ADVANCED SOCIAL, TEXT AND MEDIA ANALYTICS

Text Mining

UNIT – I

Text Mining: Introduction, Core text mining operations, Preprocessing techniques, Categorization, Clustering, Information extraction, Probabilistic models for information extraction, Text mining applications, Methods & Approaches: Content Analysis;

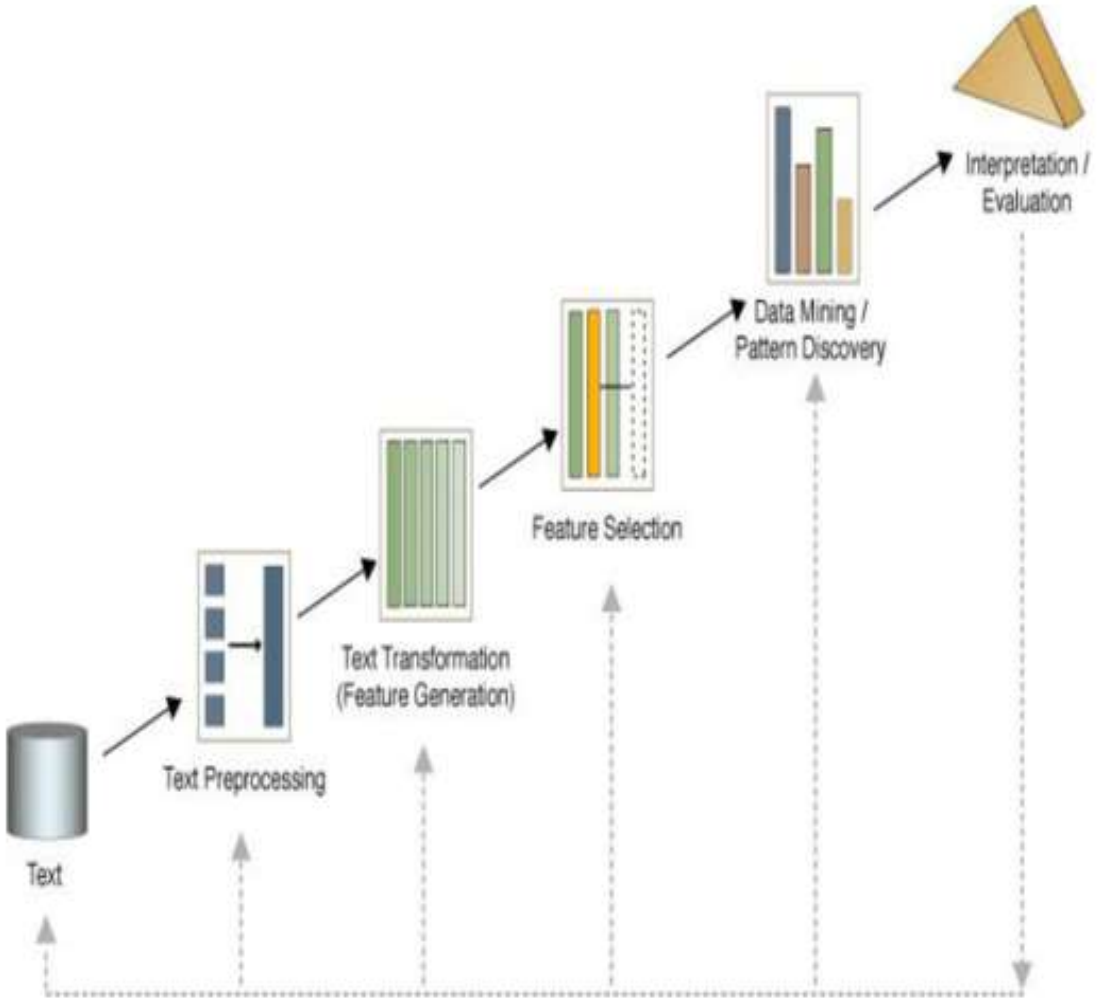
What is Text Mining

Text mining, also known as text data mining, is the process of transforming **unstructured text** into a structured format to identify meaningful patterns and new insights.

TEXT MINING

Text mining process

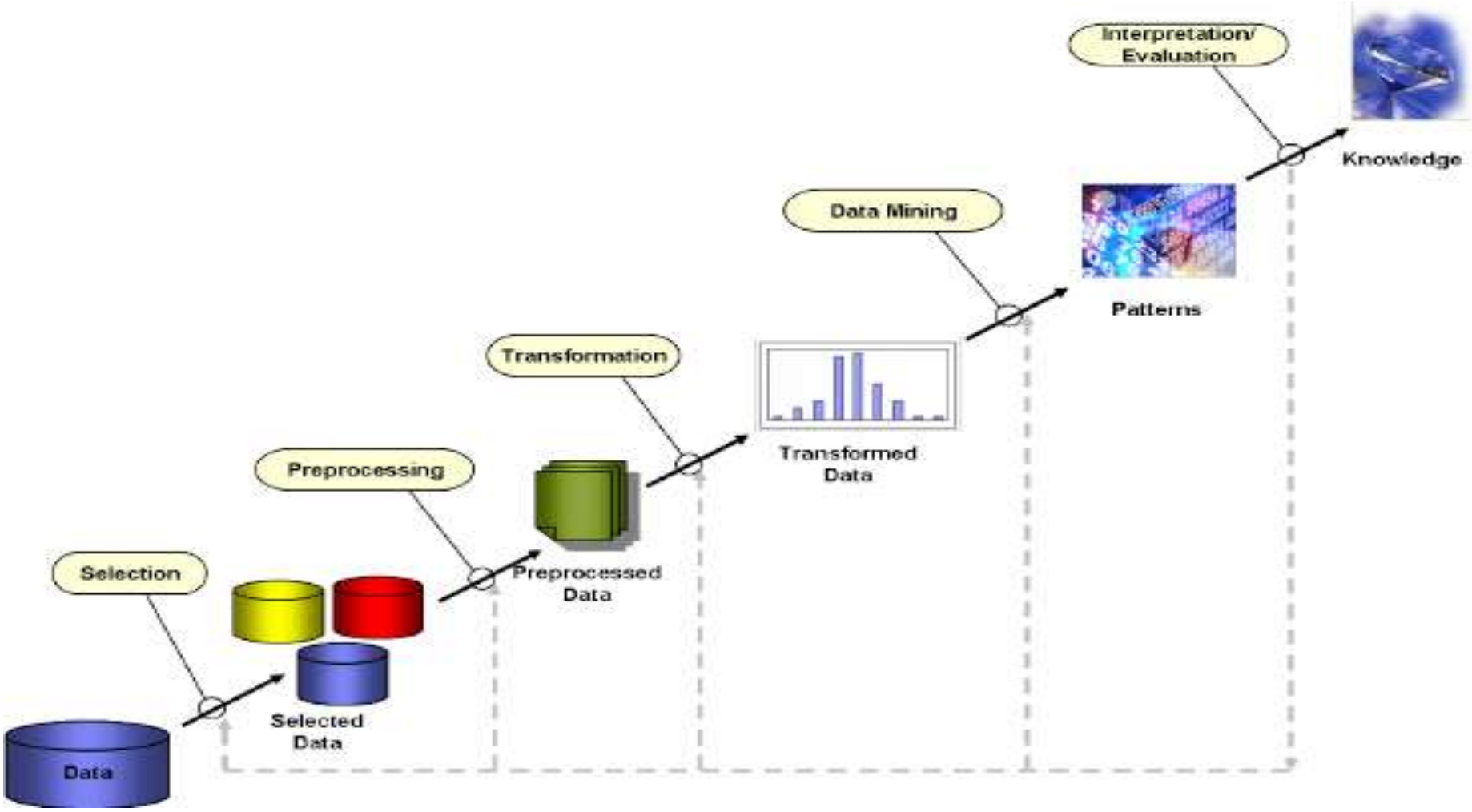
- Text preprocessing
  - Syntactic/Semantic text analysis
- Features Generation
  - Bag of words
- Features Selection
  - Simple counting
  - Statistics
- Text/Data Mining
  - Classification- Supervised learning
  - Clustering- Unsupervised



Cont ..

By applying advanced analytical techniques, such as Naïve Bayes, Support Vector Machines (SVM), and other deep learning algorithms, companies are able to explore and discover hidden relationships within their **unstructured data**.

KDD Steps


Interpretation/
Evaluation
Knowledge
Data Mining
Patterns
Transformation
Transformed
Data
Preprocessing
Preprocessed
Data
Selection
Selected
Data
Data

Types of Data

Text is a one of the most common data types within databases. Depending on the database, this data can be organized as:

1. **Structured data**
2. **Semi structured data**
3. **Un structured data**

Example

Structured data: This data is standardized into a tabular format with numerous rows and columns, making it easier to store and process for analysis and machine learning algorithms. Structured data can include inputs such as names, addresses, and phone numbers.

Unstructured data: This data does not have a predefined data format. It can include text from sources, like social media or product reviews, or rich media formats like, video and audio files.

Semi-structured data: As the name suggests, this data is a blend between structured and unstructured data formats. While it has some organization, it doesn’t have enough structure to meet the requirements of a relational database. Examples of semi-structured data include XML, JSON and HTML files.

Example

Unstructured data

The university has 5600 students.
John's ID is number 1, he is 18 years old and already holds a B.Sc. degree.
David's ID is number 2, he is 31 years old and holds a Ph.D. degree. Robert's ID is number 3, he is 51 years old and also holds the same degree as David, a Ph.D. degree.

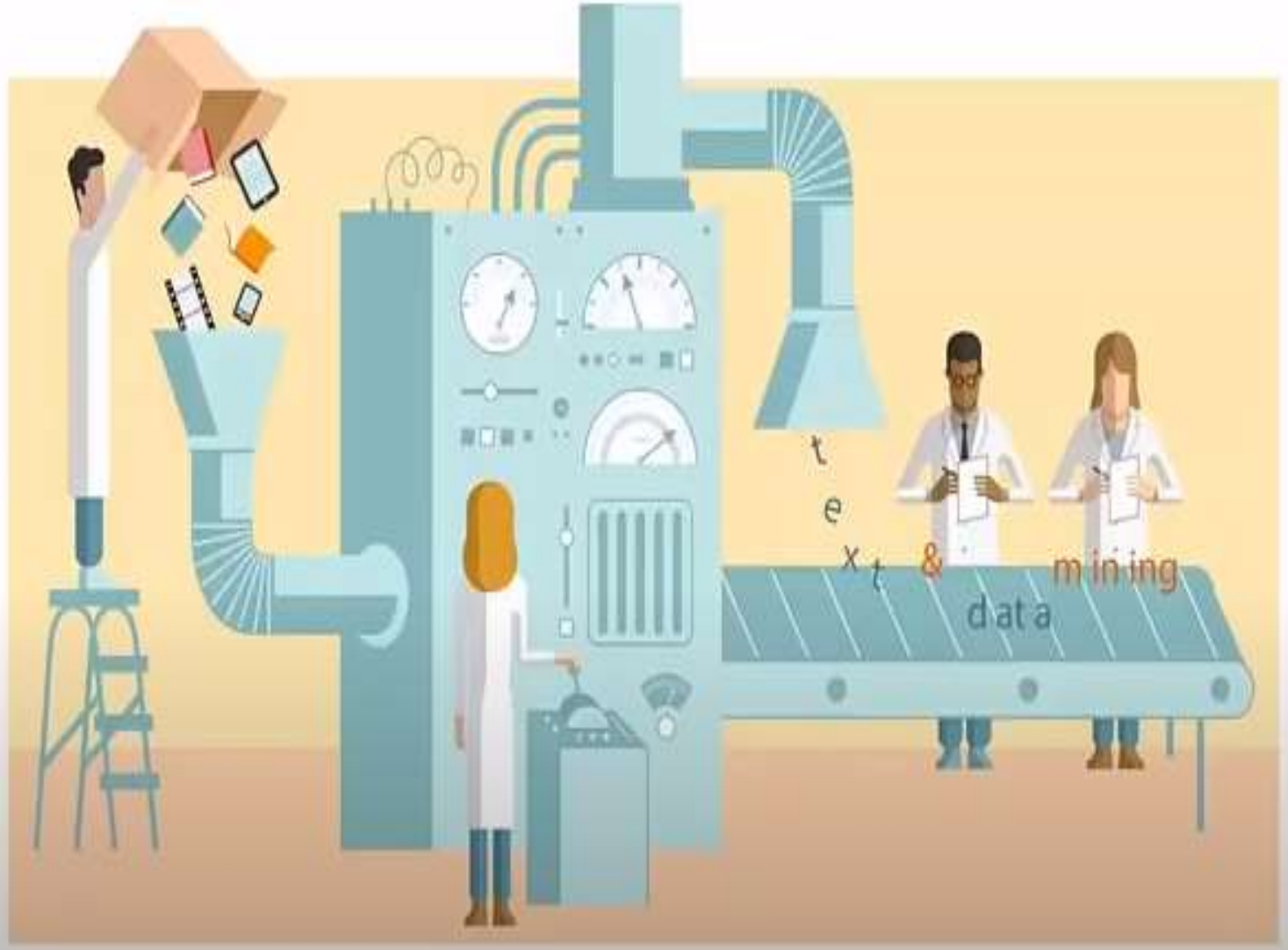
Semi-structured data

```
<University>
 <Student ID="1">
  <Name>John</Name>
  <Age>18</Age>
  <Degree>B.Sc.</Degree>
 </Student>
 <Student ID="2">
  <Name>David</Name>
  <Age>31</Age>
  <Degree>Ph.D. </Degree>
 </Student>
....
</University>
```

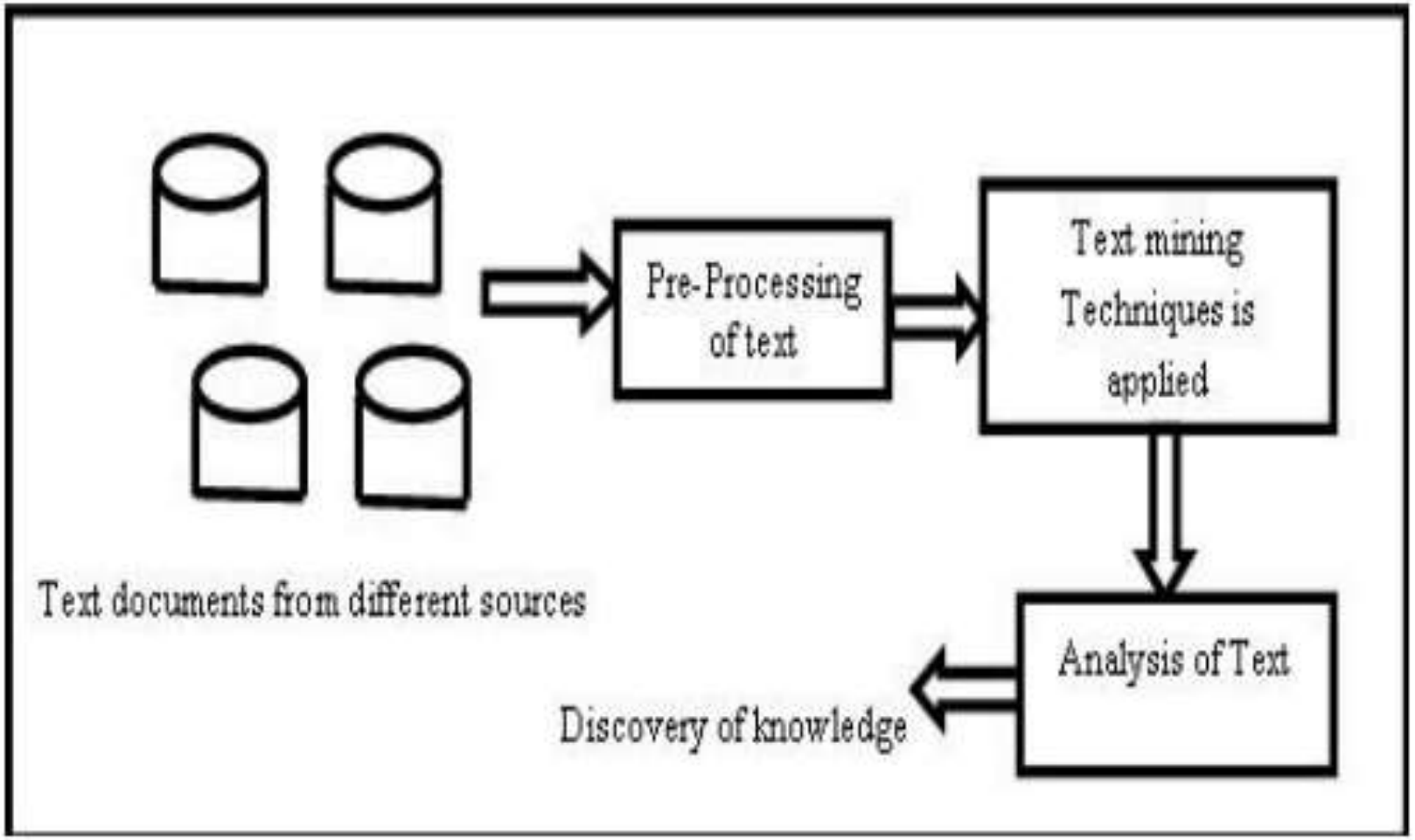
Structured data

ID	Name	Age	Degree
1	John	18	B.Sc.
2	David	31	Ph.D.
3	Robert	51	Ph.D.
4	Rick	26	M.Sc.
5	Michael	19	B.Sc.

