**What is Natural Language Processing?**

Natural language processing is the application of computational linguistics to build real-world applications which work with languages comprising of varying structures. We are trying to teach the computer to learn languages, and then also expect it to understand it, with suitable efficient algorithms.

All of us have come across Google’s keyboard which suggests auto-corrects, word predicts (words that would be used) and more. Grammarly is a great tool for content writers and professionals to make sure their articles look professional. It uses ML algorithms to suggest the right amounts of gigantic vocabulary, tonality, and much more, to make sure that the content written is professionally apt, and captures the total attention of the reader. Translation systems use language modelling to work efficiently with multiple languages.

**How did Natural Language Processing come to exist?**

People involved with language characterization and understanding of patterns in languages are called linguists. Computational linguistics kicked off as the amount of textual data started to explode tremendously. Wikipedia is the greatest textual source there is. The field of computational linguistics began with an early interest in understanding the patterns in data, Parts-of Speech(POS) tagging, easier processing of data for various applications in the banking and finance industries, educational institutions, etc.

**How does Natural Language Processing work?**

NLP aims at converting unstructured data into computer-readable language by following attributes of natural language. Machines employ complex algorithms to break down any text content to extract meaningful information from it. The collected data is then used to further teach machines the logics of natural language.

Natural language processing uses syntactic and semantic analysis to guide machines by identifying and recognising data patterns. It involves the following steps:

**Syntax:** Natural language processing uses various algorithms to follow grammatical rules which are then used to derive meaning out of any kind of text content. Commonly used syntax techniques are lemmatization, morphological segmentation, word segmentation, part-of-speech tagging, parsing, sentence breaking, and stemming.

**Semantics:** This is a comparatively difficult process where machines try to understand the meaning of each section of any content, both separately and in context. Even though semantical analysis has come a long way from its initial binary disposition, there’s still a lot of room for improvement. NER or Named Entity Recognition is one of the primary steps involved in the process which segregates text content into predefined groups. Word sense disambiguation is the next step in the process, and takes care of contextual meaning. Last in the process is Natural language generation which involves using historical databases to derive meaning and convert them into human languages.

**Why natural language processing is important?**

The amount of data generated by us keep increasing by the day, raising the need for analysing and documenting this data. NLP enables computers to read this data and convey the same in languages humans understand.

From medical records to recurrent government data, a lot of these data is unstructured. NLP helps computers to put them in proper formats. Once that is done, computers analyse texts and speech to extract meaning. Not only is the process automated, but also near-accurate all the time.

**Why is Advancement in the Field of NLP Necessary?**

NLP is the process of enhancing the capabilities of computers to understand human language. Databases are highly structured forms of data. Internet, on the other hand, is completely unstructured with minimal components of structure in it. In such a case, understanding human language and modelling it is the ultimate goal under NLP. For example, Google Duplex and Alibaba’s voice assistant are on the journey to mastering non-linear conversations. Non-linear conversations are somewhat close to the human’s manner of communication. We talk about cats in the first sentence, suddenly jump to talking tom, and then refer back to the initial topic. The person listening to this understands the jump that takes place. Computers currently lack this capability.

**What can natural language processing do?**

Currently, NLP professionals are in a lot of demand, for the amount of unstructured data available is increasing at a very rapid pace. Underneath this unstructured data lies tons of information that can help companies grow and succeed. For example, monitoring tweet patterns can be used to understand the problems existing in the societies, and it can also be used in times of crisis. Thus, understanding and practicing NLP is surely a guaranteed path to get into the field of machine learning. For beginners, creating a NLP portfolio would highly increase the chances of getting into the field of NLP.

**What are some of the applications of NLP?**

Grammarly, Microsoft Word, Google Docs

Search engines like DuckDuckGo, Google

Voice assistants – Alexa, Siri

News feeds- Facebook,Google News

Translation systems – Google translate

**How to learn Natural Language Processing (NLP)?**

To start with, you must have a sound knowledge of programming languages like Python, Keras, NumPy, and more. You should also learn the basics of cleaning text data, manual tokenization, and NLTK tokenization. The next step in the process is picking up the bag-of-words model (with Scikit learn, keras) and more. Understand how the word embedding distribution works and learn how to develop it from scratch using Python. Embedding is an important part of NLP, and embedding layers helps you encode your text properly. After you have picked up embedding, it’s time to lean text classification, followed by dataset review.

**Available Open-Source softwares in NLP Domain**

* NLTK
* Stanford toolkit
* Gensim
* Open NLP

We will understand traditional NLP, a field which was run by the intelligent algorithms that were created to solve various problems. With the advance of deep neural networks, NLP has also taken the same approach to tackle most of the problems today. In this article we will cover traditional algorithms to ensure the fundamentals are understood.

We look at the basic concepts such as regular expressions, text-preprocessing, POS-tagging and parsing.

**What are Regular Expressions?**

Regular expressions are effective matching of patterns in strings. Patterns are used extensively to get meaningful information from large amounts of unstructured data. There are various regular expressions involved. Let us consider them one by one:

(a period): All characters except for n are matched

w: All [a-z A-Z 0-9] characters are matched with this expression

$: Every expression ends with $

: used to nullify the speciality of the special character.

W (upper case W) matches any non-word character.

s: This expression (lowercase s) matches a single white space character – space, newline,

return, tab, form [nrtf].

r: This expression is used for a return character.

S: This expression matches any non-white space character.

t: This expression performs a tab operation.

n: Used to express a newline character.

d: Decimal digit [0-9].

^: Used at the start of the string.

