

which model would be Apply It depends on the

⇒ Amount of Data

⇒ Nature of problem

If we have less data then we should apply Heuristic. If data little bit complex the ML or more complex then use DL.

If we have existing solution in cloud then we don't use above those technique.

## NLP

What is Natural Language Processing (NLP) pipeline?

⇒ NLP is a set of steps followed to build an end to end NLP software.

NLP software consists of following steps:

⇒ **Data Acquisition**

⇒ **Text Preparation**

- **Text Clean up** : his involves removing any unwanted characters or symbols from the text, such as punctuation, HTML tags, and emojis, spelling check[asp means as soon as possible].

```
# Original text:
```

```
This is a tweet with punctuation and emojis: .
```

```
# Cleaned text:
```

```
This is a tweet with punctuation and emojis.
```

- **Basic Pre-processing (ex: tokenization)**: This includes tasks such as **tokenization(sentence and word), stemming(dance , dances and dancing), and lemmatization**. Tokenization involves splitting the text into individual tokens, such as words or phrases. Stemming and lemmatization are both techniques for reducing word forms to their base or root form.

```
# Original text:
```

```
I love to eat pizza.
```

```
# Tokenized text:
```

```
["I", "love", "to", "eat", "pizza"]
```

```
# Stemmed text:
```

```
["lov", "eat", "pizz"]
```

```
# Lemmatized text:
```

```
["love", "eat", "pizza"]
```

- **Advance Pre-processing**: This includes more sophisticated tasks such as stop-word removal, part-of-speech tagging, and named entity recognition. Stop-word removal involves removing common words from the text that do not add much meaning, such as "the," "is," and "of." Part-of-speech tagging involves assigning a part-of-speech (e.g., noun, verb, adjective) to each word in the text. Named entity recognition involves identifying named entities in the text, such as people, places, and organizations.

**Original text:**

I love to eat pizza with my friends.

**Stop words removed text:**

love eat pizza friends

**Part-of-speech tagged text:**

[('I', 'PRONOUN'), ('love', 'VERB'), ('to', 'TO'), ('eat', 'VERB'), ('pizza', 'NOUN'), ('with', 'WITH'), ('my', 'DETERMINER'), ('friends', 'NOUN')]

**Named entities recognized text:**

[('I', 'PERSON'), ('friends', 'PERSON')]

Text preparation can be a complex and time-consuming process, but it is an essential step in building and deploying high-performing NLP models.



