

# Text Mining

---

Quite different from previous data mining tasks where variables or features are numeric, categorical and well-defined. Text mining can be hard, but also exciting and extremely useful

## Sample applications

- Spam filtering
- Natural Language Processing,
- Sentiment Analysis
- Basis for chat bot
- Law (previous case studies)
- Finance (market sentiments)
- Fraud and Deception detection

## Numerous Data Sources :

- Tweets
- Blogs
- Books
- News Feeds

# Data Mining versus Text Mining

---

- Both seek for novel and useful patterns
  - Both are semi-automated processes
- Difference is the nature of the data:
  - Data Mining works on structured data stored in databases
  - Text Mining works on unstructured data in Word documents, PDF files, XML files, etc
- Text mining – first, impose structure to the data, then mine the structured data

# Text Mining Fundamental Concepts

---

## Text mining Objective

A semi-automated process of extracting knowledge from unstructured data sources i.e. knowledge discovery in textual databases

## Structuring a collection of text

Traditional approach: bag-of-words

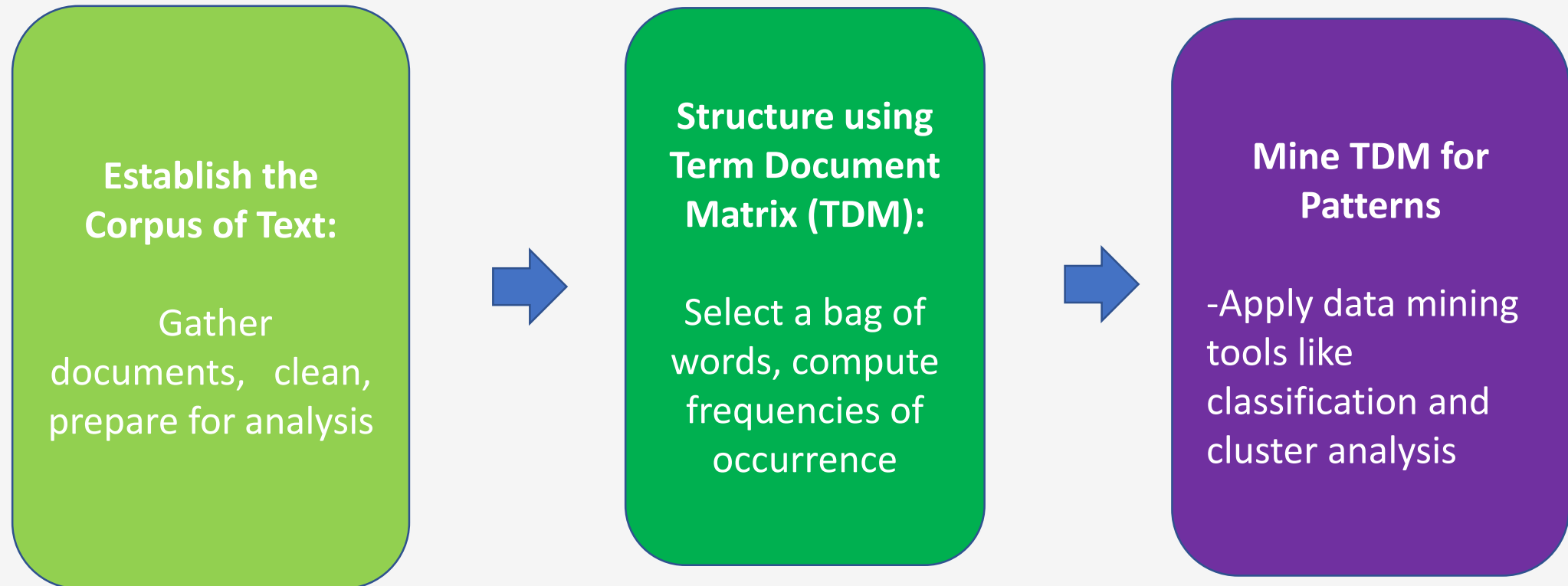
New approach: natural language processing for understanding nuances of spoken words

## Sentiment Analysis

A technique used to detect favorable and unfavorable opinions toward specific products and services

## Text Mining Process – three steps

---



# Text Mining Process

---

## Step 1: Establish the corpus

- Collect all relevant unstructured data  
e.g., textual documents, XML files, emails, Web pages, short notes, voice recordings...
- Digitize, standardize the collection  
e.g., all in ASCII text files
- Place the collection in a common place  
e.g., in a flat file, or in a directory as separate files

# Text Mining Process

---

## Step 2: Create the Term-by-Document Matrix

	Term Document Matrix				
Document / Terms	investment	Profit	happy	Success	...
Doc 1	10	4	3	4	
Doc 2	7	2	2		
Doc 3			2	6	
Doc 4	1	5	3		
Doc 5		6		2	
Doc 6	4		2		
...					

# Text Mining Process

---

## Step 2: Create the Term-by-Document Matrix (TDM), cont.

- Should all terms be included?
  - Stop words, include words
  - Synonyms, homonyms
  - Stemming
- What is the best representation of the indices (values in cells)?
  - Row counts; binary frequencies; log frequencies;
  - Inverse document frequency

### Step 2: Create the Term-by-Documents Matrix (TDM), cont.

- TDM is a sparse matrix. How can we reduce the dimensionality of the TDM?
  - Manual - a domain expert goes through it
  - Eliminate terms with very few occurrences in very few documents
  - Transform the matrix using singular value decomposition (SVD)
  - SVD is similar to principle component analysis
  - Phrase-Mining and Term-Mining



