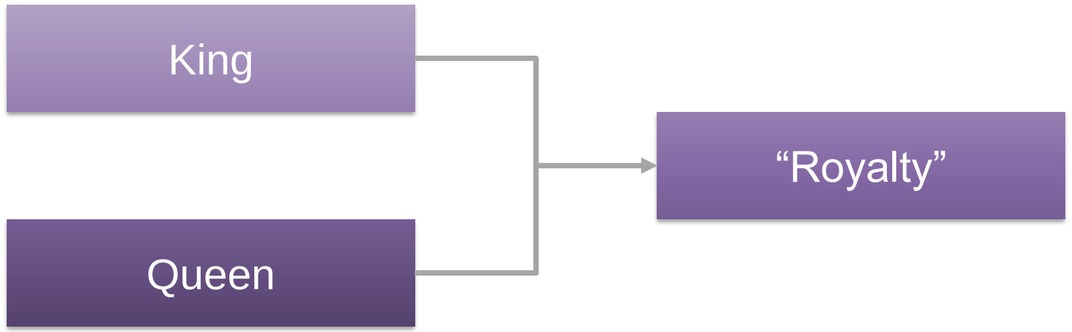
GloVe's Approach

Combines local context with global statistical information.

Copyright © edureka and/or its affiliates. All rights reserved.

Role of Co-occurrence Matrix

A matrix where each entry $M(i,j)$ represents how often word i co-occurs with word j in the corpus.



Copyright © edureka and/or its affiliates. All rights reserved.

Co-occurrence Matrix Representation

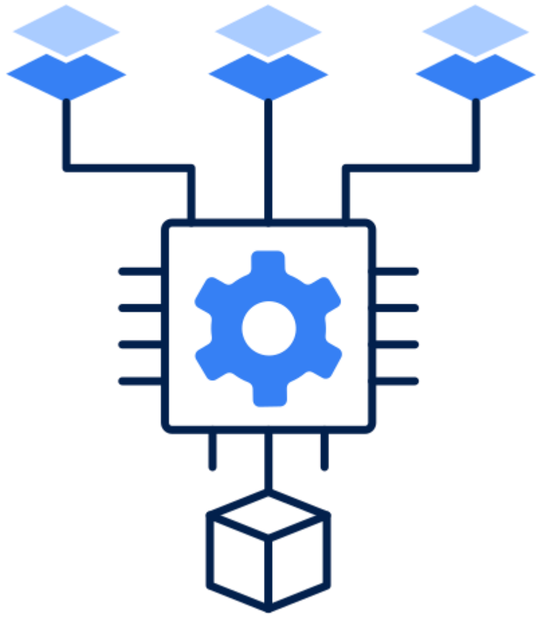
	first	Is	second	sentence	the	this
first	0.0	1.0	0.0	1	1.0	1.0
is	1.0	0.0	1.0	3	2.0	2.0
second	0.0	1.0	0.0	2	1.0	1.0
sentence	1.0	3.0	2.0	2	3.0	3.0
the	1.0	2.0	1.0	3	0.0	2.0
this	1.0	2.0	1.0	3	2.0	0.0

Copyright © edureka and/or its affiliates. All rights reserved.

What are Pre-Trained Embeddings

Word embeddings trained on large corpora to capture semantic relationships and linguistic patterns.

GloVe **Word2Vec** **Fast Text**



Copyright © edureka and/or its affiliates. All rights reserved.

Overview of FastText Embeddings

Leverages sub word-level information, breaking words into character n-grams.

jumping → <ju>, <jum>, <jump>, <mpin>, <ping>

Copyright © edureka and/or its affiliates. All rights reserved.

Comparing FastText, Word2Vec and GloVe

Feature	fast Text	Word2Vec	GloVe
Handles OOV Words	Yes(via sub words)	No	No
Context Focus	Local + Sub words	Local Context Only	Global Statistics
Morphological Support	Strong(sub word embeddings)	Weak	Weak
Training Complexity	Higher (sub word-level model)	Moderate	Moderate
Pre-trained Models	Available	Available	Available

Copyright © edureka and/or its affiliates. All rights reserved.

