Module 2: Foundations of Text Mining

What is Text Mining?

Text mining (also called **text data mining** or **text analytics**) is the process of extracting useful information, patterns, and knowledge from unstructured text data using techniques from natural language processing (NLP), machine learning (ML), statistics, and information retrieval.

- What is the importance of text mining, and why do we need it? Since most data in the real world (emails, articles, reports, social media posts, reviews, etc.) is in unstructured text form, text mining helps convert this raw text into structured and meaningful insights.

What are the Applications of Text Mining to Humanities, Social Science, Medicine, and Business?

- **Business Intelligence:** It is used to review customer reviews and feedback about products or services offered.
- **Healthcare:** It is used for mining medical records for diagnosis patterns.
- **Finance:** It is used for extracting insights from news articles for stock prediction.
- **Social Media:** It is used for sentiment analysis and trend tracking.
- **Legal:** It is used for analyzing case documents for precedents.
- **Academic Research:** It is used for summarizing and categorizing scientific articles.

Low-Tech Approaches to Text Mining

- These are simpler, rule-based, or statistical methods that do not require heavy computation or advanced ML. They often focus on word frequency, simple matching, or dictionaries.
- It is easy to implement.
- It is transparent and interpretable.
- It requires minimal computations.
- However, it can give shallow insights, it cannot handle synonyms, it does not understand context (for example, like sarcasm), nor can it understand semantics.
- The methods do not scale well to larger datasets.
- **Common Methods:** Word Count/Frequency Analysis, Concordance/Keyword in Context (KWIC), Lexicon-Based Sentiment Analysis, Collocation Detection, Regular Expression (RegEx) and Rules-Based Extraction.

High-Tech Approaches to Text Mining

- These use ML, Deep Learning (DL), and advanced NLP techniques to extract semantic, contextual, and predictive insights from text.
- They require more computational resources, large datasets, and specialized models.
- They provide rich and contextual insights.
- The methods scale to large datasets and even Big Data.
- The methods can handle ambiguity, synonyms, and context.
- **Common Methods:** Vector Space Models, Topic Modeling, ML Classification, DL for NLP, Named Entity Recognition (NER), Knowledge Graphs, etc.

Characters

- **Character:** The smallest unit of text, akin to how atoms are to a Chemist, an elementary particle is to a Physicist, or a prime number is to a Mathematician.
- Examples of characters: The Latin alphabet (a–z, A–Z), digits $(0-9)$, punctuation marks (.,?!;), white spaces (tabs, new lines), special symbols/characters (@, #, $, %), emojis, non-Latin alhphabets like Hindi, Chinese, Russian, Arabic, etc.

Encoding

- Computers do not "understand" characters directly – They only work with numbers (binary $= \{0, 1\}$).
- **Encoding** is the mapping between characters and their numeric representation.
- Common encoding standards:
  - **ASCII:** *American Standard Code for Information Interchange*.
  - **Extended ASCII/ISO-8859-1 (Latin-1)**
  - **Unicode**
  - **UTF-8 (Unicode Transformation Format)**

ASCII (American Standard Code for Information Interchange)

- Early encoding scheme composed of 7 bits that supports 128 characters.
- Supports only English letters, digits, and punctuation.
- **Example:** "A" $\rightarrow 65$, "a" $\rightarrow 97$.

Extended ASCII / ISO-8859-1 (Latin-1)

- Has 8 bits that represent 256 characters.
- The big innovation here is that it adds accented letters like ñ, á that are prevalent in many European languages.

Unicode

- A universal standard to represent text across all languages.
- Supports more than 150 scripts, including emojis.
- Within Unicode, Chinese has a unique code point.

UTF-8 (Unicode Transformation Format – 8-bit)

- It is the most ubiquitous encoding standard on the web.
- The encoding scheme has variable length. This means that:
  - 1 byte $=$ 8 bits is used for ASCII. For example, A $\rightarrow 01000001$.
  - Up to 4 bytes for other characters. For example, the smiling emoji $\rightarrow$ F0 9F 98 80.

The Importance of Encoding in NLP

- **Data Preprocessing:** Reading a file with the wrong encoding may cause errors.
- **Tokenization:** Proper encoding ensures punctuation, emojis, and non-Latin characters are handled correctly. We will speak more about *tokenization* later on!
- **Multilingual NLP:** Unicode allows mixing scripts (e.g., English and Hindi in the same dataset).
- **Model Training:** Neural networks rely on consistent numerical representations of characters/words.

Words, Subwords, Tokens, Sentences, and Embeddings

Depending on the granularity of analysis, text can be broken into:

- **Words:** The basic linguistic units of meaning in human language. Words are typically separated by spaces in languages like English. For example, "Natural Language Processing" has 3 words. However, in other languages like Chinese and Japanese, there are no explicit spaces.
- **Subwords:** These are smaller units obtained by splitting words into meaningful pieces. For example, the word "unhappiness" can be broken up into "un", "happi", and "ness". Famous models like `BERT` (Bidirectional Encoder Representations from Transformers) use algorithms like `WordPiece` and `SentencePiece` to handle subwords.

- **Tokenization:** It is the fundamental process of breaking down raw text into smaller, manageable units called **tokens**. These tokens are typically words, but can also be subwords or characters. The primary purpose of tokenization is to transform unstructured text into a standardized format that NLP models can understand and process, enabling them to analyze, understand, and derive meaning from human language for various tasks.
- **Tokens:** These are the units a model actually processes after tokenization. For example, consider "Hello World", the tokens can be:
  - **Characters:** [H, e, l, l, o, W, o, r, l, d]
  - **Words:** [Hello, World]
  - **Subwords:** [Hel, ##lo, World]

  A common misconception is that a token = word, but this is not necessarily the case!

- **Sentence:** A *sentence* is a specific combination of words and subwords that form a linguistic unit. In NLP, sentences are often detected by punctuation marks (., ,, ;, ?, !,...) For example, "I love NLP. It is fascinating!" has two sentences. However, to a computer, segmentation is not always as trivial as the previous example. As a demonstration, see this sentence "Dr. Patel went to the U.S. He works in A.I." Thus, we need careful handling of abbreviations.
- **Embeddings:** These are numerical vector representations of text units (words, tokens, sentences). With embeddings, the goal is to capture semantic meaning so that similar items are close in vector space.

Corpora

- The singular form is *corpus*, the plural is *corpora*.
- They are large collections of text documents used for training or evaluation.
- Examples of corpora in NLP are:
  - News articles.
  - Wikipedia
  - Research databases (PubMed, arXiv, etc.)
  - Dialogue datasets (Reddit, Quora, etc.)
- In NLP research, a corpus is the raw material from which tokens, embeddings, and models are built.

Terminology Summary

Term	**Definition**	**Example**
Word	Linguistic unit separated by spaces	"language", "processing"
Subword	Piece of a word (useful for rare words)	"un", "kind"
Token	Unit processed by the model (can be word, subword, or character)	"un", "equal"
Sentence	Sequence of words/tokens forming a unit of meaning	"I love chocolate. It tastes so good!"
Embedding	Vector representation of text in $\mathbb{R}^d$	$[0.21, -0.437, \ldots]$
Corpus	Large text collection for NLP tasks	Wikipedia articles, news datasets

Lab 2

Applications of frequency analysis, KWIC, lexicon-based sentiment analysis, collocation detection, RegEx, and the various encoding schemes (ASCII, Unicode, UTF-8) in the **business context**.