Natural Language Processing, or NLP a field that is focused on enabling computers to understand and process human languages. At present day, with the growth of social media, most of the data available is textual in nature.

We find its application everywhere- from ‘hey SIRI’, ‘ok google’ or Alexa. When you're typing on a phone, like many of us do every day, you'll see word suggestions based on what you type and what you're currently typing. That's natural language processing in action.

Text is the most unstructured form of all the available data, various types of noise are present in it and the data is not readily analyzable without any pre-processing. This is the most important phase in NLP.

For this experiment we used the sentiment labelled data set from the UCI repository and the NLTK python library.

We tokenized each comment and then removed the punctuation and stop words using the English corpus from the NLTK library. This is an essential noise removal process from the dataset. It is also essential to transform the all the tokens into lower case before any comparisons are made. The frequency distribution helps us to understand the most common occurring words in the dataset. This changes significantly after the removal of the stop words.

Every word in a sentence is also associated with a part of speech (pos) tag (nouns, verbs, adjectives, adverbs etc). The pos tags defines the usage and function of a word in the sentence. POS tags are the basis of lemmatization process for converting a word to its base form (lemma).

Stemming is a rudimentary rule-based process of stripping the suffixes (“ing”, “ly”, “es”, “s” etc) from a word.

Lemmatization, on the other hand, is an organized & step by step procedure of obtaining the root form of the word

A combination of N words together are called N-Grams. N grams (N > 1) are generally more informative as compared to words (Unigrams) as features. Also, bigrams (N = 2) are considered as the most important features of all the others.

TF-IDF is a weighted model commonly used for information retrieval problems. It aims to convert the text documents into vector models on the basis of occurrence of words in the documents without taking considering the exact ordering.

Term Frequency (TF) – TF for a term “t” is defined as the count of a term “t” in a document “D”

Inverse Document Frequency (IDF) – IDF for a term is defined as logarithm of ratio of total documents available in the corpus and number of documents containing the term T.

The TF gives importance to frequently occurring words in a document, however the IDF part weighs/dampens the effect if a particular word is occurring over the entire set of documents.

The parameters for the Tfidfvectorizer are explained in detail here (<http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html>)

The model in this case is trained using a Naïve Bayes model here and predict by analysing the comments whether they were positive or negative.

We also performed an experiment using the youtube comments dataset, to figure out if the comments were spam/ham. We performed the data pre-processing and cleaning steps as before and applied the RandomForestClassifer. We performed a GridSearch to figure out the optimum set of the hyper-parameters.

To Do: Apply other algorithms on the dataset and figure out if we can better the results.