Processing Code-mixed (multilingual) Text

Copyright © edureka and/or its affiliates. All rights reserved.

What is Code-mixed (multilingual) Text ?

Code-mixed (multilingual) text refers to language content where **two or more languages** are interwoven within the same sentence, phrase, or conversation.

*“I'm heading to the market to buy some **sabzi** and fruits.”*

“Sabzi ?”

Copyright © edureka and/or its affiliates. All rights reserved.

Why It Occurs ?

Social Identity

Practical Reasons

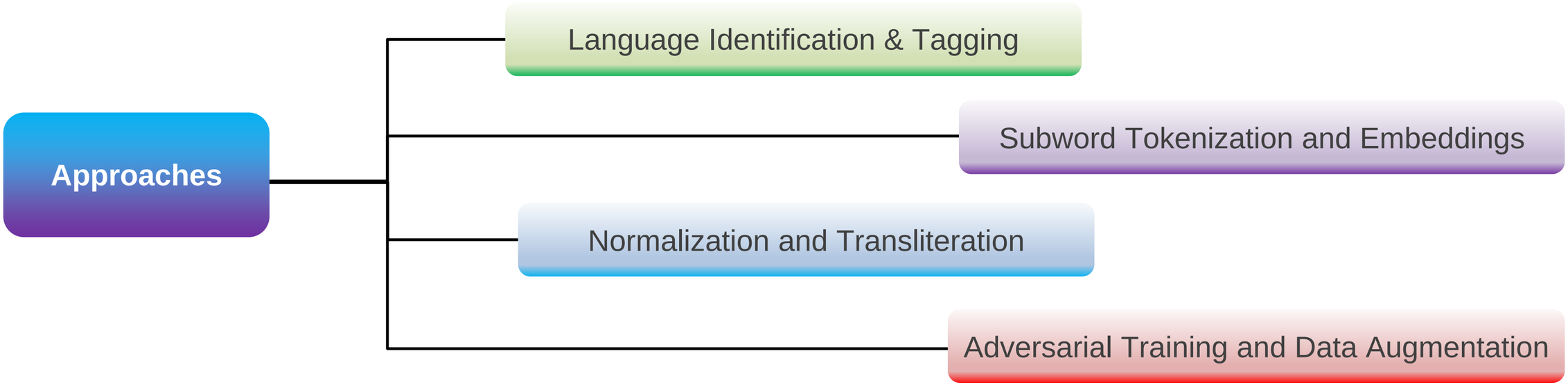
Expressive Nuance

edureka! a Veranda ENTERPRISE

Copyright © edureka and/or its affiliates. All rights reserved.

Tackling Code-mixed (multilingual) Text

To tackle code-mixed (multilingual) text using NLP, here are a few key approaches:



Copyright © edureka and/or its affiliates. All rights reserved.

Language Identification & Tagging

Multilingual pre-trained models (like **mBERT or XLM-R**) use **subword tokenization** (WordPiece or BPE) to break words into smaller units. This approach helps in handling:

OOV words or intra-word mixing:

For example, “chating” might be tokenized into “chat” and “ing”, allowing the model to infer meaning even when languages mix at the subword level.

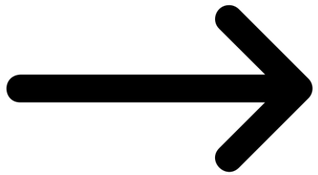
“chating” **“chat” + “ing”**

Copyright © edureka and/or its affiliates. All rights reserved.

Subword Tokenization and Embeddings

Code-mixed text is often informal and noisy. Normalization converts non-standard **spellings** into a **standard form**, and transliteration maps words written in one script to another.

For instance: A user might write “donlod” instead of “download” in a mixed sentence. A normalization module can correct this before further processing.

“donlod”  **“download”**

Copyright © edureka and/or its affiliates. All rights reserved.

Adversarial Training and Data Augmentation

When real code-mixed data is scarce, synthetic code-mixed examples can be generated using adversarial attacks. These examples are used to augment the training data, making the model more robust.

Generating adversarial examples: Perturb monolingual sentences by substituting words with their counterparts from another language (e.g., replacing “help” with “ayuda”) to simulate real-world code-switching.