**Feature Engineering**



**Modelling (Apply Algorithm)**

- Model Building
- Evaluation



**Deployment**

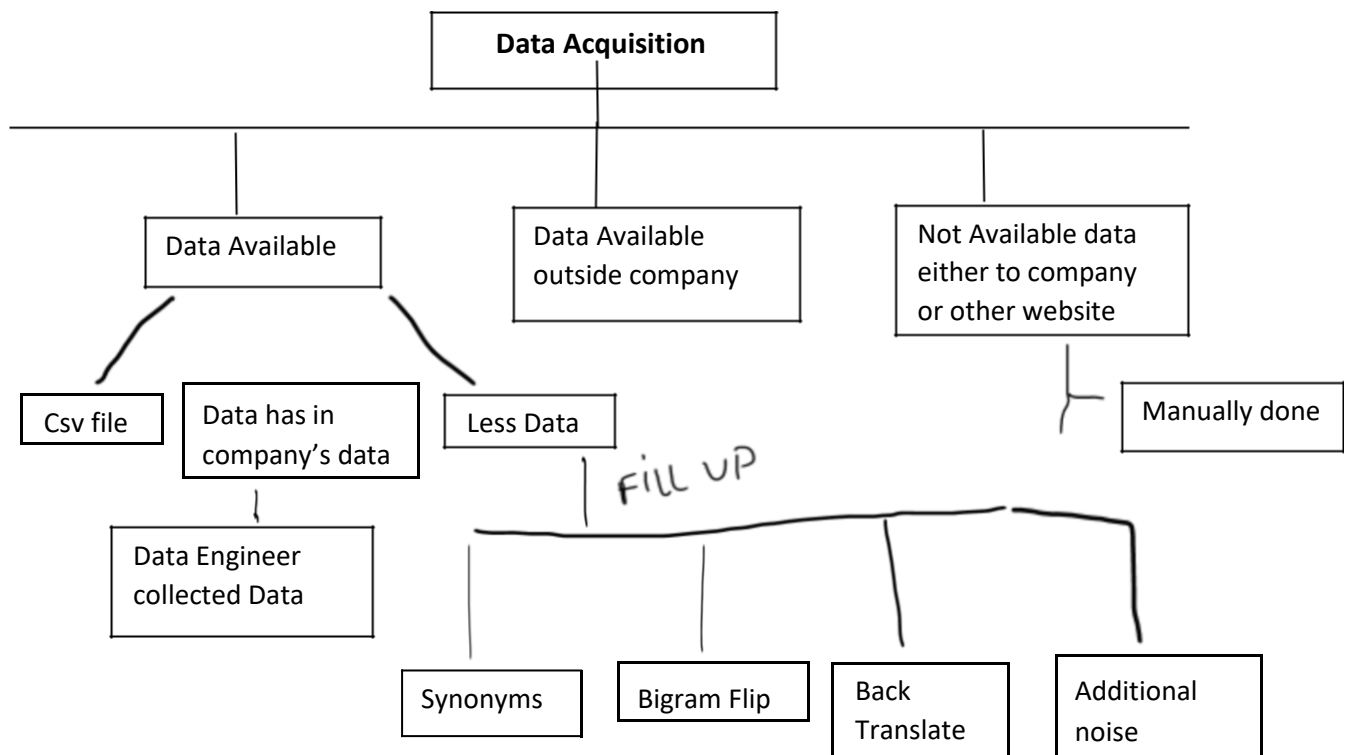
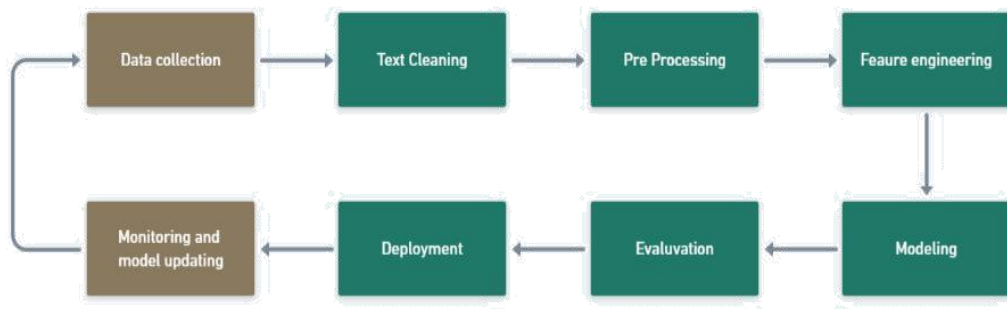
- Deployment
- Monitoring
- Model Updatex

**Points to remember:**

For NLP machine learning and Deep Learning Pipeline is not same.

For ML we have to create feature manually[**need domain knowledge, all features are not good**], on the other hand , In DPL it's create automatically[ **no need domain knowledge and interpret sometimes could lose and it's tough to understand**]

- It's not an universal pipeline (universal means apply everywhere)
- Deep Learning Pipelines are slightly different
- Pipeline is non-linear