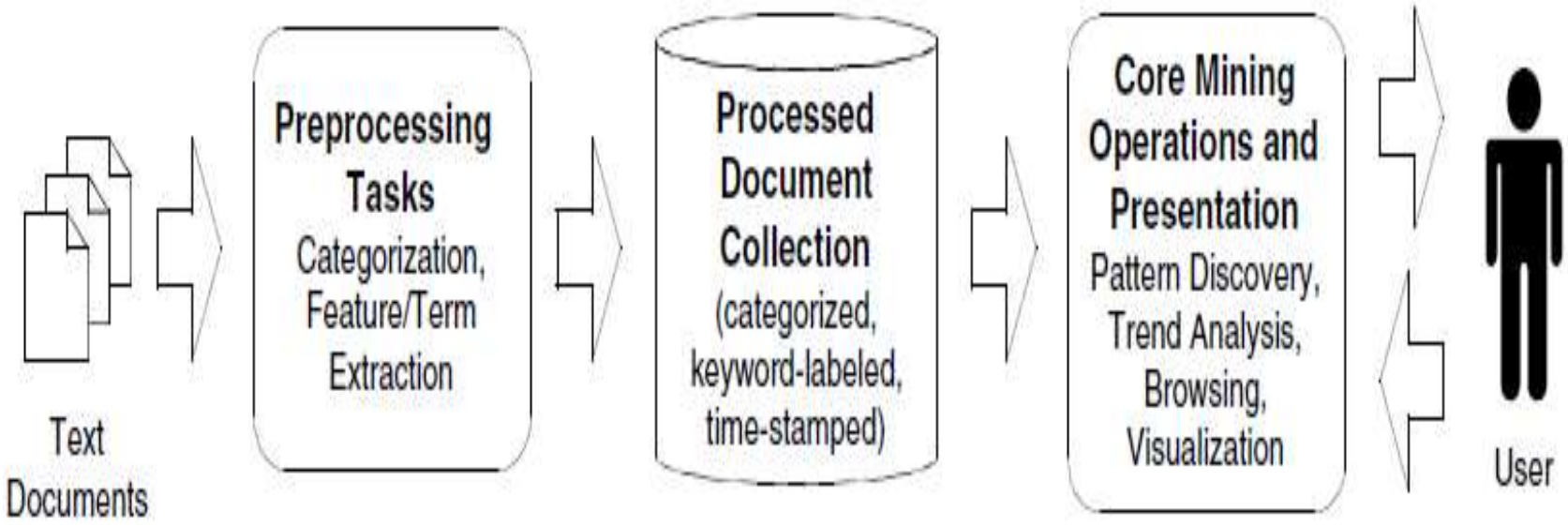
Text-mining is a form of data-mining



Text Mining Process



High-level text mining functional architecture



Core text mining operations

Core text mining operations consist of various mechanisms for **discovering patterns** of concept occurrence within a given document collection or subset of a document collection.

The three most common types of patterns encountered in text mining are:

1. **Frequent sets**
2. **Associations**
3. **Distributions**

Frequent and Near Frequent Sets

A set of concepts represented in the document collection with co-occurrences at or above a minimal support level (given as a threshold parameter *s*; i.e., all the concepts of the frequent concept set appear together in at least *s* documents).

Although originally defined as an intermediate step in finding *association rules*, frequent concept sets contain a great deal of information of use in text mining.

Cont ..

Discovery methods for frequent concept sets in text mining build on the ***Apriori*** algorithm of Agrawal et al. (1993) used in data mining for **market basket** association problems.

With respect to frequent sets in natural language application, ***support*** is the number (or percent) of documents containing the given rule that is, the co-occurrence frequency.

Confidence is the percentage of the time that the rule is true.

Associations

A formal description of association rules was first presented in the same research on “market basket” problems that led to the identification of ***frequent sets*** in data mining.

Subsequently, associations have been widely discussed in literature on knowledge discovery targeted at both structured and unstructured data.

Cont ..

In text mining, ***associations*** specifically refer to the directed relations between concepts or sets of concepts.

An ***association rule*** is generally an expression of the form $A \Rightarrow B$, where A and B are sets of features.

An association rule $A \Rightarrow B$ indicates that transactions that involve A tend also to involve B.

Example

From the original market-basket problem, an association rule might be 25 percent of the transactions that contain milk also contain bread; 8 percent of all transactions contain both items.

In this example, 25 percent refers to the **confidence** level of the association rule, and 8 percent refers to the rule's level of **support**.

Distributions

1. Concept Selection
2. Concept Proportion
3. Conditional Concept Proportion
4. Concept Proportion Distribution
5. Conditional Concept Proportion Distribution
6. Average Concept Proportion
7. Average Concept Distribution

Pre-processing techniques

A few of the most common preprocessing techniques used in text mining are tokenization, term frequency, stemming and lemmatization.

1.Tokenization

2.Term frequency

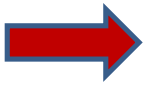
3.Stemming

4.Lemmatization

Stop word removal

Original text

- Information systems Asia web provides research, is related commercial materials, interaction, and even research sponsorship by interested corporations with a focus on Asia pacific region.
- Survey of information retrieval guide to ir, with an emphasis on web-based projects. Includes a glossary, and pointers to interesting papers.



After the stop-words removal

- Information systems Asia web provides research related commercial materials interaction research sponsorship interested corporations focus Asia pacific region
- Survey information retrieval guide ir emphasis web based projects includes glossary pointers interesting papers

Tokenization

Tokenization is the process of **breaking text up into separate tokens**, which can be individual words, phrases, or whole sentences.

In some cases, **punctuation** and **special characters** (symbols like %, &, $) are discarded in the process.

Python Implementation

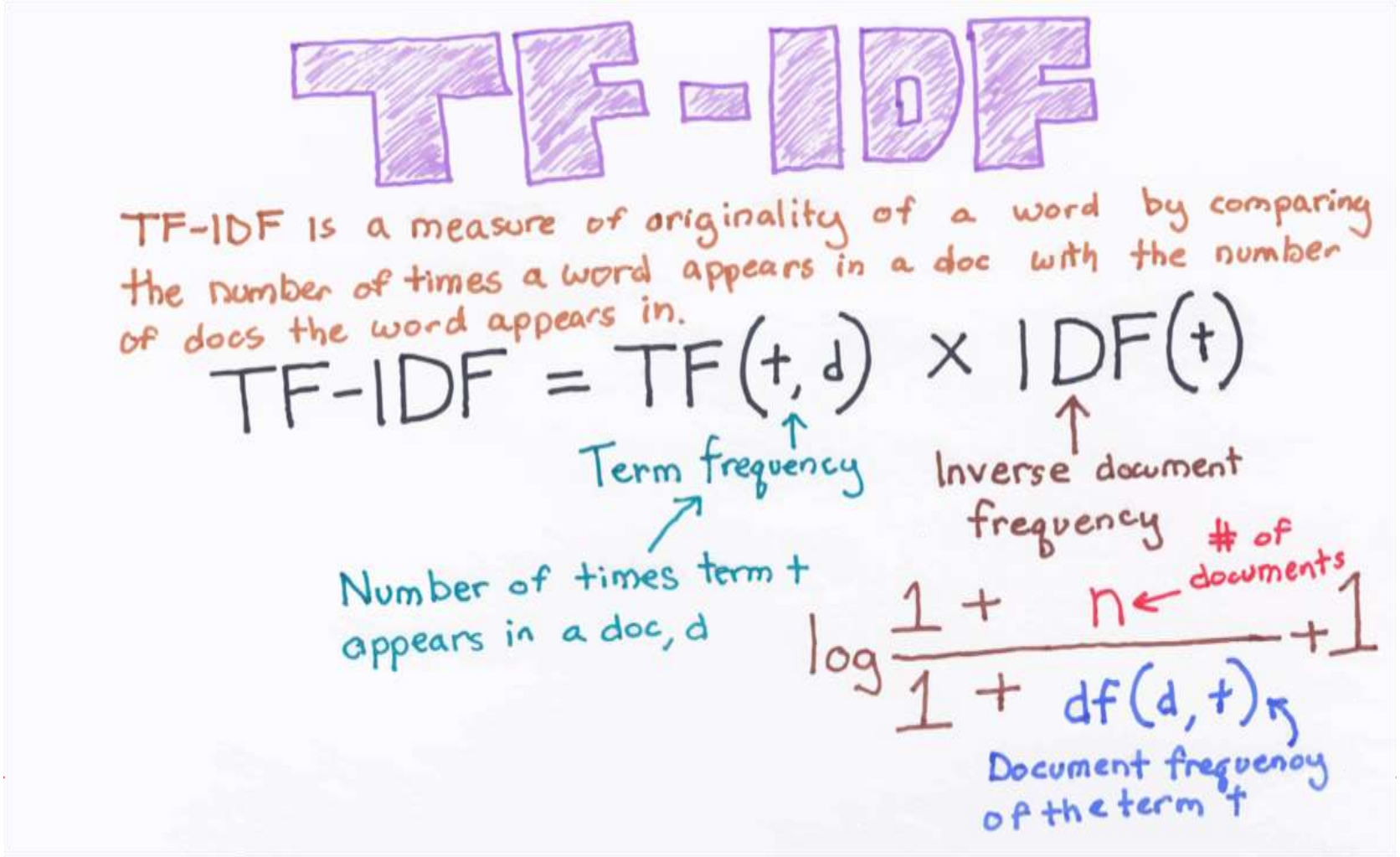
```
[1]: from nltk.tokenize import sent_tokenize

     sentence = "I`love`ice cream. I also like steak."
     sent_tokenize(sentence)

[1]: ['I love ice cream.', 'I also like steak.']
```

Term Frequency

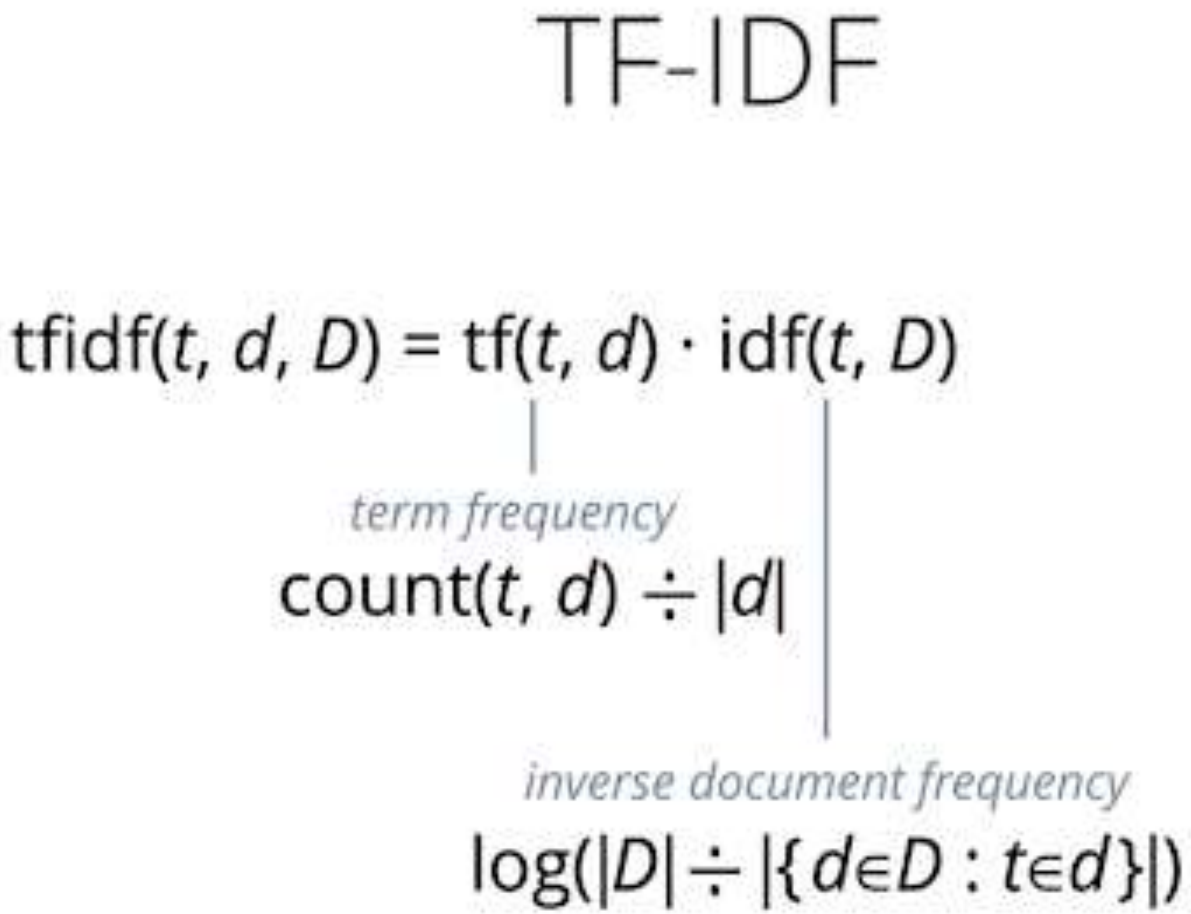
Term frequency tells you **how much a term occurs in a document**. Terms can be either individual words or phrases containing multiple words. Since documents differ in length, it's possible that a term would appear more times in longer documents than shorter ones. Thus, you can calculate term **frequency by dividing the number of times the term appears**, by the **total number of terms in the document**, as a way of normalization.