Copyright © edureka and/or its affiliates. All rights reserved.

Text Classification in NLP using Common ML Models

edureka! a Veranda ENTERPRISE

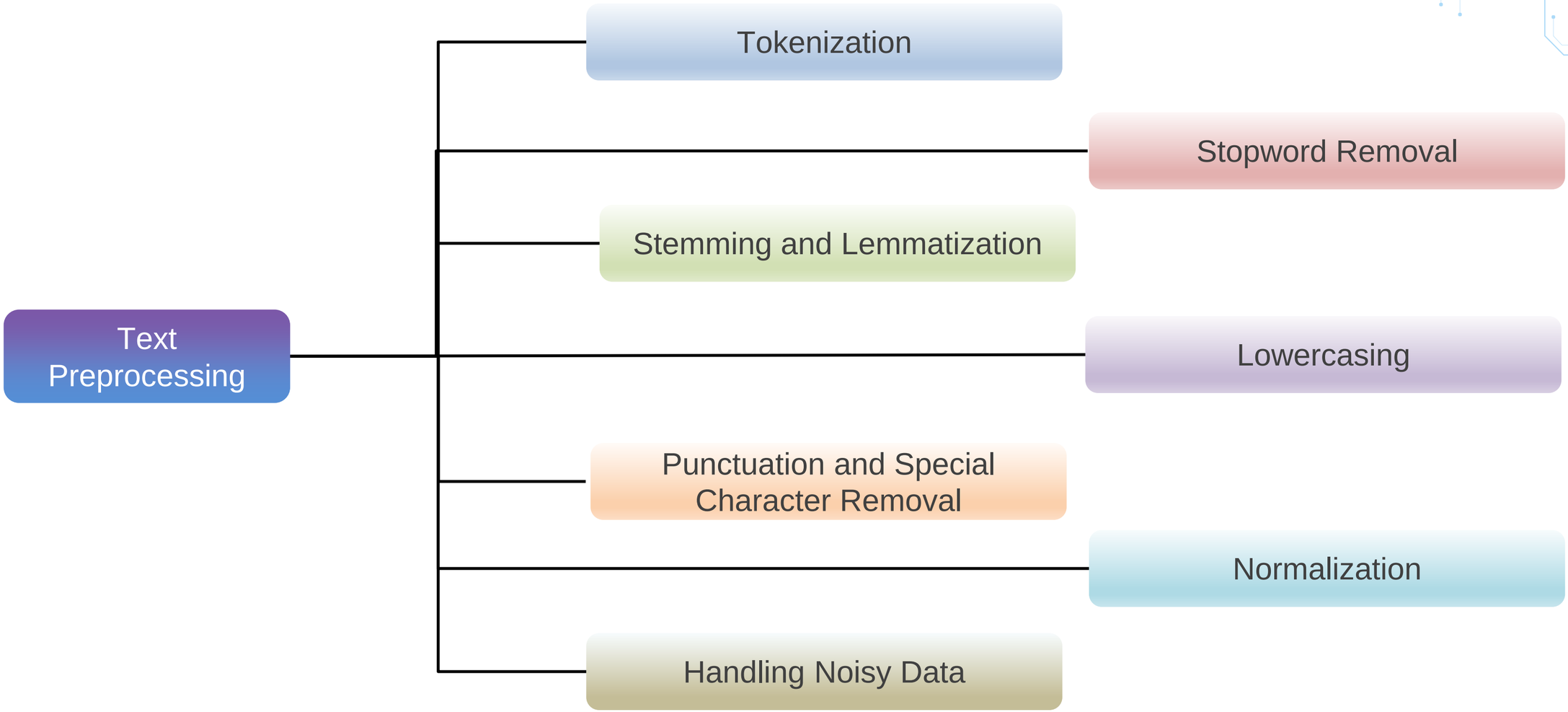
Copyright © edureka and/or its affiliates. All rights reserved.

What is Text Classification in NLP?

Text classification is a Natural Language Processing (NLP) task that involves automatically assigning predefined categories to text based on its content. It is widely used in spam detection, sentiment analysis, topic categorization, and more.

