**Applications of NLP**

So far we have seen basic concepts in NLP such as elements of text, annotations, tagging, handling a corpus of text, and lexical resources.

In this module, we will be using the concepts learnt so far to build the following sample applications:

1. SMS spam classification - Classify if an SMS is spam or ham based on the content in it
2. Sentiment Analysis of product reviews - Classify if a product review has a positive sentiment or a negative sentiment
3. Indian Railways FAQ chatbot - Automate the response to frequently asked questions on Indian Railways ticket booking
4. Topic modelling - Identify key topics in a document and group document that belong to similar topics together

Before we start developing the applications let us review the sequence of steps that we typically go through while developing NLP applications

The typical NLP pipeline involves a variety of steps as listed below:

# Collecting textual data

* This step involves bringing in relevant data from different sources. Sources can include both external and internal systems and depends on the domain of the problem.

# Cleaning the input text data

* We use various techniques to clean the data, depending on the domain of the problem. Regular expressions are extensively used at this stage in identifying anomalies in the data and transforming them as required

# Normalizing the data

* Normalization of text data includes techniques such as "removal of stop words", "converting all text to a standard case", "stemming/lemmatization", etc.

# Feature engineering

* Feature engineering textual data typically involves converting the cleaned and normalized data into vector representations. These vectors can then be used by machine learning algorithms to build models

# Apply machine learning techniques to build models

* Depending on the problem at hand, we can apply different machine learning algorithms. For example, when we are build a spam classifier/Sentiment Analyzer, we would use popular classification algorithms such as Naive Bayes, Decision Trees, etc. If we are building a system that auto-completes user input, then we may use hidden markov models.

# Validating the model built

* After building the model, we check it against test data to analyse how the model performs on previously unseen data. If the model performance is satisfactory, we can be more confident in deploying it

# Deploying the model and making prediction on new data.

* The model is now used on real-time data to make predictions/provide responses. For example, when a spam classifier is deployed, it reads messages in real-time and classifies which of those are spam and which are not, and take neccessary action.