Term Frequency = [Number of times the term appears in the document] / [Total number of terms in the document]

TF-IDF


TF-IDF
TF-IDF is a measure of originality of a word by comparing the number of times a word appears in a doc with the number of docs the word appears in.
TF-IDF = TF(t, d) × IDF(t)
Term frequency
Number of times term t appears in a doc, d
Inverse document frequency
# of documents
log (1 + n) / (1 + df(d, t)) + 1
Document frequency of the term t

Cont ..


TF-IDF
tfidf(t, d, D) = tf(t, d) · idf(t, D)
term frequency
count(t, d) ÷ |d|
inverse document frequency
log(|D| ÷ |{d∈D : t∈d}|)

Python implementation for TF

```
In [94]: def computeTF(wordDict, bow):
             tfDict = {}
             bowCount = len(bow)
             for word, count in wordDict.items():
                 tfDict[word] = count/float(bowCount)
             return tfDict
```

```
Out[20]: {'on': 0.0,
          'The': 0.0,
          'road': 0.30102999566639812,
          'car': 0.30102999566639812,
          'truck': 0.30102999566639812,
          'driven': 0.0,
          'the': 0.0,
          'is': 0.0,
          'highway': 0.30102999566639812}
```

Stemming

Stemming is the process of reducing words to their **root** form.

For example, we would reduce the word ***robotics*** to the stem ***robot***.

For example, the **Porter stemmer**, a widely used algorithm for removing common suffixes from English words, reduces the words ***universal***, ***university***, and ***universe*** to the stem ***univers***.

Example


Consult
Consultant
Consulting
Consultantative
Consultants
Consulting
consult

Python Implementation -1

```
1 import nltk
2 from nltk.stem import PorterStemmer
```

```
1 words=['done','doing','studying','identify','this']
2 ps=PorterStemmer()
3 for word in words:
4     print(f"{word}: {ps.stem(word)}")
```

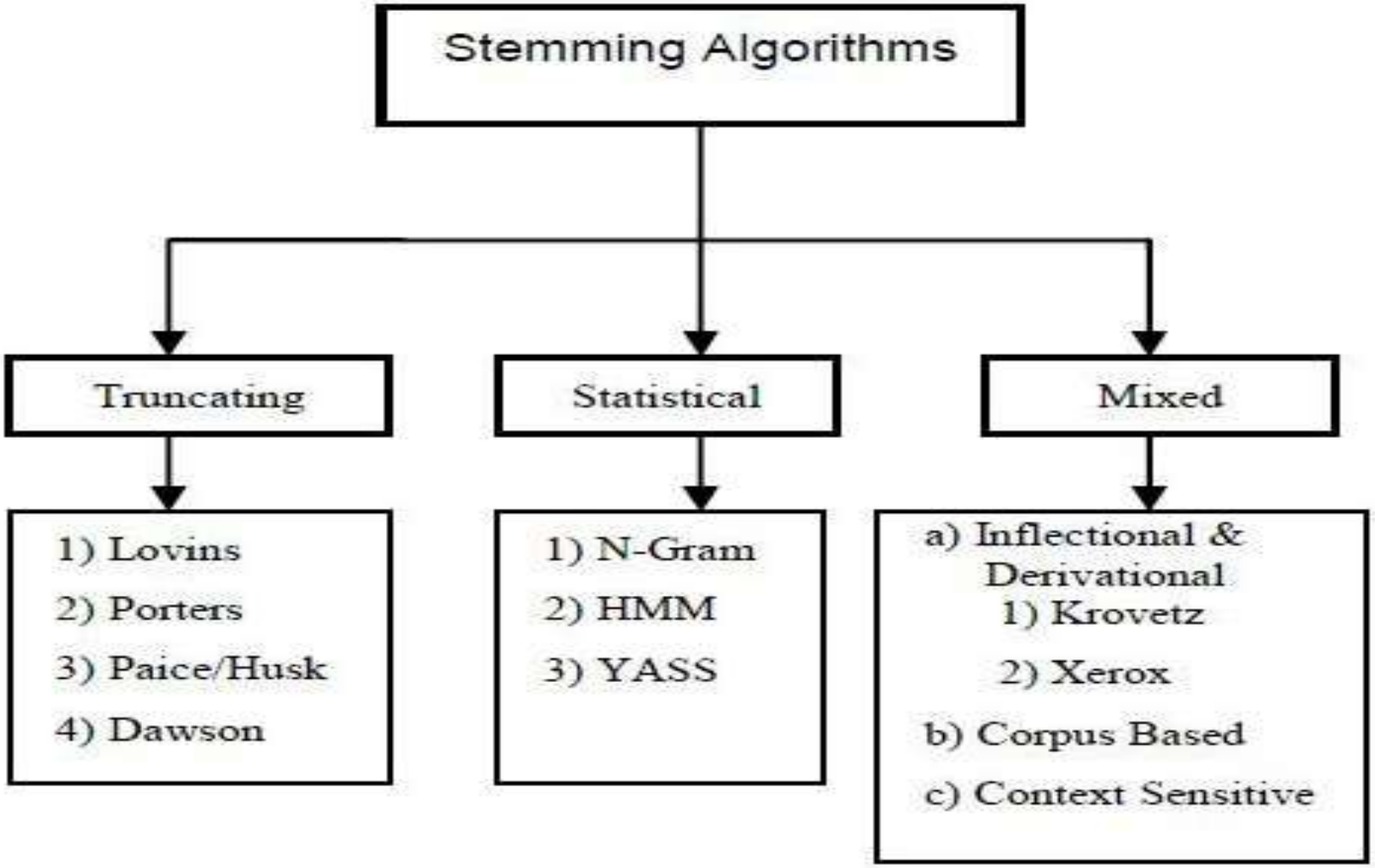
```
done: done
doing: do
studying: studi
identify: identifi
this: thi
```

Python Implementation -2

```
from nltk.stem import PorterStemmer

porter = PorterStemmer()

#Word-list for stemming :
word_list = ["Study","Studying","Studies","Studied"]

for w in word_list:
    print(porter.stem(w))
```

```
studi
studi
studi
studi
```

Types of Stemming Algorithms



Lemmatization

As we saw with the **Porter stemmer example**, the simple suffix rules that are commonly used for stemming could modify the stem.

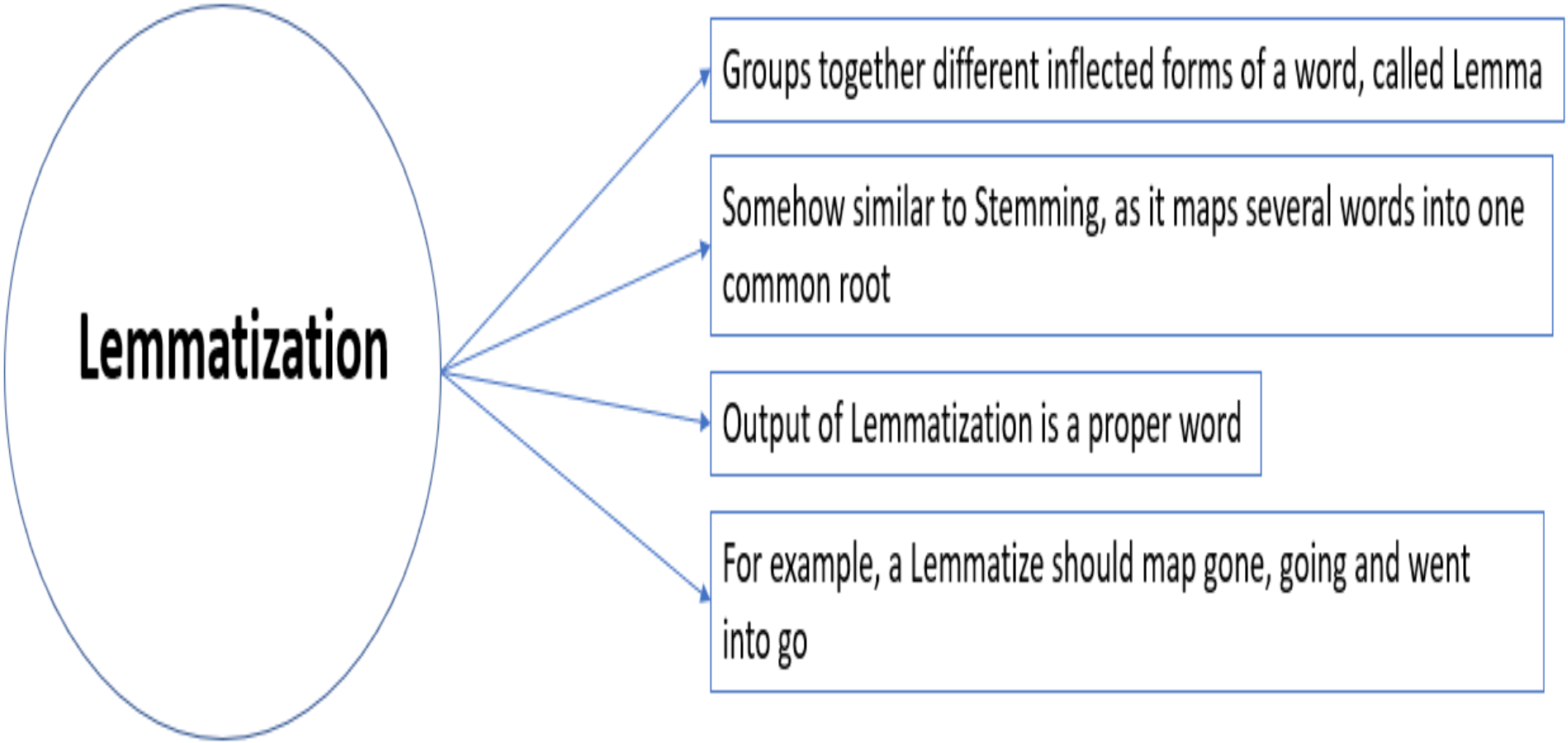
Lemmatization is a more complex approach to determining word stems, which addresses this potential problem. In **lemmatization, we use different normalization rules** depending on a word's lexical category (**part of speech**).

The stemmer can grasp more information about the word being stemmed, and use that to group similar words more accurately.

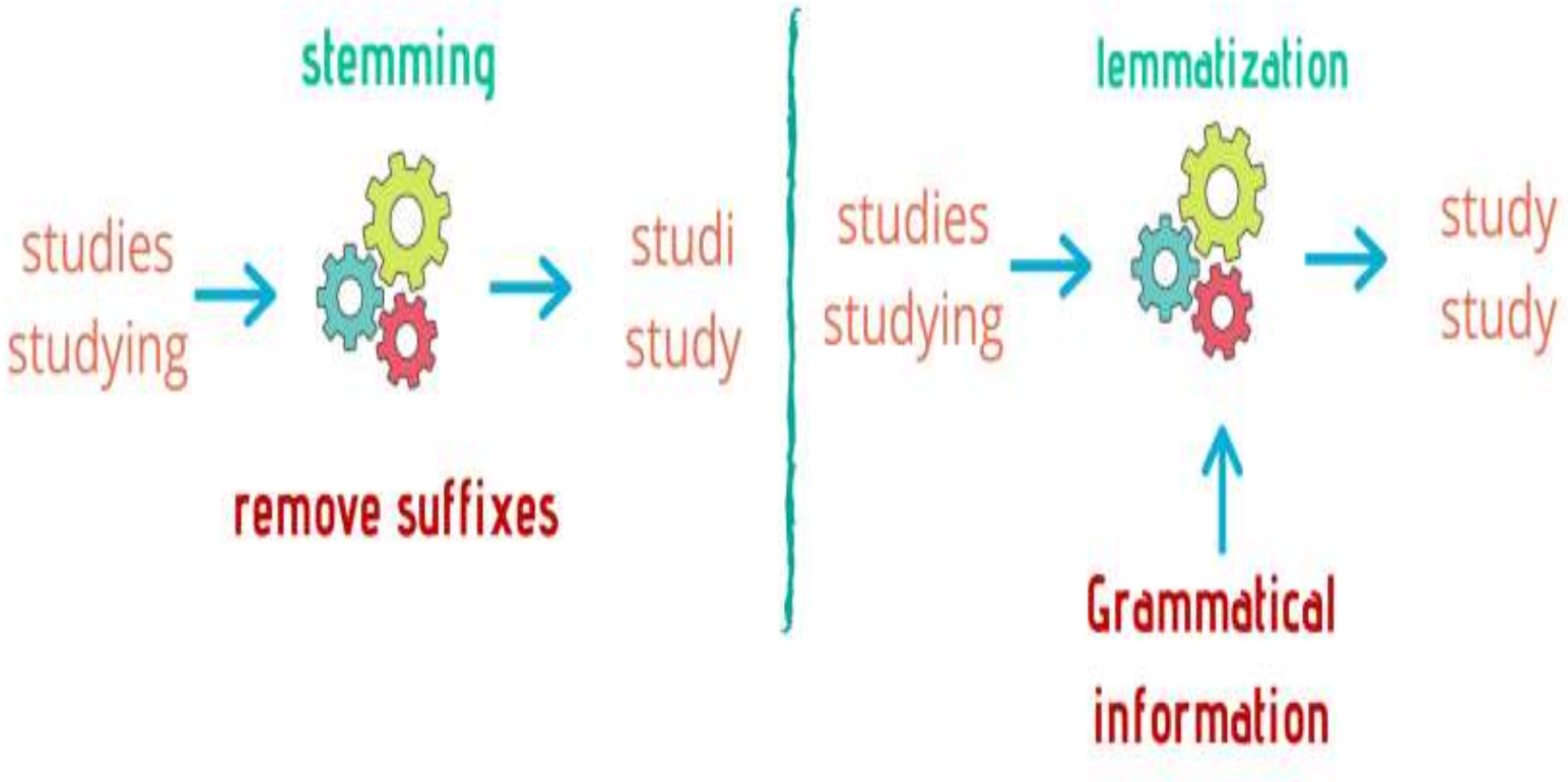
Example

Original Word	--->	Root Word (lemma)	Feature
meeting	--->	meet	(core-word extraction)
was	--->	be	(tense conversion to present tense)
mice	--->	mouse	(plural to singular)