Copyright © edureka and/or its affiliates. All rights reserved.

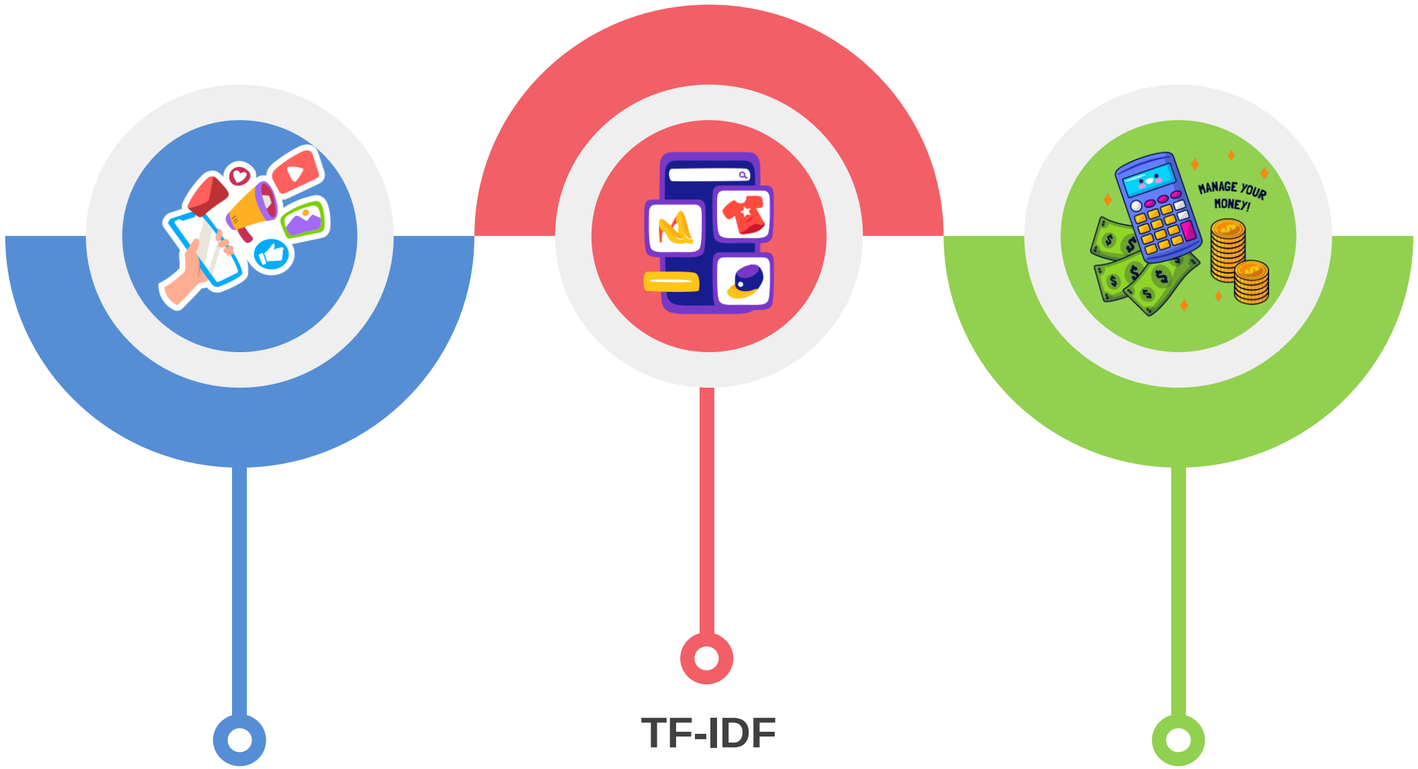
Key Steps in Text Preprocessing



edureka! a Veranda ENTERPRISE

Copyright © edureka and/or its affiliates. All rights reserved.

Feature Extraction (Text into Numerical Form)



edureka! a Veranda ENTERPRISE

Copyright © edureka and/or its affiliates. All rights reserved.