# Text Mining Process

---

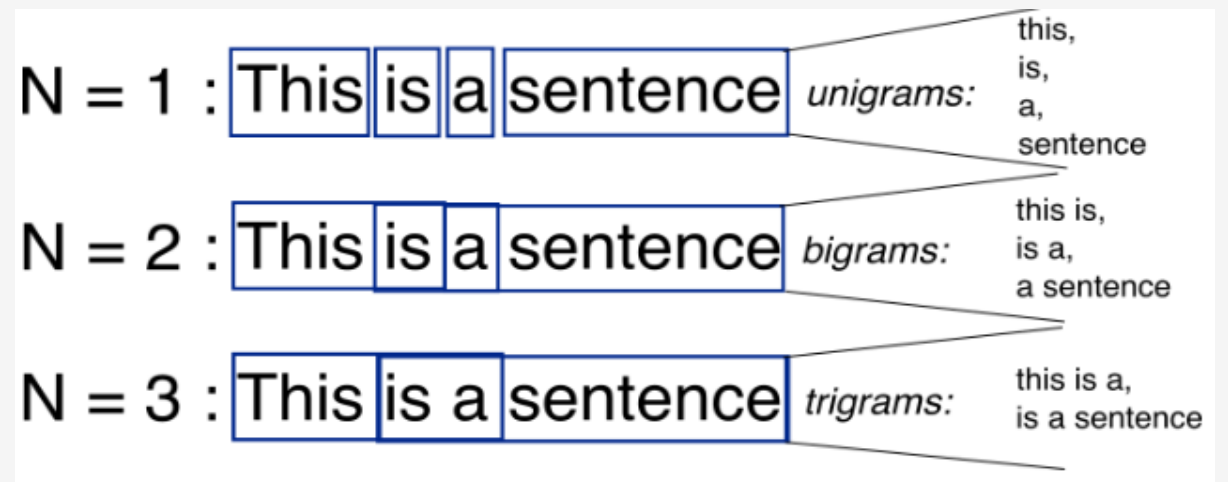
## Step 3: Extract patterns/knowledge

- Classification (text categorization)
- Clustering (natural groupings of text)
  - Improve search recall & precision
  - Scatter/gather
  - Query-specific clustering
- Association rules among the documents
- Trend Analysis

## Now, let's be more specific on the terminology

---

- Document: the unit that contains the text (Books, individual tweets, one emails)
- Corpus: a collection of documents ( books in a library, tweets feeds, emails received in a company)
- Stop words: relatively useless words in text mining ( a, an, the, she, he, why, .... )
- Tokenizer: function to split the text into individual words
- Stemming / Lemmatization: utility to group similar words (e.g wait, waiting, waited into wait)
- Bag of words: a bag of words
- N-grams: instead of consider a single word, it may be more meaningful to consider combination of words.



## Terminology cont'd (Term Frequency (TF), Inverse Document Frequency (IDF) )

Imagine you are a librarian to find the most relevant book from a search query “Book for Analytics newbie”

**Term Frequency** is just the frequency of a word in the document. The only thing is if a word X occurs in document A 1 time and in B 10 times, its generally not true that the word X is 10 times more relevant in B than in A. The difference is generally lesser as compared to the actual ratio. Hence we TF is defined as

$$\text{TF} = 1 + \log(\text{TF}) \text{ if } \text{TF} > 0$$
$$= 0 \text{ if } \text{TF} = 0$$

Book Number	Word Frequency								
	The	Big-Data	Analytics	Tree	newbie	book	for	Girl	honest
1	120	80	60	20	1	5	120	0	0
2	110	0	0	100	10	20	100	40	10
3	130	0	0	10	11	30	110	20	10
4	100	0	0	2	20	40	100	10	100
5	90	0	0	10	30	20	100	100	40

Book Number	TF								
	The	Big-Data	Analytics	Tree	newbie	book	for	Girl	honest
1	3.1	2.9	2.8	2.3	1.0	1.7	3.1	0.0	0.0
2	3.0	0.0	0.0	3.0	2.0	2.3	3.0	2.6	2.0
3	3.1	0.0	0.0	2.0	2.0	2.5	3.0	2.3	2.0
4	3.0	0.0	0.0	1.3	2.3	2.6	3.0	2.0	3.0
5	3.0	0.0	0.0	2.0	2.5	2.3	3.0	3.0	2.6

## TD-IDF Matrix

- **Inverse Document Frequency (IDF)** is based on the principle that less frequent words are generally more informative.

IDF =  $\log (N / DF)$  where

N = number of documents

DF = number of documents that has the word

IDF	The	Big-Data	Analytics	Tree	newbie	book	for	Girl	honest
N	5	5	5	5	5	5	5	5	5
DF	5	1	1	5	5	5	5	4	4
N/DF	1	5	5	1	1	1	1	1.25	1.25
Log(N/DF)	0.00	0.70	0.70	0.00	0.00	0.00	0.00	0.10	0.10

- **TF-IDF Matrix** is simply the multiplication of the TF and IDF
- Document 1 is the most relevant to a search query of “Book for Analytics newbie”

[illegible]

## Spam Mail Classification as NLP showcase

---

- **SMS Spam Collection Data Set from UCI**

<https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection>

- pip install nltk
- pip install wordcloud
- Pip install TextBlob

Learning by doing

## Some online tutorials

---

<https://www.analyticsvidhya.com/blog/2015/04/information-retrieval-system-explained/>

<https://www.analyticsvidhya.com/blog/2018/02/the-different-methods-deal-text-data-predictive-python/>

<https://medium.com/towards-artificial-intelligence/text-mining-in-python-steps-and-examples-78b3f8fd913b>

<https://towardsdatascience.com/spam-classifier-in-python-from-scratch-27a98ddd8e73>

<https://jakevdp.github.io/PythonDataScienceHandbook/05.05-naive-bayes.html>