* How to increase efficiency of Word2Vec ?
  - ↳ Increasing the training dataset
  - ↳ Increase dimension of vectors
  - ↳ Increase window size.

LECTURE - 6 TEXT CLASSIFICATION

Text classification

classification → Supervised
- ↳ Tabular
- ↳ Image
- ↳ Text
  - ↳ email
  - ↳ sentiment
  - ↳ message → sales, support

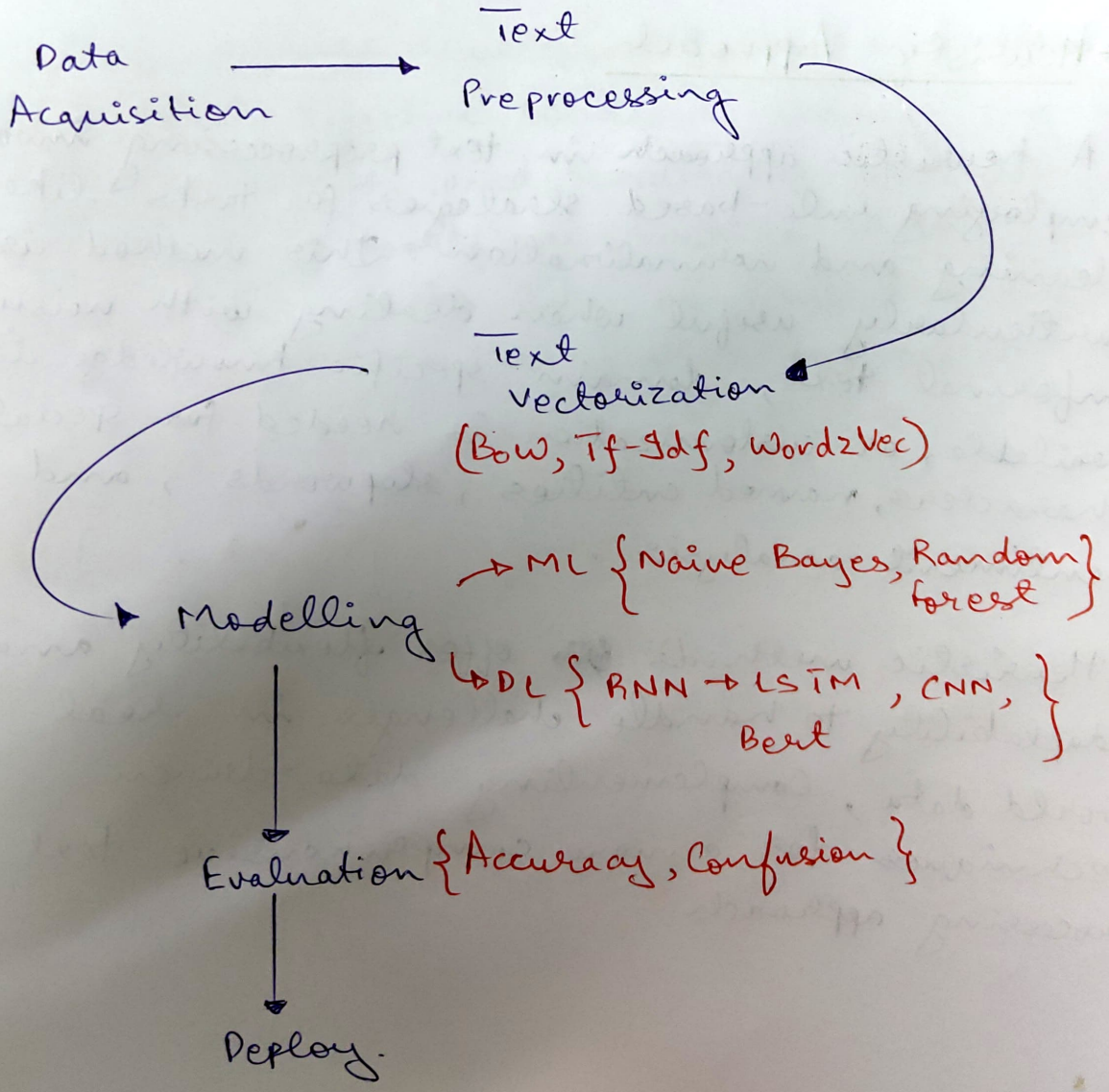
Text classification

- Binary {2 classes}
- Multi class {more than 2 classes}
- Multi Label 1 i/p multiple o/p.