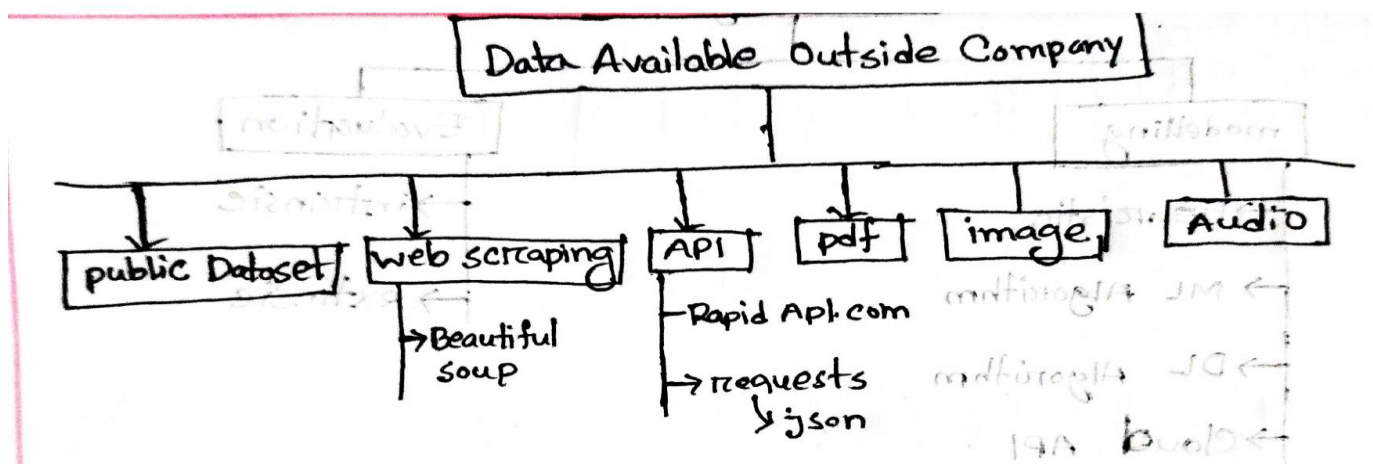
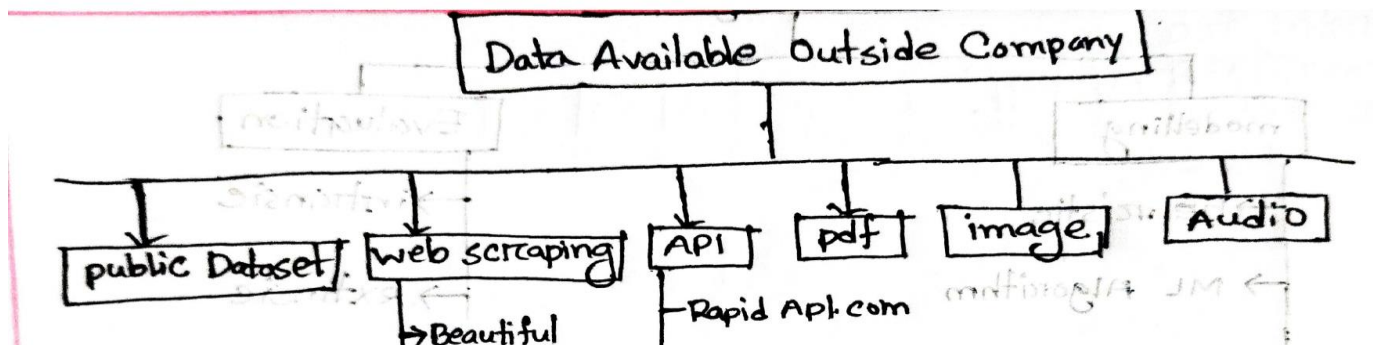


**Bigram:** Combination of two words.


Like : I'm walking on the read, here **on the = the on**

**Back Translate:** First translate sentences then It again translate into original sentences.

**Additional Noise:** add noise into the sentences but it remains same meaningful sentences.



In NLP, data acquisition is the process of collecting or creating the data necessary for an NLP task. This data can be in the form of text, speech, or images.

There are a number of different ways to  acquire data for NLP. One common approach is to use publicly available datasets. There are many websites and repositories that offer datasets for a variety of NLP tasks, such as sentiment analysis, machine translation, and question answering.