Model Training for Text Classification

Naïve Bayes Classifier

Support Vector Machine (SVM)

Decision Trees & Random Forest

Deep Learning Models

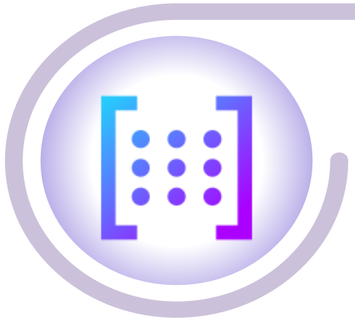
Copyright © edureka and/or its affiliates. All rights reserved.

Prediction and Model Evaluation

Once the model is trained, it is tested on **new text** to classify it into one of the predefined categories. The performance is evaluated using:

Accuracy

Precision, Recall, and F1-score



Confusion Matrix

Copyright © edureka and/or its affiliates. All rights reserved.

Feature Selection for Classification

edureka! a Veranda ENTERPRISE

Copyright © edureka and/or its affiliates. All rights reserved.

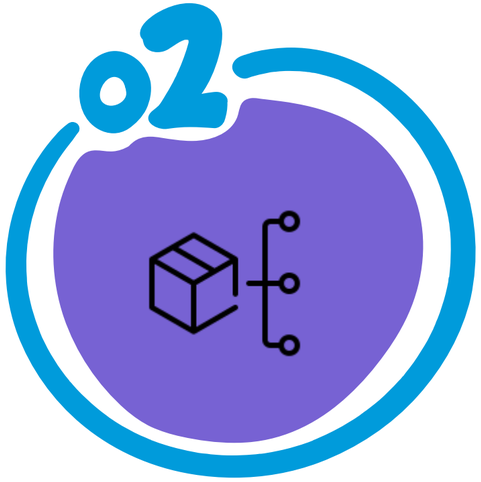
What is Feature Selection?

In a dataset, not all features contribute equally to the classification task. Some may be irrelevant or highly correlated, which can negatively impact the model. Feature selection helps identify the best subset of features that maximize classification accuracy.

Copyright © edureka and/or its affiliates. All rights reserved.

Feature Selection- Filter Methods

Correlation Coefficient
(Pearson/Spearman/
Kendall)



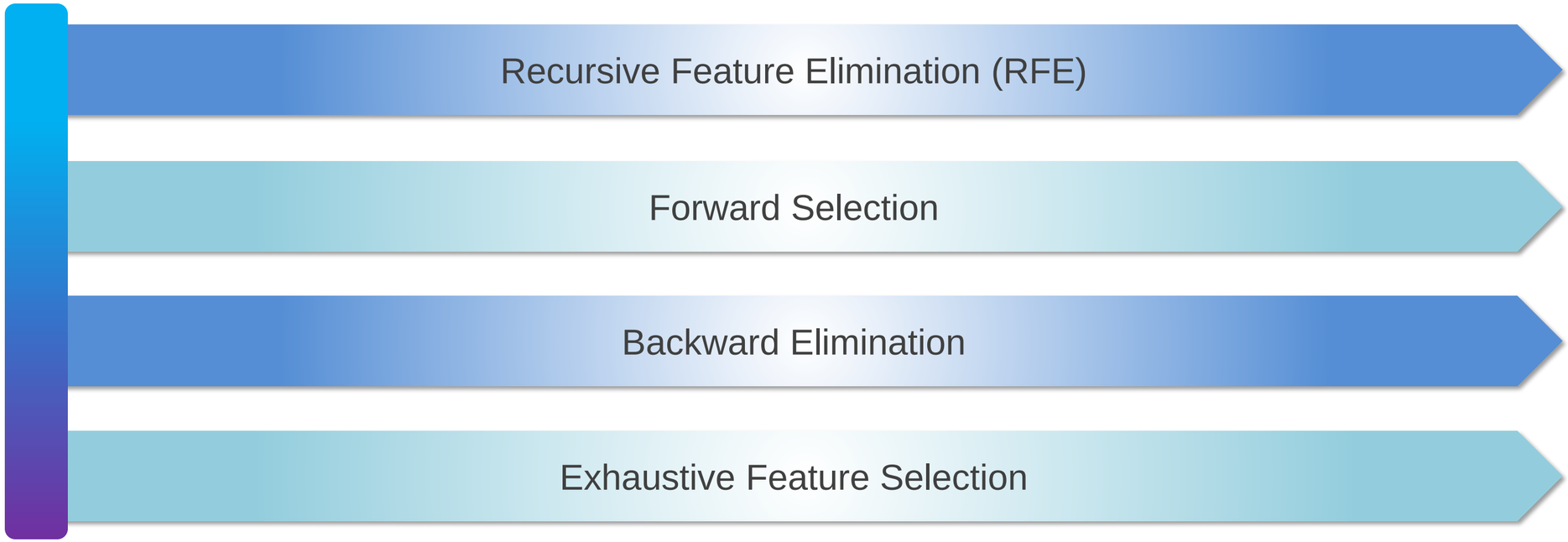
Chi-Square Test
(For Categorical Data)