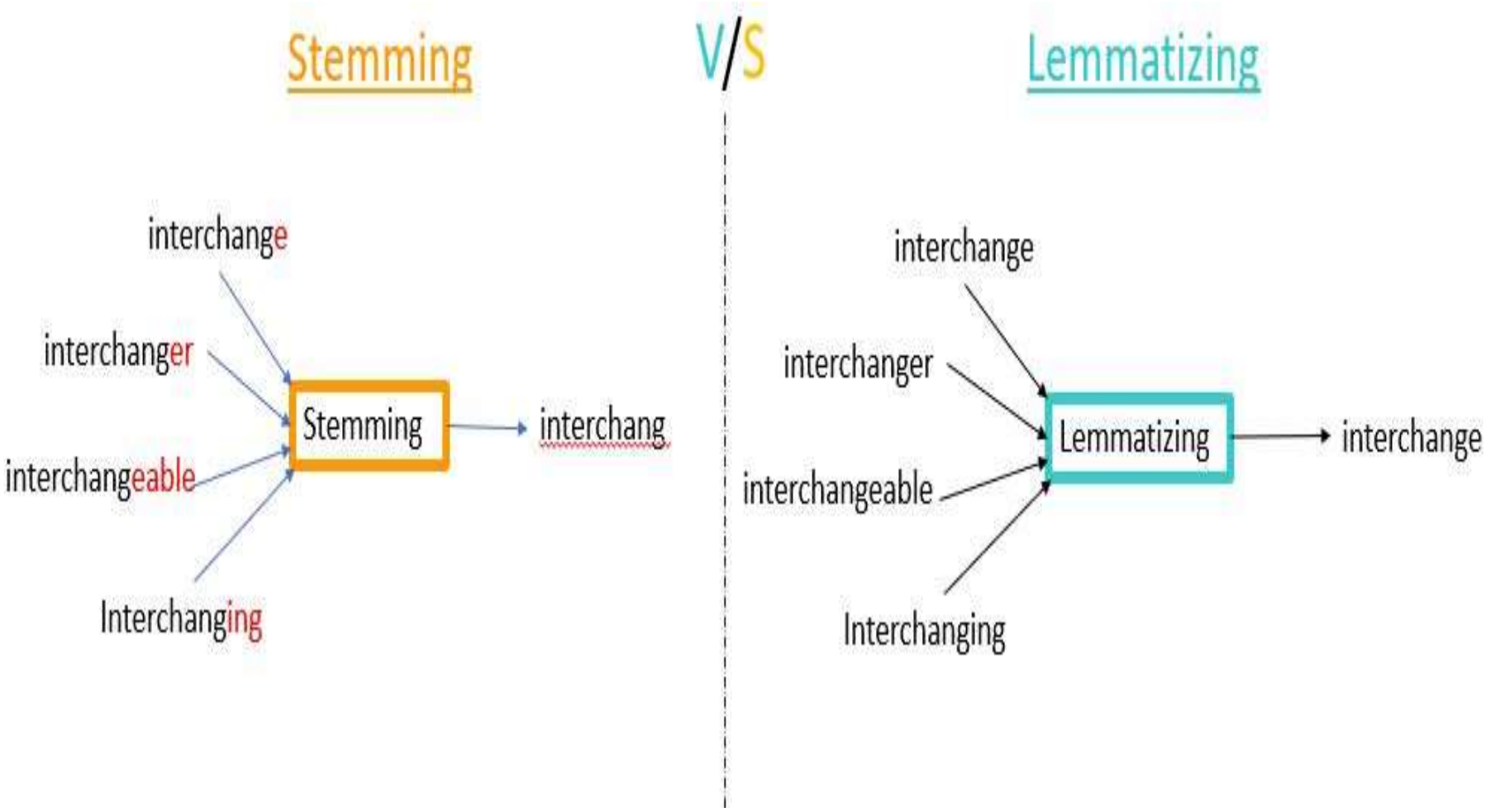
Cont ..


Lemmatization
Groups together different inflected forms of a word, called Lemma
Somehow similar to Stemming, as it maps several words into one common root
Output of Lemmatization is a proper word
For example, a Lemmatize should map gone, going and went into go

Stemming Vs Lemmatization



Cont ..


Stemming
V/S
Lemmatizing
interchange
interchanger
interchangeable
Interchanging
Stemming
interchang
interchange
interchanger
interchangeable
Interchanging
Lemmatizing
interchange

Approaches to Lemmatization

We will be going over **9 different approaches** to perform Lemmatization along with multiple examples and code implementations.

WordNet

WordNet (with POS tag)

TextBlob

TextBlob (with POS tag)

spaCy

TreeTagger

Pattern

Gensim

Stanford CoreNLP

Categorization

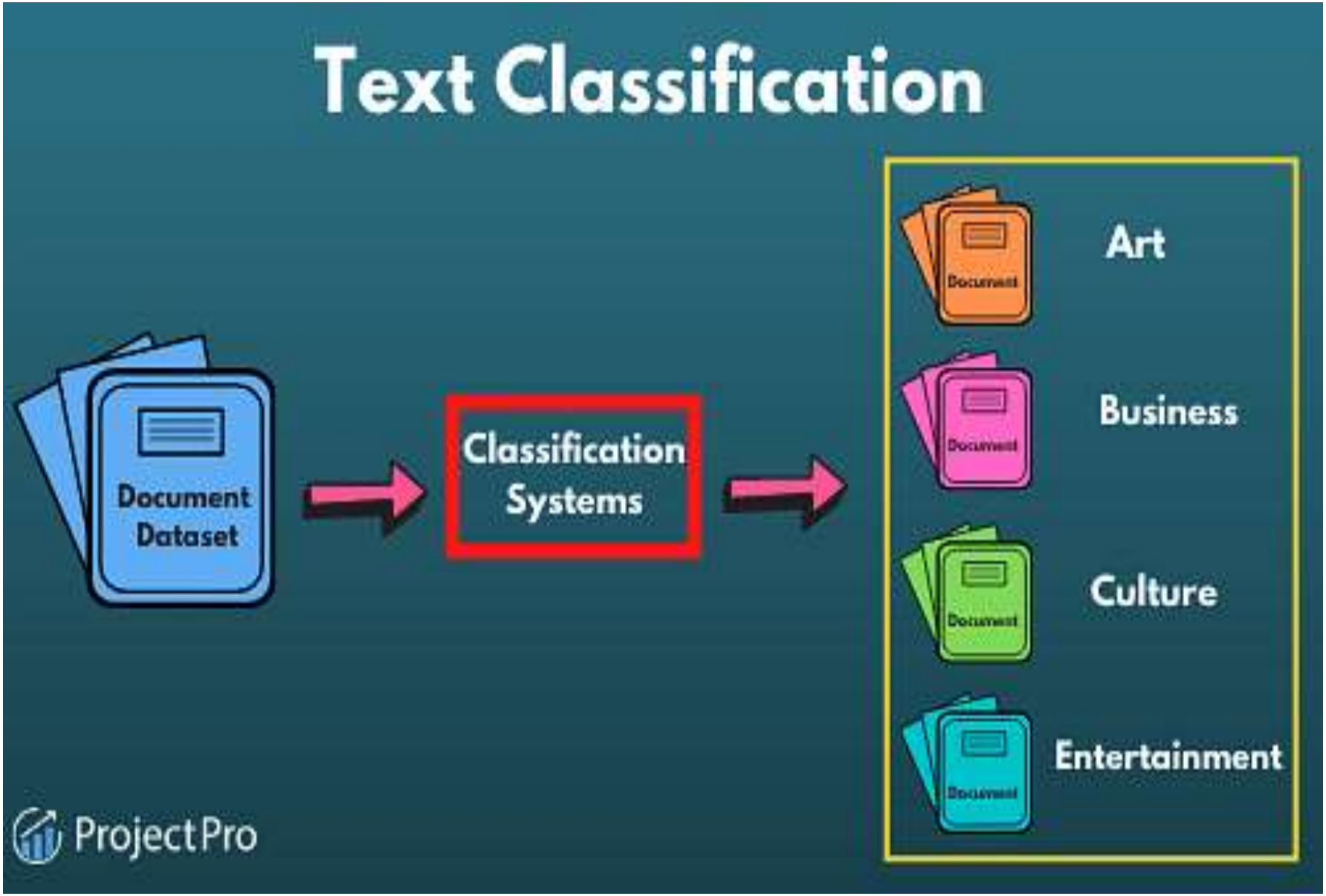
- Text categorization (TC) - given a set of categories (subjects, topics) and a collection of text documents, the process of finding the correct topic (or topics) for each document.

Given:

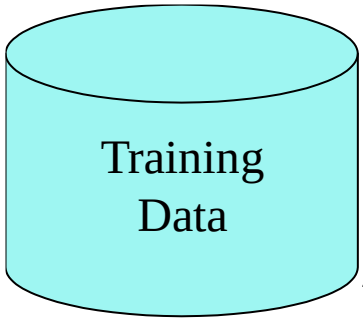
- A description of an instance, $x \in X$, where X is the instance language or instance space.
- A fixed set of categories: $C=\{c_1, c_2, \ldots c_n\}$

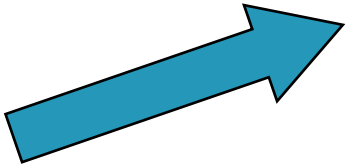
Determine:

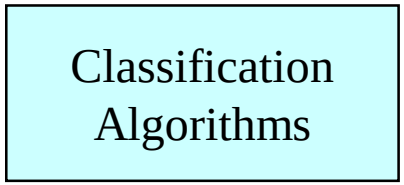
- The category of x: $c(x) \in C$, where $c(x)$ is a categorization function whose domain is X and whose range is C.

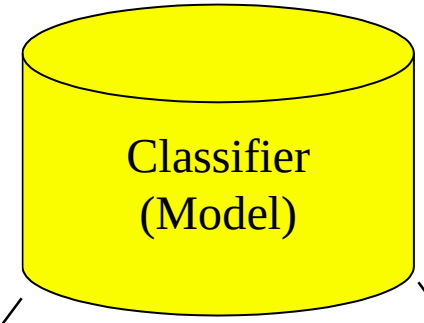

Text Classification
Document Dataset
Classification Systems
Document
Art
Document
Business
Document
Culture
Document
Entertainment
ProjectPro

Classification Process (1): Model Construction





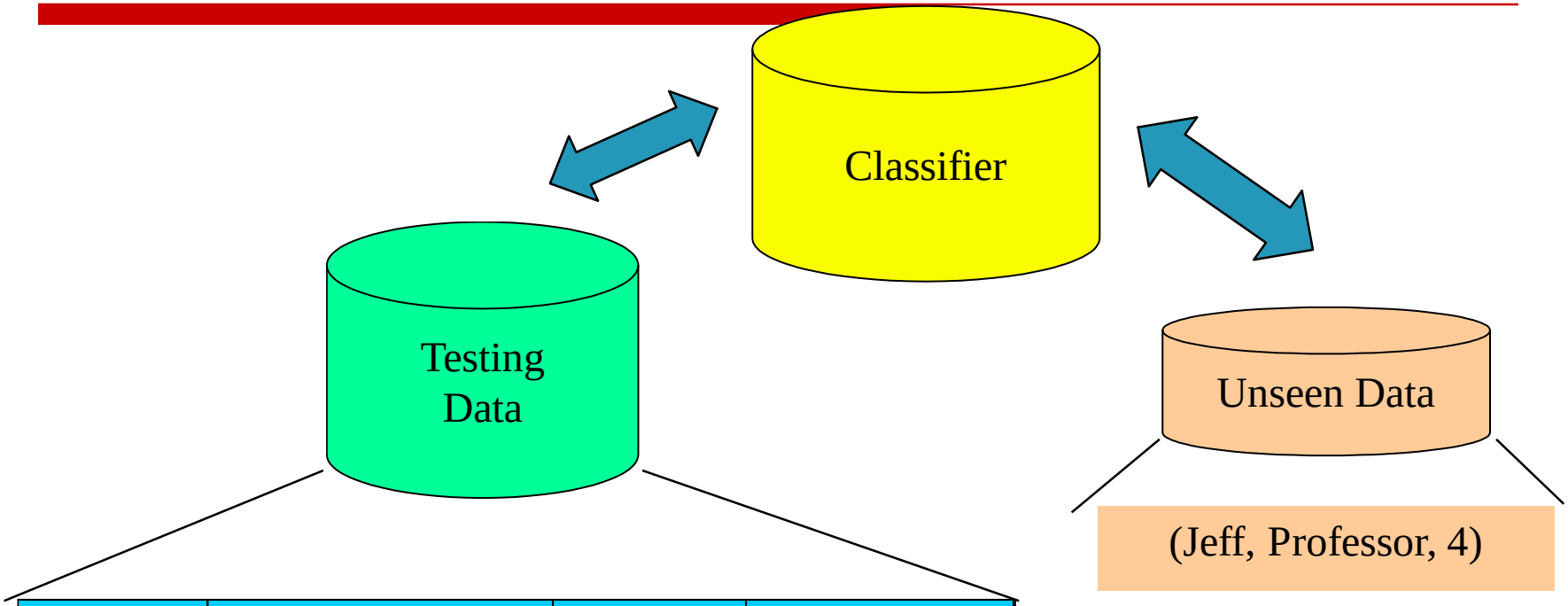




NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = ‘professor’
OR years > 6
THEN tenured = ‘yes’

Classification Process (2): Use the Model in Prediction



NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Tenured?

Yes

Text Categorization

- Pre-given categories and labeled document examples (Categories may form hierarchy)
- Classify new documents
- A standard classification (supervised learning) problem

Applications of Text Categorization

Text Indexing

Indexing of Texts Using Controlled Vocabulary

- The documents according to the user queries, which are based on the key terms. The key terms all belong to a finite set called controlled vocabulary.
- The task of assigning keywords from a controlled vocabulary to text documents is called text indexing.

Document Sorting and Text Filtering

Document Sorting

- Sorting the given collection of documents into several "bins."
- Examples:
  - In a newspaper, the classified ads may need to be categorized into "Personal," "Car Sale," "Real Estate," and so on.
  - E-mail coming into an organization, which may need to be sorted into categories such as "Complaints," "Deals," "Job applications," and others.

Text Filtering

- Text filtering activity can be seen as document sorting with only two bins – the "relevant" and "irrelevant" documents.
- Examples:
  - A sports related online magazine should filter out all non-sport stories it receives from the news feed.
  - An e-mail client should filter away spam.

Web page categorization

Hierarchical Web Page Categorization

- A common use of TC is the automatic classification of Web pages under the hierarchical catalogues posted by popular Internet portals such as Yahoo.
- constrains the number of documents belonging to a particular category to prevent the categories from becoming excessively large.
- Hyper textual nature of the documents is also another feature of the problem.

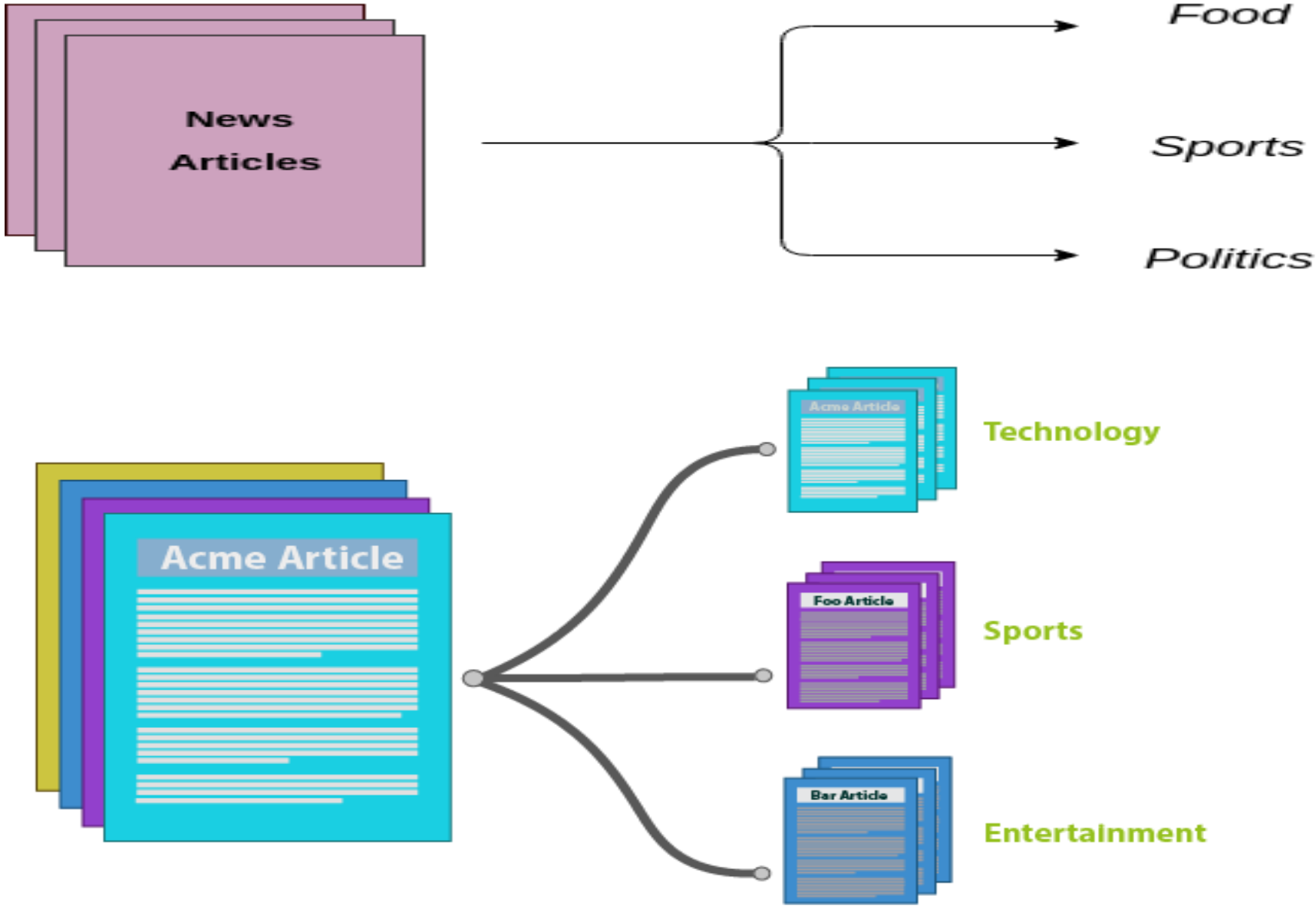
Definition of Text Categorization

- The task of approximating an unknown category assignment function $F:D \times C \rightarrow \{0,1\}$.