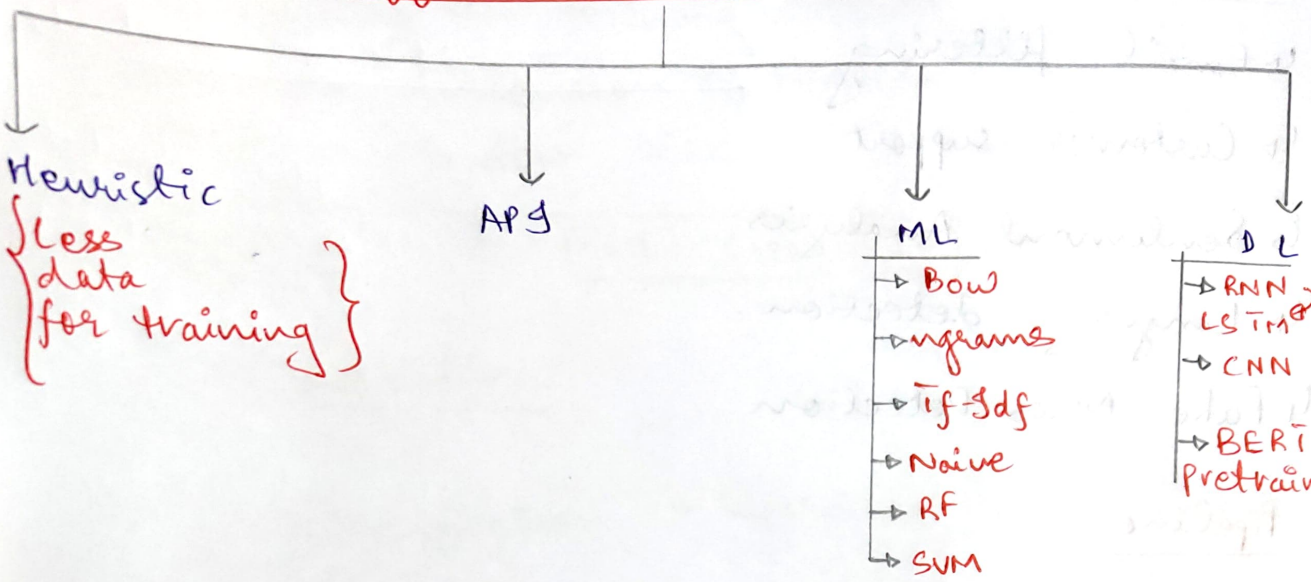
Applications

- Email filtering
- Customer support
- Sentiment Analysis
- Language detection.
- Fake News Detection

Pipeline



Different Approaches



* Heuristic Approach

↳ A heuristic approach in text preprocessing involves employing rule-based strategies for tasks like cleaning and normalization. This method is particularly useful when dealing with noisy or informal text, domain-specific knowledge is available, or customization is needed for special characters, named entities, stopwords, and sentiment analysis.

↳ Heuristic methods offer flexibility and adaptability to handle challenges in real world data, complementing data-driven techniques for a more comprehensive text processing approach.

* APIs

↳ An API approach in text preprocessing is employed for efficient, scalable, and advanced language processing. This method is advantageous when dealing with large scale data, accessing pre-trained models, supporting multiple languages, and utilizing constantly updated linguistic resources.

↳ APIs reduce development time and effort, offer cost-effective solutions through pay-as-you-go models, and seamlessly integrate with existing workflows, making them valuable for tasks like sentiment analysis, entity recognition, and other NLP applications.

* Machine learning Approach

① Bag of words (BoW)

→ BoW approach is suitable for text preprocessing when semantic relationships and word order are less critical. It's effective in scenarios like document classification, sentiment analysis, and information retrieval, where word frequency matters more than context.

→ BoW simplifies complex texts into word frequency vectors, allowing for straightforward numerical representation. Despite its lack of context, BoW is computationally efficient, making it practical for large-scale text processing tasks with a focus on overall word occurence patterns.

② Tf-Idf

↳ A Tf-Idf approach is employed in text preprocessing when aiming to highlight the importance of words in a document within a larger corpus. Useful for tasks like information retrieval and document clustering, Tf-Idf accounts for both word frequency and rarity across the dataset.

↳ By assigning weights that reflect a term's significance, tf-idf captures the distinctive features of documents, making it valuable for applications where unique keyword relevance is crucial.

③ Using Word2Vec

↳ A word2vec approach is employed in text preprocessing when aiming to capture semantic relationships b/w words. Suitable for tasks like sentiment analysis, machine translation, and document similarity, word2Vec generates dense vector representation for words, preserving contextual nuances. By placing similar words closer in the vector space, word2Vec excels at capturing word semantics.

↳ This approach is beneficial when contextual understanding and semantic relationships are essential, enhancing the performance of various NLP applications.