Mutual Information (MI)

Variance Threshold

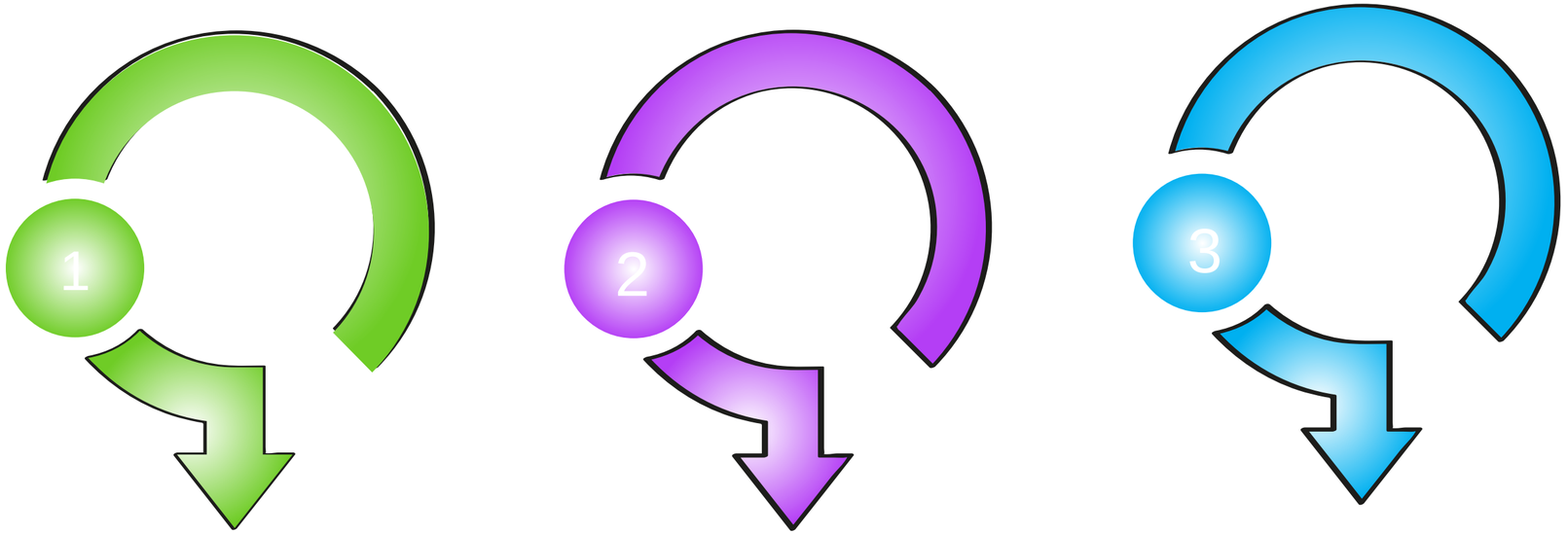
Copyright © edureka and/or its affiliates. All rights reserved.

Wrapper Methods (Feature Subset Evaluation)



Copyright © edureka and/or its affiliates. All rights reserved.

Embedded Methods(Selection During Training)



Lasso (L1 Regularization) in Logistic Regression

Decision Tree & Random Forest Feature Importance

Gradient Boosting Feature Importance (XGBoost, LightGBM, CatBoost)

Copyright © edureka and/or its affiliates. All rights reserved.