  Where, D - set of all possible documents and

  C - set of predefined categories.

- The value of F(d, c) is 1 if the document d belongs to the category c and 0 otherwise.
- The approximating function $M:D \times C \rightarrow \{0,1\}$ is called a classifier, and the task is to build a classifier that produces results as "close" as possible to the true category assignment function F.
- Classification Problems
  - Single-Label versus Multi-label Categorization
  - Document-Pivoted versus Category-Pivoted Categorization
  - Hard versus Soft Categorization


News
Articles
Food
Sports
Politics
Acme Article
Acme Article
Foo Article
Bar Article
Technology
Sports
Entertainment

Classification Problems

Single-Label versus Multi-label Categorization

- In single-label categorization, each document belongs to exactly one category.
- In multi-label categorization the categories overlap, and a document may belong to any number of categories.

Document-Pivoted versus Category-Pivoted Categorization

- For a given document, the classifier finds all categories to which the document belongs is called a document-pivoted categorization.
- Finding all documents that should be filed under a given category is called a category-pivoted categorization.

Hard versus Soft Categorization

- A particular label is explicitly assigned to the instance is called Soft/Ranking Categorization.
- A probability value is assigned to the test instance is known as Hard Categorization.

Categorization Methods

Manual: Typically rule-based

- Does not scale up (labor-intensive, rule inconsistency)
- May be appropriate for special data on a particular domain

Automatic: Typically exploiting machine learning techniques

- Vector space model based
  - ✓ Prototype-based (Rocchio)
  - ✓ K-nearest neighbor (KNN)
  - ✓ Decision-tree (learn rules)
  - ✓ Neural Networks (learn non-linear classifier)
  - ✓ Support Vector Machines (SVM)
- Probabilistic or generative model based
  - ✓ Naïve Bayes classifier

Document Representation

- The classifiers and learning algorithms cannot directly process the text documents in their original form.
- During a preprocessing step, the documents are converted into a more manageable representation.
- Typically, the documents are represented by feature vectors
  - A feature is simply an entity without internal structure - a dimension in the feature space.
  - A document is represented as a vector in this space - a sequence of features and their weights.

Vectorization

A way to **extract information** from **text in the form of word sequences**, we need a way to transform these **word sequences into numerical features**: this is **vectorization**.

The simplest **text vectorization technique** is **Bag Of Words** (**BOW**). It starts with a **list of words** called the **vocabulary** (this is often all the words that occur in the training data). Then, given an input text, it outputs a numerical vector which is simply the vector of word counts for each word of the vocabulary.

The Bag of Words Representation

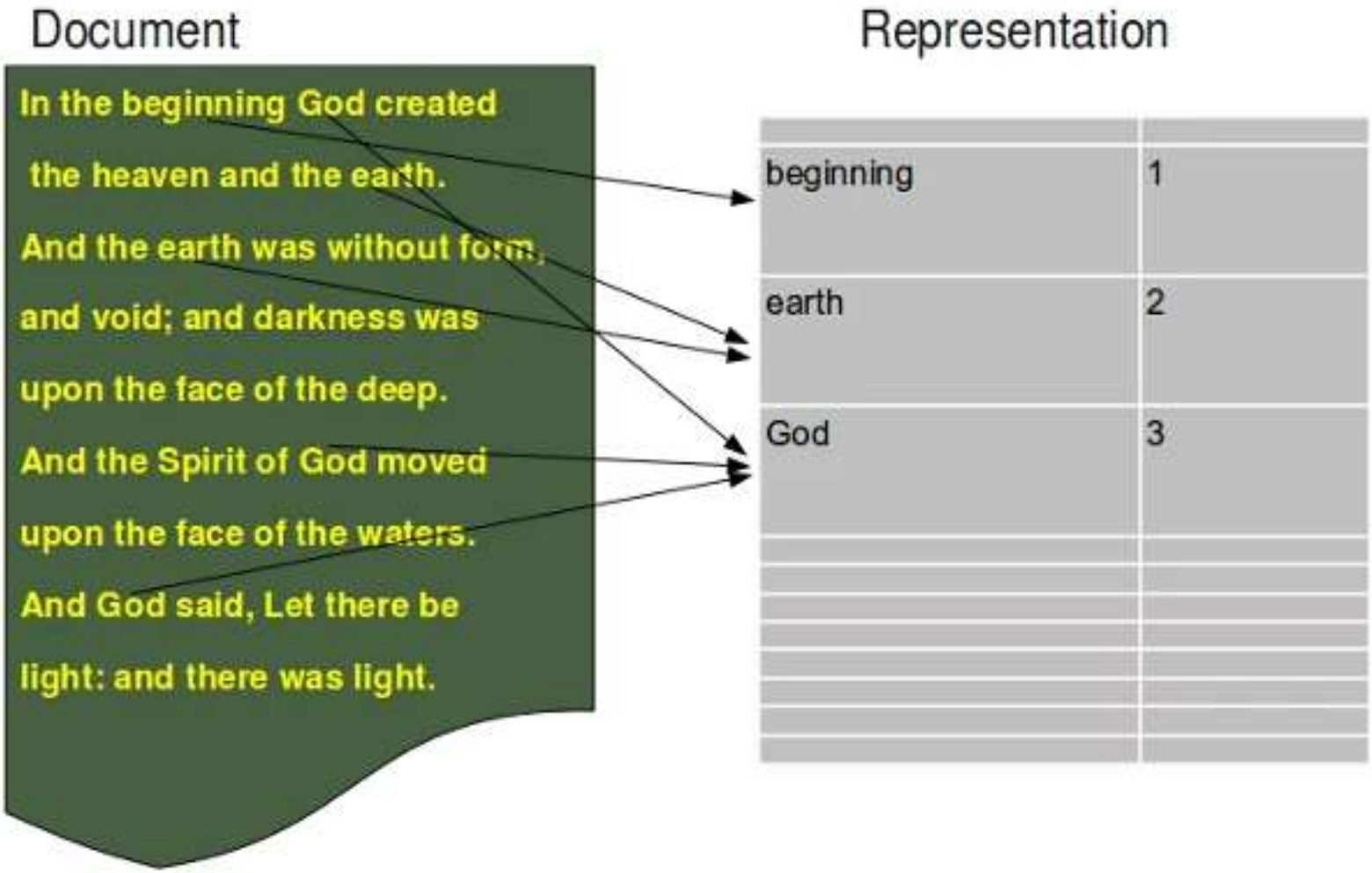
I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!

it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

Source

Document Representation

The document representation, which is based on the bag of word model, is illustrated in the following diagram:



Example

Training texts: ["This is a good cat", "This is a bad day"] => **vocabulary:** [this, **cat**, day, is, good, a, **bad**]

New text: "This day is a good day" -->

[1, 0, 2, 1, 1, 1, 0]

Example-2, if we have defined our dictionary to have the following words {*This*, *is*, *the*, *not*, *awesome*, *bad*, *basketball*}, and we wanted to vectorize the text *"This is awesome,"* we would have the following vector representation of that text: (1, 1, 0, 0, 1, 0, 0).

Common models for Doc Representation

- **Bag-of-words** model uses all words in a document as the features, and thus the dimension of the feature space is equal to the number of different words in all of the documents.
- **Binary** (simplest), in which the feature weight is either one - if the corresponding word is present in the document - or zero otherwise.
- The most common **TF-IDF scheme** gives the word *w* in the document *d* the weight,

$$\text{TF-IDF Weight}(w, d) = \text{TermFreq}(w, d) \cdot \log\left(N / \text{DocFreq}(w)\right),$$

Where,

TermFreq(w, d) - frequency of the word in the document,

N - number of all documents, and

DocFreq(w) - number of documents containing the word w.

Term Frequency

We can use **Term Frequency (TF)** instead of word counts and divide the number of occurrences by the sequence length.

We can also downscale these frequencies so that words that occur all the time (e.g., topic-related or stop words) have lower values. This downscaling factor is called **Inverse Document Frequency (IDF)** and is equal to the logarithm of the inverse word document frequency.

Put together, these new features are are called TF-IDF features. So in summary:

$$\text{TF}(\text{word}, \text{text}) = \frac{\text{number of times the word occurs in the text}}{\text{number of words in the text}}$$

$$\text{IDF}(\text{word}) = \mathbf{log}\left[\frac{\text{number of texts}}{\text{number of texts where the word occurs}}\right]$$

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

Feature Selection

Feature selection:

- Removes non-informative terms (irrelevant words) from documents.
- Improves classification effectiveness.
- Reduces computational complexity.

Measures of feature relevance:

- Relations between features and the categories.
- Feature Selection Methods
    - Document Frequency Threshold (DF)
    - Information Gain (IG)
    - χ2statistic (CHI)
    - Mutual Information (MI)

Dimensionality Reduction by Feature Extraction

- Feature reduction refers to the mapping of the original high-dimensional data onto a lower-dimensional space
- Criterion for feature reduction can be different based on different problem settings.

  - Unsupervised setting: minimize the information loss
  - Supervised setting: maximize the class discrimination

Feature Reduction Algorithms

- Unsupervised
  - Latent Semantic Indexing (LSI): truncated SVD
  - Independent Component Analysis (ICA)
  - Principal Component Analysis (PCA)
  - Manifold learning algorithms
- Supervised
  - Linear Discriminant Analysis (LDA)
  - Canonical Correlation Analysis (CCA)
  - Partial Least Squares (PLS)
- Semi-supervised

Knowledge Engineering Approach To Text Categorization

- Focus on manual development of classification rules.
- A set of sufficient conditions for a document to be labeled with a given category is defined.
- Example: CONSTRUE system

  if DNF (disjunction of conjunctive clauses) formula then category else ¬category
- Such rule may look like the following:

  If ((wheat & farm) or
  (wheat & commodity) or
  (bushels & export) or
  (wheat & tonnes) or
  (wheat & winter & ¬soft))
  then Wheat
  else ¬Wheat

Machine Learning Approach to Text Categorization

- In the ML approach, the classifier is built automatically by learning the properties of categories from a set of pre-classified training documents.
- Learning process is an instance of
  - Supervised Learning - process of applying the known true category assignment function on the training set.
  - The unsupervised version of the classification task, called clustering.
- Issues in using machine learning techniques to develop an application based on text categorization:
  - Choose how to decide on the categories that will be used to classify the instances.
  - Choose how to provide a training set for each of the categories.
  - Choose how to represent each of the instances on the features.
  - Choose a learning algorithm to be used for the categorization.

Approaches to Classifier Learning

- Probabilistic Classifiers
- Bayesian Logistic Regression
- Decision Tree Classifiers
- Decision Rule Classifiers
- Regression Methods
- The Rocchio Methods
- Neural Networks
- Example-Based Classifiers
- Support Vector Machines
- Classifier Committees: Bagging and Boosting

Clustering

Clustering: the process of grouping a set of objects into classes of similar objects

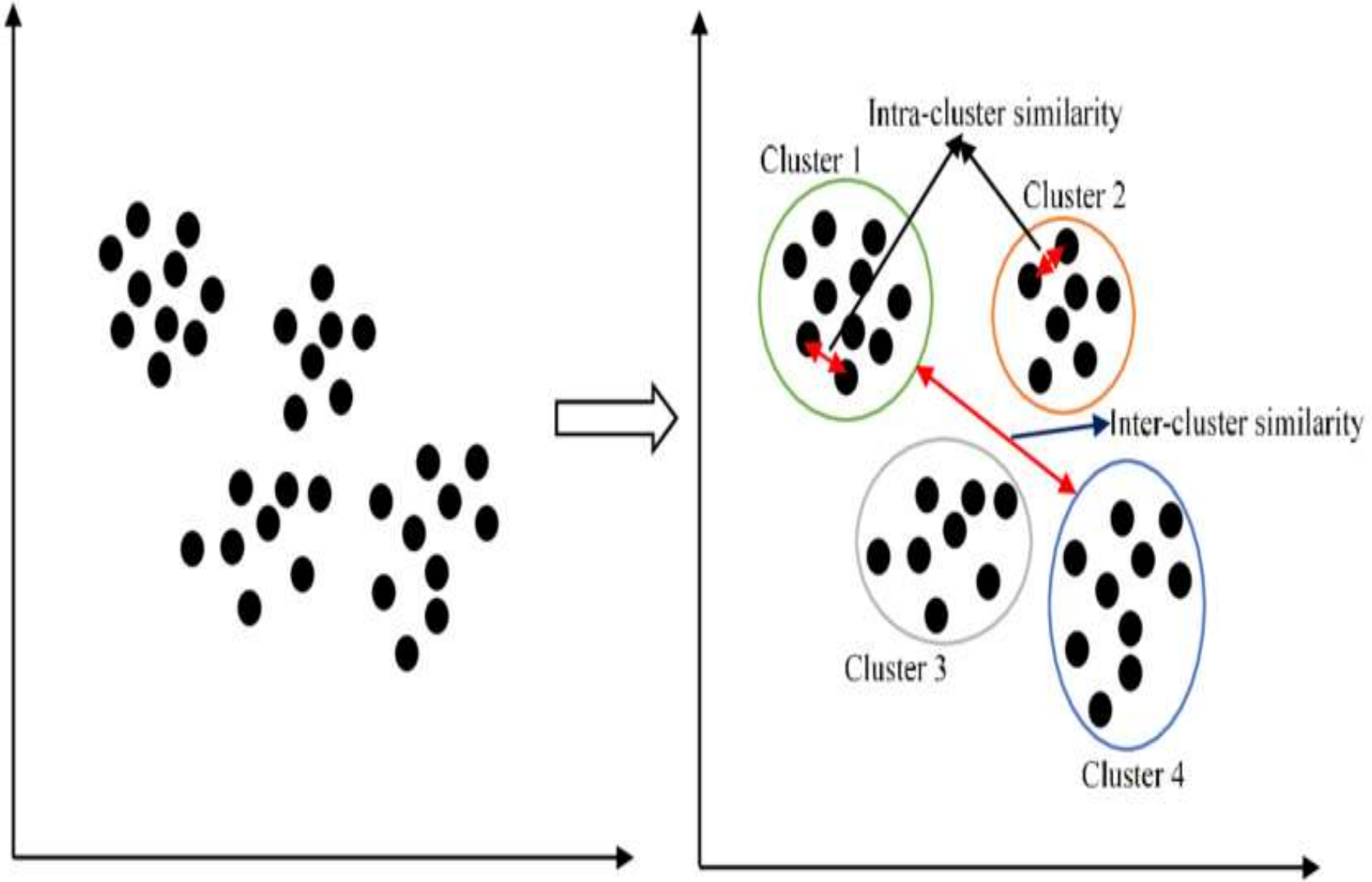
Documents within a cluster should be similar.