Another approach is to collect your own data. This can be done by creating surveys, scraping the web, or transcribing audio or video recordings. When collecting your own data, it is important to make sure that the data is representative of the task that you are trying to solve.

Once you have collected your data, you will need to preprocess it before you can use it to train your NLP model. This may involve cleaning the data, removing errors, and converting it to a format that your model can understand.

Here are some examples of data acquisition for specific NLP tasks:

- **Sentiment analysis:** You could collect a dataset of product reviews and their associated ratings. This dataset could be used to train a sentiment analysis model that can predict the sentiment of a piece of text.
- **Machine translation:** You could collect a dataset of text pairs in two different languages. This dataset could be used to train a machine translation model that can translate text from one language to the other.
- **Question answering:** You could collect a dataset of question-answer pairs. This dataset could be used to train a question answering model that can answer questions about a given topic.

The quality and quantity of the data that you use for NLP can have a significant impact on the performance of your model. Therefore, it is important to carefully consider your data acquisition strategy.

## **Data Acquisition**

The data for NLP tasks can be acquired from a variety of sources, such as:

- The web (e.g., social media, news articles, product reviews)
- Books and journals
- Public datasets (e.g., Wikipedia, IMDb)
- Private datasets (e.g., customer support tickets, medical records)

## **Text Preparation**

The text preparation step involves cleaning and transforming the raw text data into a format that is suitable for analysis. This may include:

- Removing noise and inconsistencies (e.g., punctuation, HTML tags, emojis)
- Converting the text to a consistent format (e.g., lowercase, ASCII)
- Tokenizing the text (i.e., splitting it into individual words or phrases)

## **Advanced Text Preprocessing**

In some cases, advanced text preprocessing steps may be required, such as:

- Stemming and lemmatization (i.e., reducing words to their base form)
- Stop-word removal (i.e., removing common words that do not add much meaning to the text)
- Part-of-speech tagging (i.e., assigning a part-of-speech to each word in the text)
- Named entity recognition (i.e., identifying named entities in the text, such as people, places, and organizations)

## **Feature Engineering**

Feature engineering is the process of creating new features from the existing data. This can be done to improve the performance of the machine learning model.

For example, you could create new features that represent the length of a sentence, the number of punctuation marks in a sentence, or the presence of certain keywords in a sentence.

## Modeling

The modeling step involves choosing a machine learning algorithm and training it on the prepared data.

The choice of algorithm will depend on the specific NLP task that you are trying to solve. For example, you might use a Naive Bayes classifier for text classification, a support vector machine (SVM) for sentiment analysis, or a recurrent neural network (RNN) for question answering.

## Evaluation

Once the model is trained, you need to evaluate its performance on a held-out test set. This will help you to identify any areas where the model needs improvement.

There are two main types of evaluation metrics: intrinsic and extrinsic.

**Intrinsic evaluation metrics**(TECHNICAL LEVEL) measure the performance of the model on a specific task, such as **Precision, recall, F1 score, accuracy, etc.** **Extrinsic evaluation metrics**(BUSSINESS LEVEL)measure the impact of the model on a real-world application(**Human evaluation, task-specific metrics (e.g., BLEU score for machine translation)**), etc..

## Deployment

Once you are satisfied with the performance of the model, you can deploy it to production. This may involve integrating the model into an existing application or developing a new application to use the model.

## Monitoring

Once the model is deployed, it is important to monitor its performance over time. This will help you to identify any problems with the model and make necessary adjustments.



## **Model Update Strategy**

As the data changes over time, you may need to update your model to maintain its performance. There are a few different ways to do this:

- Retrain the model on the new data.
- Fine-tune the model on the new data.
- Use a technique called transfer learning to transfer knowledge from a pre-trained model to your model.

Which approach you choose will depend on the specific NLP task that you are trying to solve and the resources that you have available.