Feature Selection vs. Feature Extraction

Feature Selection	Feature Extraction
Selects a **subset** of existing features	Creates **new features** from existing ones
Removes redundant or irrelevant features	Transforms data into a different representation
Examples: RFE, Mutual Information, Lasso	Examples: PCA, Autoencoders, Word Embeddings
Keeps original feature meanings	Can lose interpretability (e.g., PCA reduces dimensions)

Copyright © edureka and/or its affiliates. All rights reserved.

BoW, TF-IDF and Word Embeddings (Demonstration)

Note: Refer to Module 2: Demo 1 on LMS for detailed steps.

edureka! a Veranda ENTERPRISE

Copyright © edureka and/or its affiliates. All rights reserved.

Evaluating Stemming and Lemmatization Methods for Text Preprocessing (Demonstration)

Note: Refer to Module 2: Demo 2 on LMS for detailed steps.

Copyright © edureka and/or its affiliates. All rights reserved.

Summary

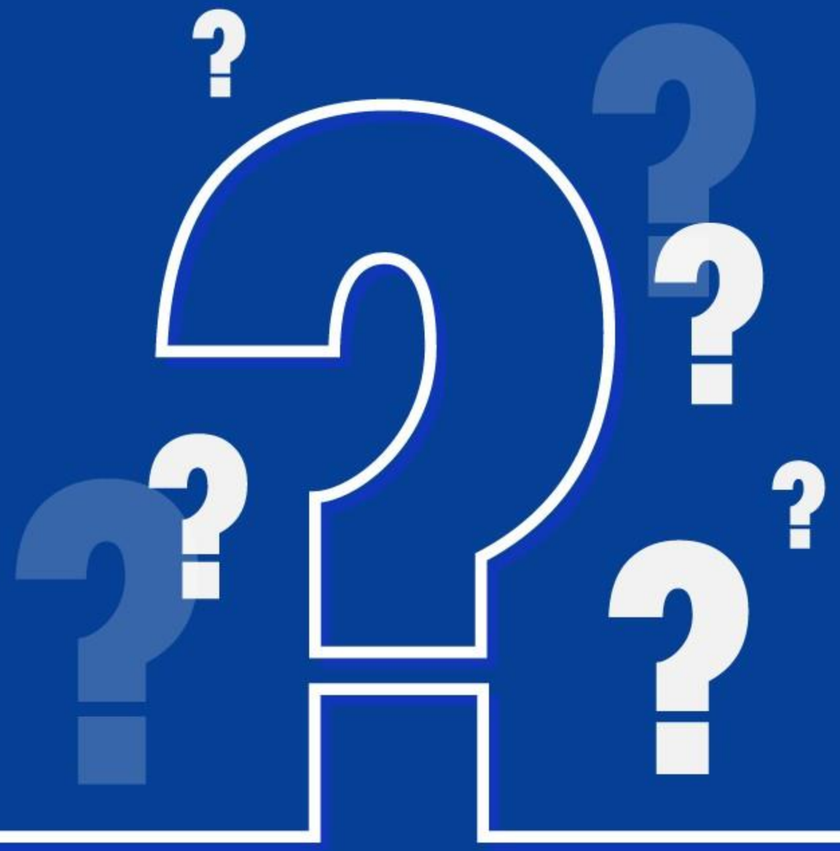
In this lesson, you have learned to:

- Explain key techniques like stemming, lemmatization, and POS tagging used in NLP preprocessing.
- Illustrate how text can be represented using methods such as Bag of Words.
- Describe different word embedding models including Word2Vec, GloVe, and FastText for language representation.
- Implement text classification strategies and feature selection techniques for multilingual and code-mixed data.

Copyright © edureka and/or its affiliates. All rights reserved.

edureka! a Veranda ENTERPRISE

Questions



Copyright © edureka and/or its affiliates. All rights reserved.

edureka! a Veranda ENTERPRISE

Feedback

© Brain4ce Education Solutions Pvt. Ltd. All rights reserved.

edureka! a Veranda ENTERPRISE

Thank You

For information, Please Visit our Website
www.edureka.co

Copyright © edureka and/or its affiliates. All rights reserved.