Documents from different clusters should be dissimilar.

The commonest form of *unsupervised learning*

Unsupervised learning = learning from raw data, as opposed to supervised data where a classification of examples is given

A common and important task that finds many applications in IR and other places



a. Data objects

b. Clustered data objects

Good Cluster

Internal criterion: A good clustering will produce high quality clusters in which:

- The **intra-class** (that is, intra-cluster) similarity is high.
- The **inter-class** similarity is low.
- The measured quality of a clustering depends on both the document representation and the similarity measure used

https://devopedia.org/text-clustering#qst-ans-1

https://www.kaggle.com/code/karthik3890/text-clustering#Introduction

Similarity Measures

The most popular metric is the usual Euclidean distance

$$D(\mathbf{x_i}, \mathbf{x_j}) = \sqrt{\sum_k (x_{ik} - x_{jk})^2},$$

which is a particular case with $p = 2$ of Minkowski metric

$$D_p(\mathbf{x_i}, \mathbf{x_j}) = \left(\sum_k (x_{ik} - x_{jk})^p \right)^{1/p}.$$

For the text documents clustering, however, the cosine similarity measure is the most common:

$$Sim(\mathbf{x_i}, \mathbf{x_j}) = (x'_i \cdot x'_j) = \sum_k x'_{ik} \cdot x'_{jk},$$

Use cases of clustering algorithms

1. Document Clustering
2. Recommendation Engine
3. Image Segmentation
4. Market Segmentation
5. Search Result Grouping

Document Clustering

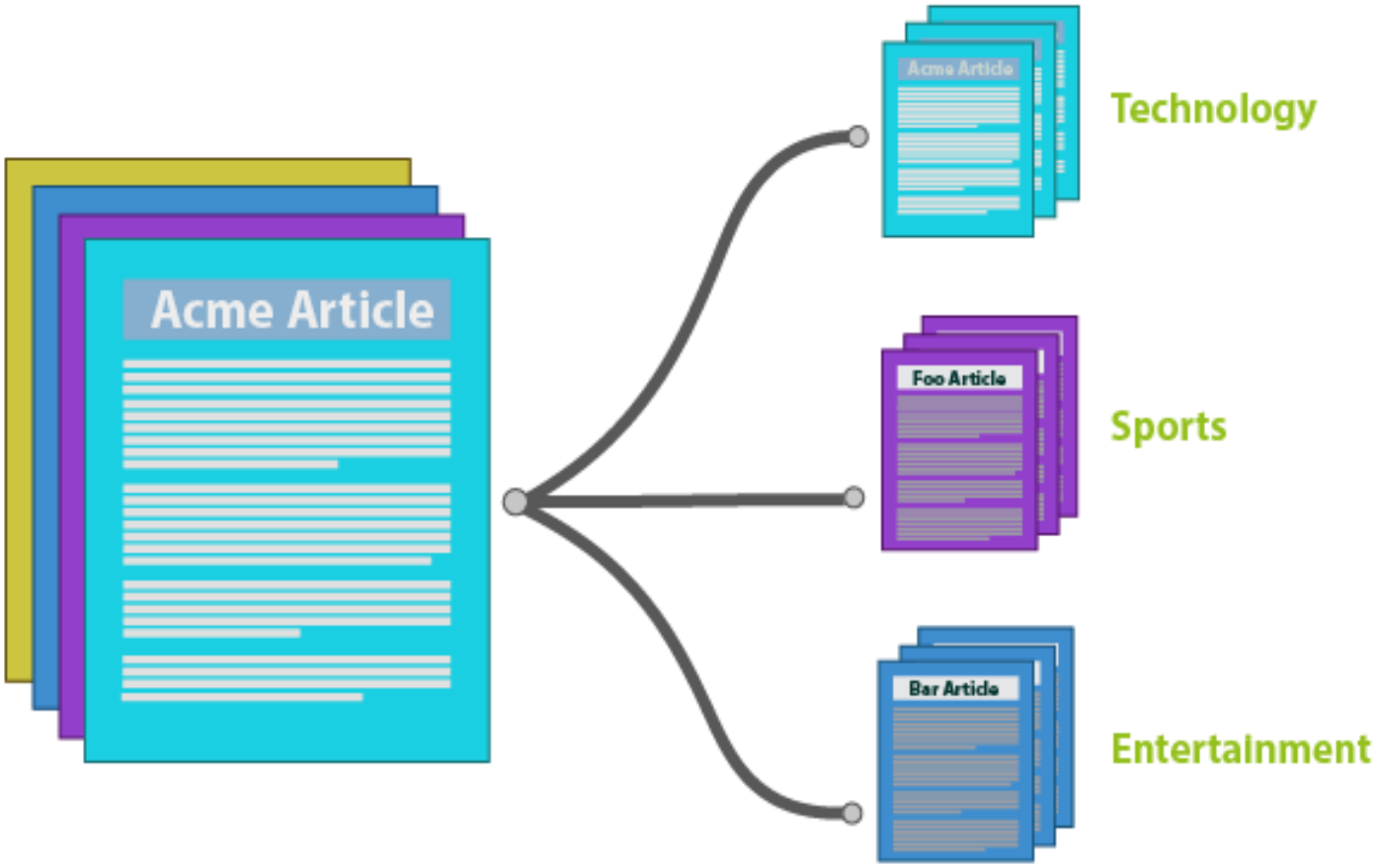
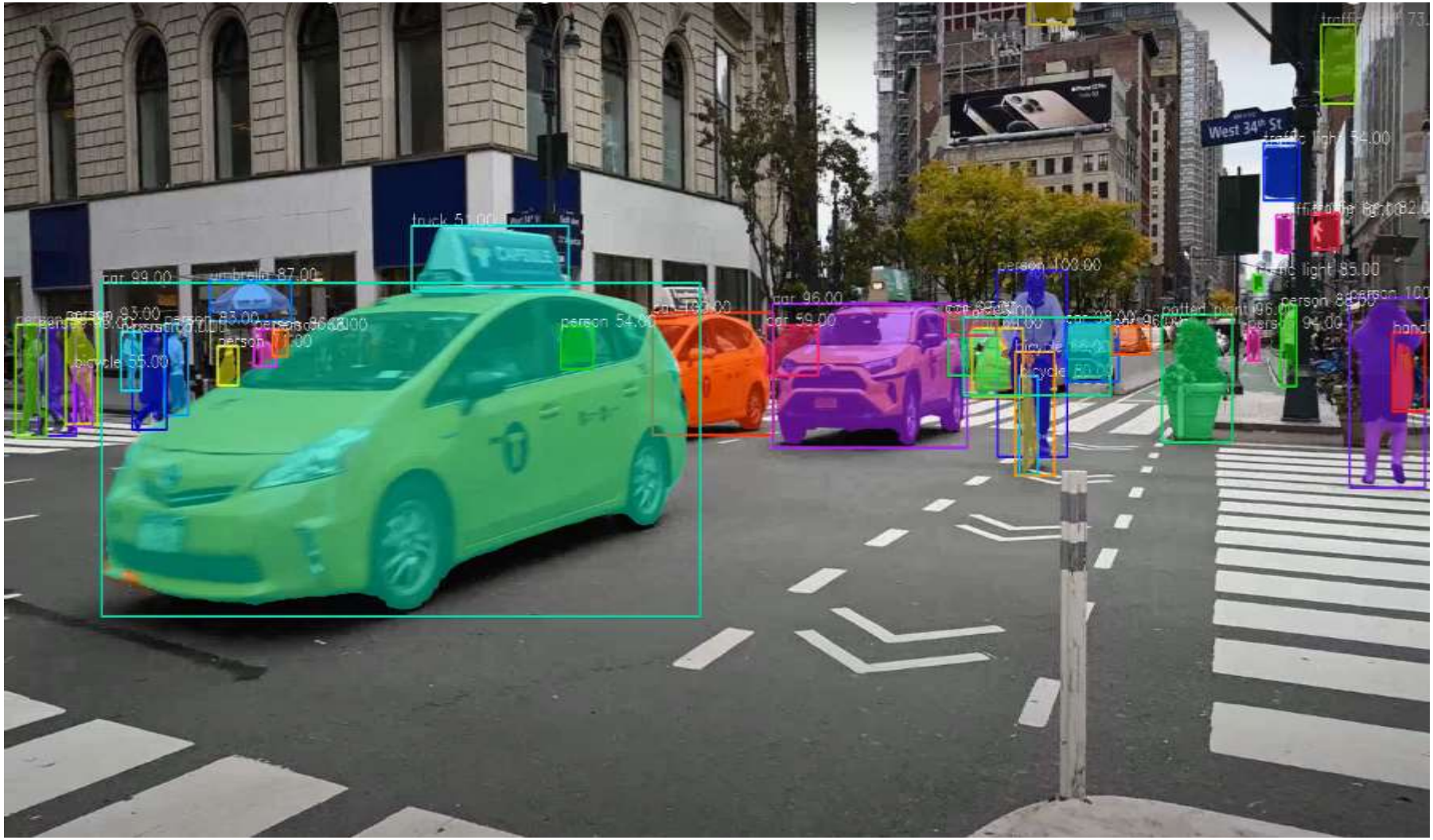


Image Segmentation



Types of Market Segmentation

Geographic Segmentation:

Consists of creating different groups of customers based on geographic boundaries.

Demographic Segmentation:

Consists of dividing the market through different variables such as age, gender, income, etc.

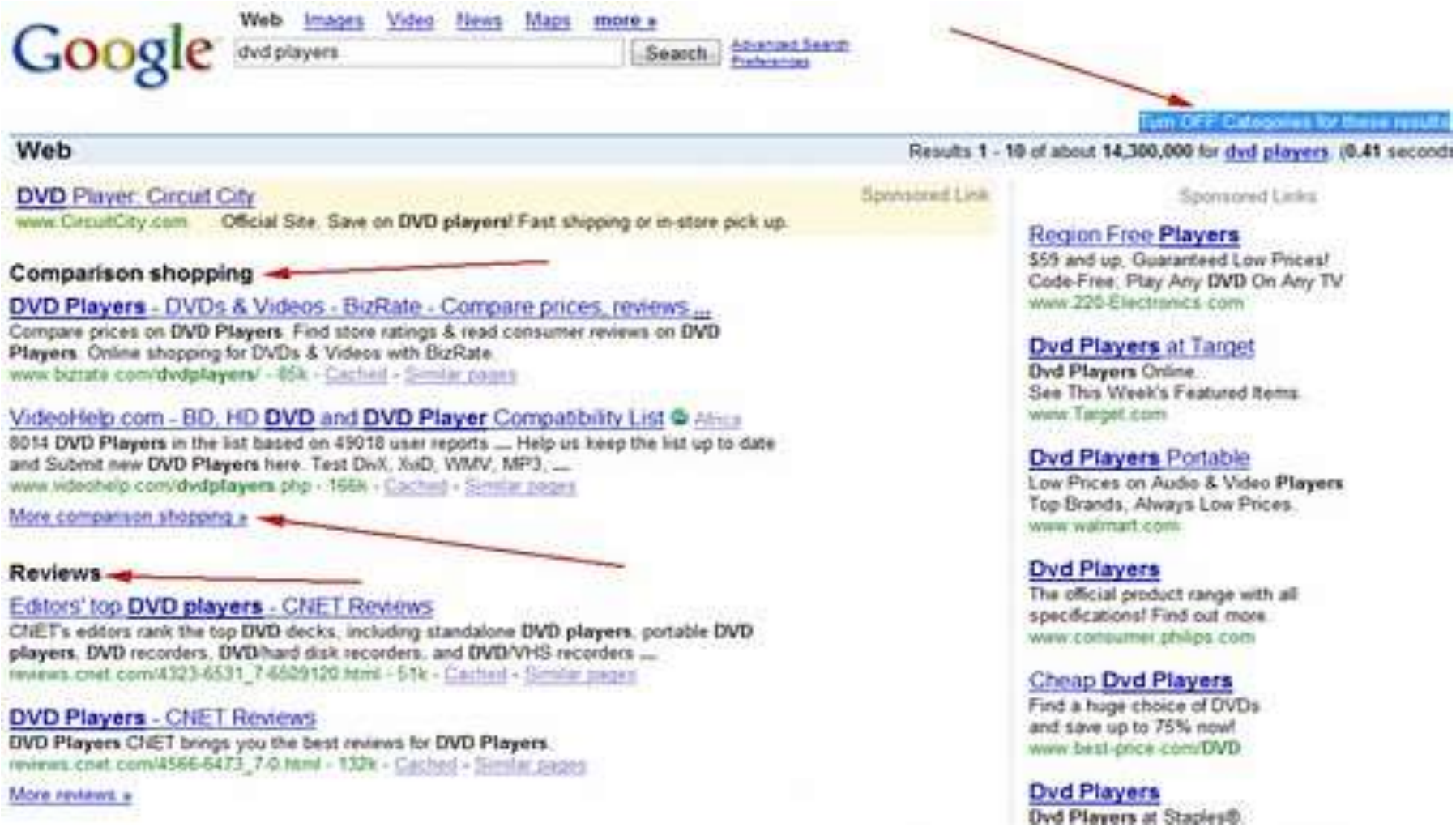
Psychographic Segmentation:

Consists of grouping the target audience based on their behavior, lifestyle, attitudes and interests.

Behavioral Segmentation:

Focuses on specific reactions and the way customers go through their purchasing processes.

Search Result Grouping



Types of Clustering Techniques

1. Partitioning Methods
2. Hierarchical Methods
3. Density-Based Methods
4. Grid-Based Methods
5. Model-Based Clustering Methods

Major Clustering Approaches

Partitioning algorithms: Construct various partitions and then evaluate them by some criterion

Hierarchy algorithms: Create a hierarchical decomposition of the set of data (or objects) using some criterion

Density-based: based on connectivity and density functions

Grid-based: based on a multiple-level granularity structure

Model-based: A model is hypothesized for each of the clusters and the idea is to find the best fit of that model to each other

Partitioning Algorithms: Basic Concept

Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters

Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion

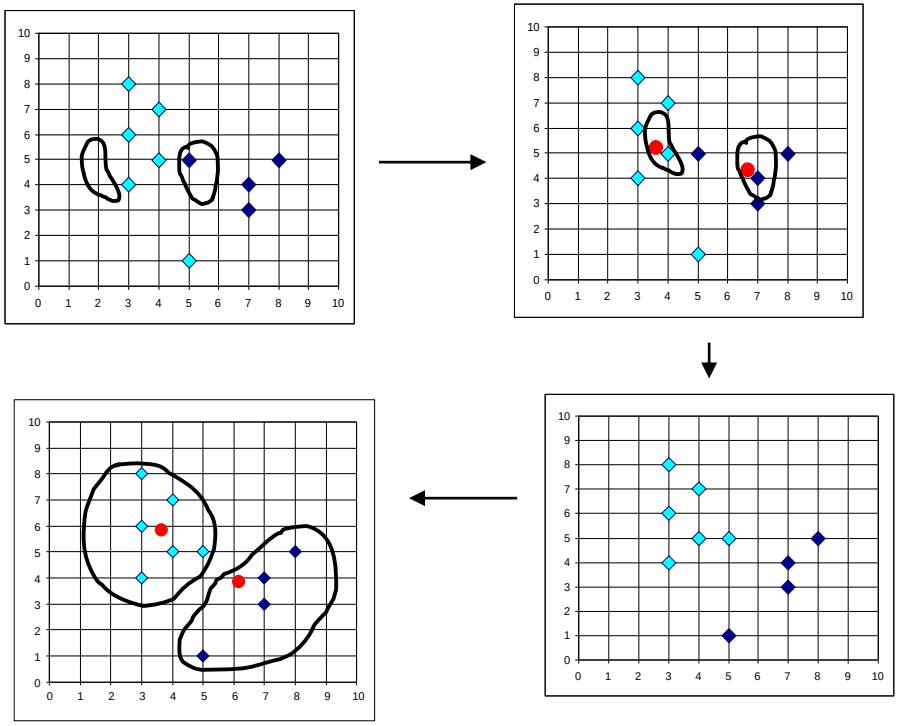
- Global optimal: exhaustively enumerate all partitions
- Heuristic methods: *k-means* and *k-medoids* algorithms
- *k-means* (MacQueen'67): Each cluster is represented by the center of the cluster
- *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

The *K-Means* Clustering Method

Given *k*, the *k-means* algorithm is implemented in 4 steps:

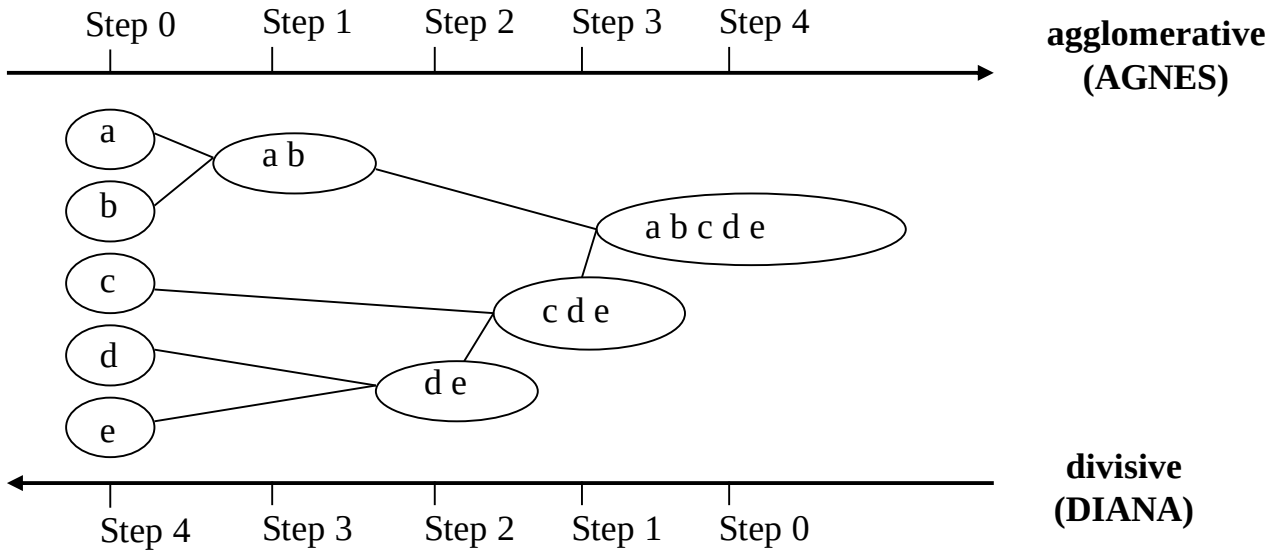
1. **Partition objects** into ***k*** nonempty subsets
2. **Compute seed points** as the **centroids** of the clusters of the current partition. The centroid is the center (**mean point**) of the cluster.
3. **Assign each object** to the cluster with the nearest seed point.
4. Go back to Step 2, **stop when no more new assignment.**

The *K-Means* Clustering Method



Hierarchical Clustering

Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



AGNES (Agglomerative Nesting)

Introduced in Kaufmann and Rousseeuw (1990)

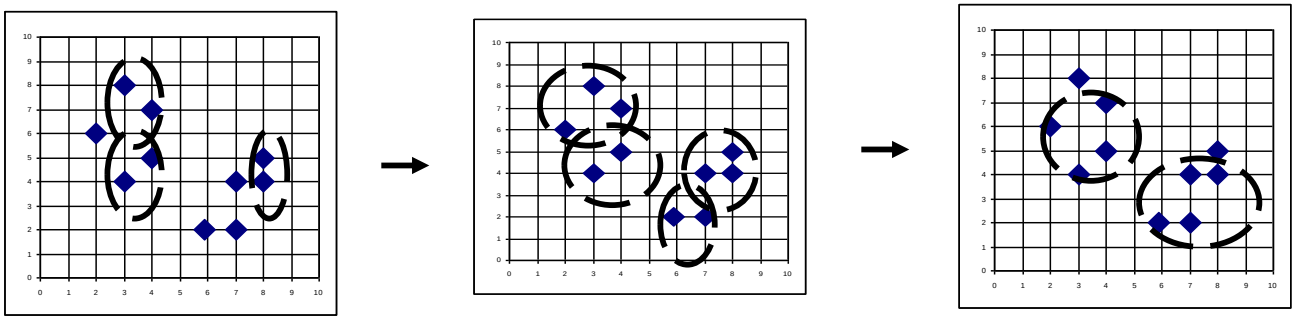
Implemented in statistical analysis packages, e.g., Splus

Use the Single-Link method and the dissimilarity matrix.

Merge nodes that have the least dissimilarity

Go on in a non-descending fashion

Eventually all nodes belong to the same cluster



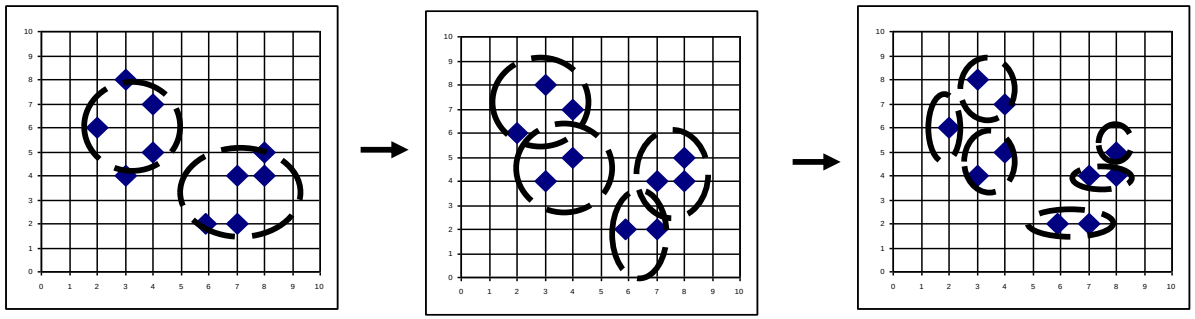
DIANA (Divisive Analysis)

Introduced in Kaufmann and Rousseeuw (1990)

Implemented in statistical analysis packages, e.g., Splus

Inverse order of AGNES

Eventually each node forms a cluster on its own



CHAMELEON

CHAMELEON: hierarchical clustering using dynamic modeling, by G. Karypis, E.H. Han and V. Kumar'99

Measures the similarity based on a dynamic model

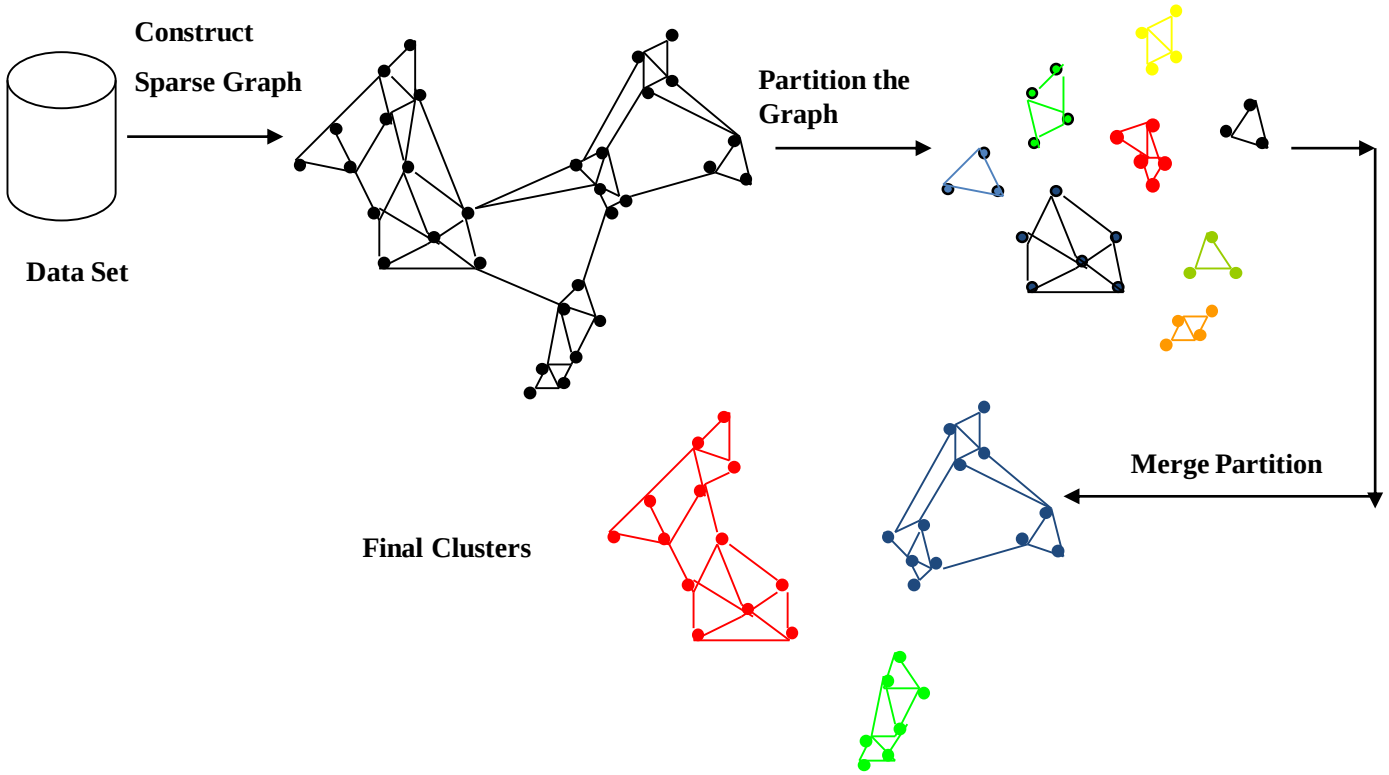
Two clusters are merged only if the *interconnectivity* and *closeness (proximity)* between two clusters are high *relative to* the internal interconnectivity of the clusters and closeness of items within the clusters

A two phase algorithm

1. Use a graph partitioning algorithm: **cluster objects into a large number of relatively small sub-clusters**

2. Use an agglomerative hierarchical clustering algorithm: **find the genuine clusters by repeatedly combining these sub-clusters**

Overall Framework of CHAMELEON



Density-Based Clustering Methods

Clustering based on density (local cluster criterion), such as density-connected points

Major features:

- Discover clusters of arbitrary shape
- Handle noise
- One scan
- Need density parameters as termination condition

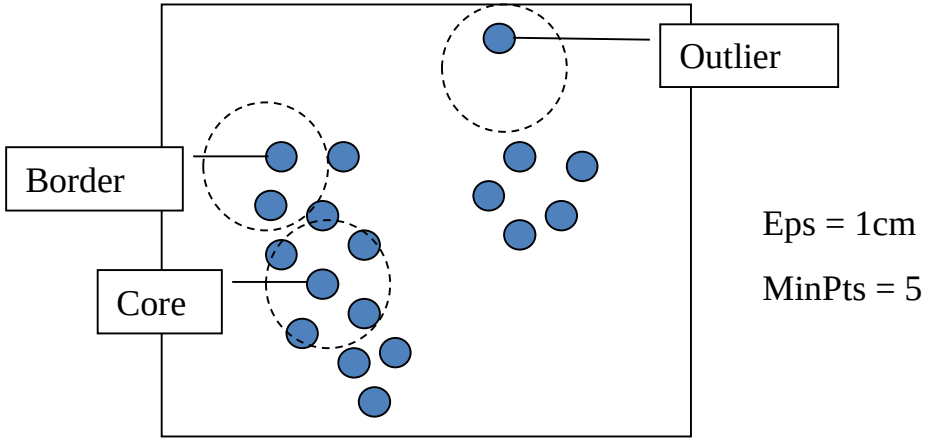
Several interesting studies:

- DBSCAN: Ester, et al. (KDD'96)
- OPTICS: Ankerst, et al (SIGMOD'99).
- DENCLUE: Hinneburg & D. Keim (KDD'98)
- CLIQUE: Agrawal, et al. (SIGMOD'98)

DBSCAN: Density Based Spatial Clustering of Applications with Noise

Relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points

Discovers clusters of arbitrary shape in spatial databases with noise



DBSCAN: The Algorithm

Arbitrary select a point ***p***

Retrieve all points density-reachable from ***p*** wrt ***Eps*** and ***MinPts***.

If ***p*** is a core point, a cluster is formed.

If ***p*** is a border point, no points are density-reachable from ***p*** and DBSCAN visits the next point of the database.

Continue the process until all of the points have been processed.

Grid-Based Clustering Method

Using multi-resolution grid data structure

Several interesting methods

- STING (a STatistical INformation Grid approach) by Wang, Yang and Muntz (1997)
- WaveCluster by Sheikholeslami, Chatterjee, and Zhang (VLDB'98)
  - A multi-resolution clustering approach using wavelet method
- CLIQUE: Agrawal, et al. (SIGMOD'98)

Model-Based Clustering Methods

- Attempt to optimize the fit between the data and some mathematical model
- Statistical and AI approach
  - Conceptual clustering
    - A form of clustering in machine learning
    - Produces a classification scheme for a set of unlabeled objects
    - Finds characteristic description for each concept (class)
  - COBWEB (Fisher'87)
    - A popular a simple method of incremental conceptual learning
    - Creates a hierarchical clustering in the form of a classification tree
    - Each node refers to a concept and contains a probabilistic description of that concept

Information extraction (IE)

The task of **Information Extraction (IE)** involves **extracting meaningful information from unstructured text data** and presenting it **in a structured format**.

Using **information extraction**, we can **retrieve pre-defined information** such as the **name of a person**, **location of an organization**, or **identify a relation between entities**, and save this information in a structured format such as a database.

Example

Let me show you another example I've taken from a cricket news article:

Indian captain Virat Kohli was dismissed cheaply for just 2 in Wellington on Friday by debutant Kyle Jamieson extending a rare lull in the batsman's stellar career. Throughout the ongoing New Zealand tour, Kohli has managed to score just a single fifty across 8 innings in all 3 international formats.

Cont ..

We can extract the following information from the text:

- Country – India, Captain – Virat Kohli
- Batsman – Virat Kohli, Runs – 2
- Bowler – Kyle Jamieson
- Match venue – Wellington
- Match series – New Zealand
- Series highlight – single fifty, 8 innings, 3 formats

This enables us to reap the benefits of powerful query tools like SQL for further analysis. Creating such structured data using information extraction will not only help us in analyzing the documents better but also help us in understanding the hidden relationships in the text.

How does Information Extraction work?

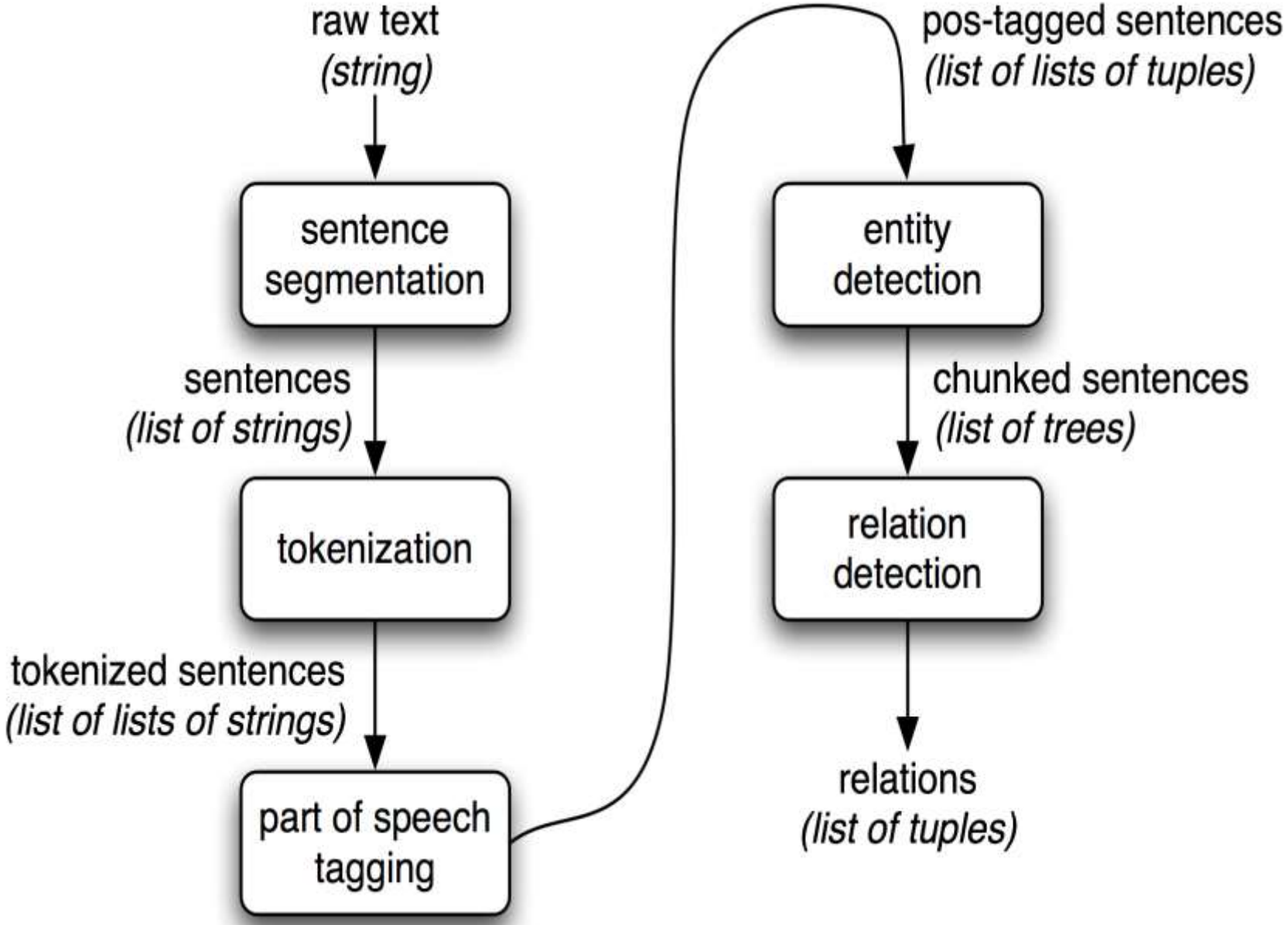
We all know that sentences are made up of words belonging to different **Parts of Speech (POS)**. There are eight different **POS** in the English language: **noun, pronoun, verb, adjective, adverb, preposition, conjunction,** and **intersection**.

The POS determines how a specific word functions in meaning in a given sentence. For example, take the word “right”. In the sentence, “**The boy was awarded chocolate for giving the right answer**”, **“right**” is used as an **adjective.** Whereas, in the sentence, “**You have the right to say whatever you want**”, **“right”** is treated as a **noun**.

Information Extraction Architecture

Shows the architecture for a simple information extraction system. first, the raw text of the **document is split into sentences using a sentence segmenter**, and each **sentence is further subdivided into words using a tokenizer.**

Next, each **sentence is tagged with part-of-speech tags**, which will prove very helpful in the next step, **named entity detection**. In this step, we search for mentions of potentially interesting entities in each sentence. Finally, we use **relation detection** to search for likely relations between different entities in the text.


raw text
(string)
sentence segmentation
sentences
(list of strings)
tokenization
tokenized sentences
(list of lists of strings)
part of speech tagging
pos-tagged sentences
(list of lists of tuples)
entity detection
chunked sentences
(list of trees)
relation detection
relations
(list of tuples)

https://www.nltk.org/book/ch07.html

Probabilistic models for information extraction

http://hanj.cs.illinois.edu/pdf/bkchap12_ysun.pdf

https://link.springer.com/chapter/10.1007/978-1-4614-3223-4_8

https://www.researchgate.net/publication/287910198_Probabilistic_Models_for_Text_Mining

Text Mining Applications

https://www.upgrad.com/blog/what-is-text-mining-techniques-and-applications/

https://www.expert.ai/blog/10-text-mining-examples/

https://www.analyticssteps.com/blogs/what-text-mining-process-methods-and-applications

https://www.repustate.com/blog/text-mining-applications/

https://www.promptcloud.com/blog/best-applications-of-text-mining-analysis/

Text Mining Applications

1. Spam Filtering
2. Customer Care Services
3. Prediction and Prevention of Crime
4. Risk Management
5. Knowledge Management
6. Fraud Detection by Insurance Companies
7. Personalized Advertising
8. Business Intelligence
9. Content Enrichment

Spam Filtering

E-mails are still considered as the most official way of communication in most organizations. But it has a dark side that has only increased in the twenty-first century – spam.

Out of every **ten emails in my mailbox**, at **least nine are spam**. Spams not only **fill up space** but also serve as an **entry point** for **viruses, scams**, and more.

Companies are pushing hard to filter more and more spam by using **intelligent text analytics** as compared to the keyword matching used earlier, to filter out more spam emails and give the user a healthier experience.

In case you run a business that can grow on **text scraping** and **text analytics**, remember that data is power, and before you decide how to harness data, make sure you consult someone who has already used data to their benefit or helped others do so.

Customer Care Services

Text mining and natural language processing are frequently being used in customer care services, be it over chat or voice call.

The " ***press one for recharge, press two for*** ….." format has been changed to the " ***say yes for account closure or no for cancellation …..*****"** format in many places to make the system appear more humane.

Most banks and e-commerce companies are using natural language processing-based **chatbots** that try to mimic a human customer care officer when talking to a customer. Improvement in customer care experience is taking place as these bots are using the information on the customer that they are interacting with, to make the experience more customized. **By automating customer care services, companies are providing customers a better experience while at the same time, saving money**.

Prediction and Prevention of Crime

“Prevention is better than Cure”

So what if you could prevent crimes by knowing beforehand, where and when they might take place. Since the internet is anonymous and so is most of the communication software that operates via it, most criminals plan and communicate using these methods. However, you can understand that **millions of normal people use these means of communication as well, and it is a difficult task to pinpoint messages that might be considered a threat.**

This is easily done using advanced text analysis software that scans communication sources in real-time and sounds different levels of threat alert on finding different types of text. Law enforcement across the world have been using **these technologies to prevent terrorist attacks, catch sleeper cells, and stop people from carrying out other unlawful activities**.

Risk Management

Many finance players including **banks**, **microfinance institutions**, and others, are now **depending on risk management software** that can go through **documents** and **profiles** to decide on **investment risks**, **credit scores**, and more.

The text mining technologies used by such high-end software absorb petabytes of data and present information in a consumable format.

This helps in **risk moderation.** Such software is helping financial institutions all over the world, to decrease their percentage of non-performing assets.

Knowledge Management

In many industries like the **healthcare industry**, **managing a huge amount of textual information** has become a problem. The amount of information gathered every single hour is huge. All this data has to be stored in such a manner that the information can be retrieved as and when required. It may so happen, that there is an epidemic and hospitals need to coordinate to go through all their **data to pinpoint the source or the first infected person.**

Such a huge exercise would be impossible without the help of proper text analytics systems in place that would manage the data and information and keep them in **a structured tree-like format**. This would lead to people being able to access the data in any way they need- region-based, gender-based, disease-based, and more. The inability to find important information quickly may cripple such organizations dealing with large volumes of text documents.

Fraud Detection by Insurance Companies

With rising cases of insurance fraud, text analytics has proved effective in going over **huge collections of case files to understand the chances of an insurance claim being a fraud**. It greatly reduces the workload of the company officials since the fraud recognition software would automatically flag cases where a high probability of fraud is determined.

Insurance companies are tying up with technology giants to take full advantage of the developments in text mining technologies, and combine their results to produce structured data to prevent frauds and swiftly process claims.

Personalized Advertising

Remember how you saw ads of the same mobile phone on Facebook that you were viewing on Amazon? No that is not a coincidence.

Digital advertising has been revolutionized by text and web data mining.

Text data related to all that you type, view, or do online is stored by technology giants, or sold to other companies to show you advertisements that you have a higher probability of clicking on, and which have a higher probability of getting converted into a sale. This is one of the latest and most widely used applications of text analytics and mining.

Business Intelligence

Decision-making is difficult. It is even more difficult when you have to answer to your shareholders as to why you took the decision and how you think that the decision will positively impact the company.

Text mining helps gather evidence and draw up charts and graphs to put the information to back your gut feeling.

Only relevant information and data are extracted so that the people who lead can make the best decisions by going through only a few pages of information.

Content analysis

Content analysis is a research tool used to determine the **presence of certain words** or **concepts** within **texts** or **sets of texts**.

Researchers quantify and analyze the presence, meanings and relationships of such words and concepts, then make inferences about the messages within the texts, the writer(s), the audience, and even the culture and time of which these are a part.

Texts can be defined broadly as books, book chapters, essays, interviews, discussions, newspaper headlines and articles, historical documents, speeches, conversations, advertising, theater, informal conversation, or really any occurrence of communicative language.

Types of Content Analysis

There are two general types of content analysis:

1. **Conceptual** or **quantitative** analysis
2. **Relational** or **qualitative** analysis.

Conceptual analysis determines the existence and frequency of concepts in a text.

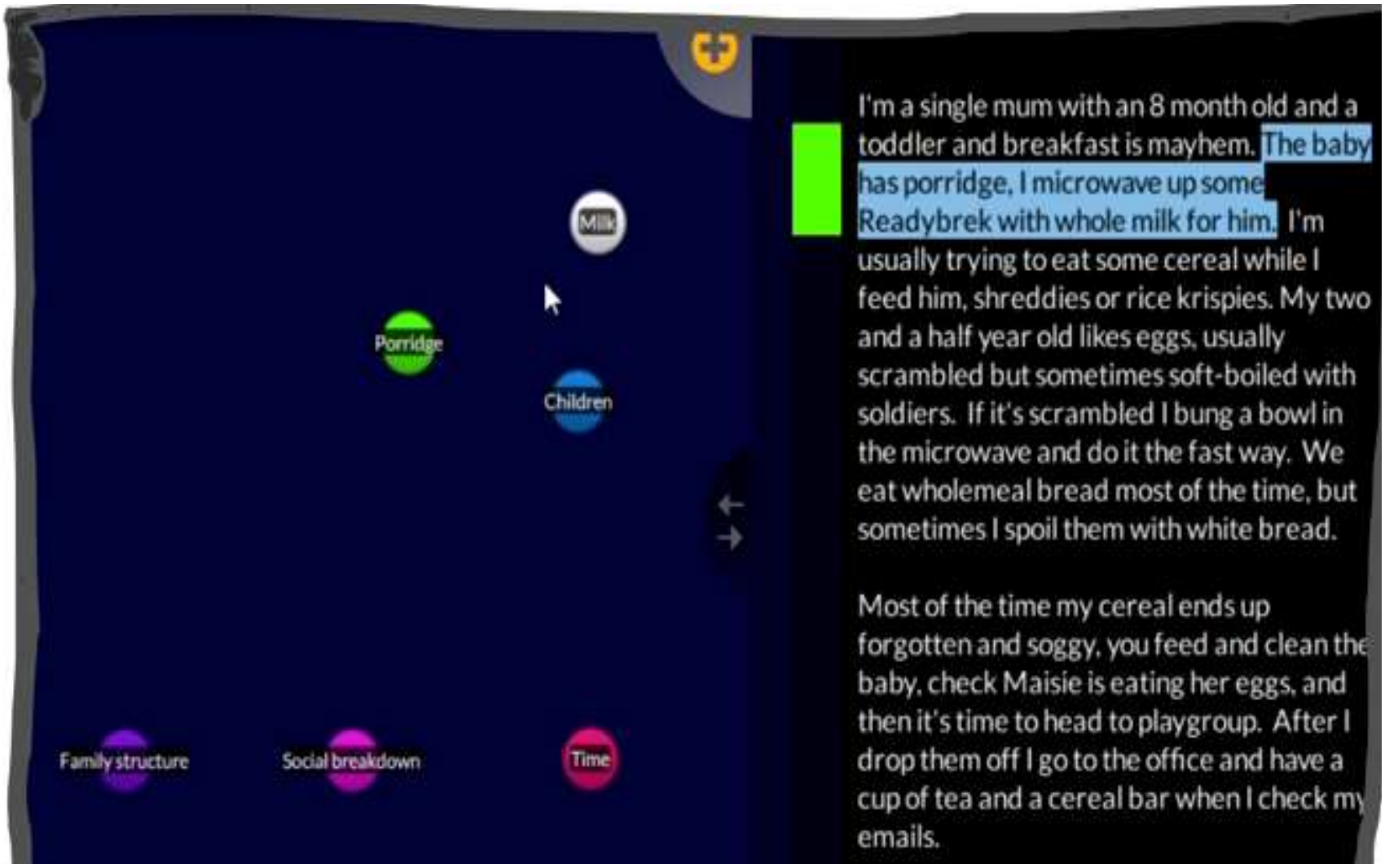
Relational analysis develops the conceptual analysis further by examining the relationships among concepts in a text.

Each type of analysis may lead to different results, conclusions, interpretations and meanings.

Steps in Content Analysis

- ❑ Choose data sources
- ❑ Code data
- ❑ Develop categories
- ❑ Assess validity and reliability
- ❑ Analyse results

Example



https://www.youtube.com/watch?v=dxxES6YYwMs


Thank
you