**What is Sentiment Analysis?**

Sentiment analysis is the process of analyzing digital text to determine if the emotional tone of the message is positive, negative, or neutral. Today, companies have large volumes of text data like emails, customer support chat transcripts, social media comments, and reviews. Sentiment analysis tools can scan this text to automatically determine the author’s attitude towards a topic. Companies use the insights from sentiment analysis to improve customer service and increase brand reputation.

**Why is sentiment analysis important?**

Sentiment analysis, also known as opinion mining, is an important business intelligence tool that helps companies improve their products and services. We give some benefits of sentiment analysis below.

**Provide objective insights**

Businesses can avoid personal bias associated with human reviewers by using artificial intelligence (AI)–based sentiment analysis tools. As a result, companies get consistent and objective results when analyzing customers’ opinions.

For example, consider the following sentence:

I'm amazed by the speed of the processor but disappointed that it heats up quickly.

Marketers might dismiss the discouraging part of the review and be positively biased towards the processor's performance. However, accurate sentiment analysis tools sort and classify text to pick up emotions objectively.

Build better products and services

A sentiment analysis system helps companies improve their products and services based on genuine and specific customer feedback. AI technologies identify real-world objects or situations (called entities) that customers associate with negative sentiment. From the above example, product engineers focus on improving the processor's heat management capability because the text analysis software associated disappointed (negative) with processor (entity) and heats up (entity).

Analyze at scale

Businesses constantly mine information from a vast amount of unstructured data, such as emails, chatbot transcripts, surveys, customer relationship management records, and product feedback. Cloud-based sentiment analysis tools allow businesses to scale the process of uncovering customer emotions in textual data at an affordable cost.

Real-time results

Businesses must be quick to respond to potential crises or market trends in today's fast-changing landscape. Marketers rely on sentiment analysis software to learn what customers feel about the company's brand, products, and services in real time and take immediate actions based on their findings. They can configure the software to send alerts when negative sentiments are detected for specific keywords.

**What are sentiment analysis use cases?**

Businesses use sentiment analysis to derive intelligence and form actionable plans in different areas.

Improve customer service

Customer support teams use sentiment analysis tools to personalize responses based on the mood of the conversation. Matters with urgency are spotted by artificial intelligence (AI)–based chatbots with sentiment analysis capability and escalated to the support personnel.

Brand monitoring

Organizations constantly monitor mentions and chatter around their brands on social media, forums, blogs, news articles, and in other digital spaces. Sentiment analysis technologies allow the public relations team to be aware of related ongoing stories. The team can evaluate the underlying mood to address complaints or capitalize on positive trends.

Market research

A sentiment analysis system helps businesses improve their product offerings by learning what works and what doesn't. Marketers can analyze comments on online review sites, survey responses, and social media posts to gain deeper insights into specific product features. They convey the findings to the product engineers who innovate accordingly.

Track campaign performance

Marketers use sentiment analysis tools to ensure that their advertising campaign generates the expected response. They track conversations on social media platforms and ensure that the overall sentiment is encouraging. If the net sentiment falls short of expectation, marketers tweak the campaign based on real-time data analytics.

**How does sentiment analysis work?**

Sentiment analysis is an application of natural language processing (NLP) technologies that train computer software to understand text in ways similar to humans. The analysis typically goes through several stages before providing the final result.

Preprocessing

During the preprocessing stage, sentiment analysis identifies key words to highlight the core message of the text.

Tokenization breaks a sentence into several elements or tokens.

Lemmatization converts words into their root form. For example, the root form of am is be.

Stop-word removal filters out words that don't add meaningful value to the sentence. For example, with, for, at, and of are stop words.

Keyword analysis

NLP technologies further analyze the extracted keywords and give them a sentiment score. A sentiment score is a measurement scale that indicates the emotional element in the sentiment analysis system. It provides a relative perception of the emotion expressed in text for analytical purposes. For example, researchers use 10 to represent satisfaction and 0 for disappointment when analyzing customer reviews.

**What are the approaches to sentiment analysis?**

There are three main approaches used by sentiment analysis software.

Rule-based

The rule-based approach identifies, classifies, and scores specific keywords based on predetermined lexicons. Lexicons are compilations of words representing the writer's intent, emotion, and mood. Marketers assign sentiment scores to positive and negative lexicons to reflect the emotional weight of different expressions. To determine if a sentence is positive, negative, or neutral, the software scans for words listed in the lexicon and sums up the sentiment score. The final score is compared against the sentiment boundaries to determine the overall emotional bearing.

Rule-based analysis example

Consider a system with words like happy, affordable, and fast in the positive lexicon and words like poor, expensive, and difficult in a negative lexicon. Marketers determine positive word scores from 5 to 10 and negative word scores from -1 to -10. Special rules are set to identify double negatives, such as not bad, as a positive sentiment. Marketers decide that an overall sentiment score that falls above 3 is positive, while - 3 to 3 is labeled as mixed sentiment.

Pros and cons

A rule-based sentiment analysis system is straightforward to set up, but it's hard to scale. For example, you'll need to keep expanding the lexicons when you discover new keywords for conveying intent in the text input. Also, this approach may not be accurate when processing sentences influenced by different cultures.

ML

This approach uses machine learning (ML) techniques and sentiment classification algorithms, such as neural networks and deep learning, to teach computer software to identify emotional sentiment from text. This process involves creating a sentiment analysis model and training it repeatedly on known data so that it can guess the sentiment in unknown data with high accuracy.

Training

During the training, data scientists use sentiment analysis datasets that contain large numbers of examples. The ML software uses the datasets as input and trains itself to reach the predetermined conclusion. By training with a large number of diverse examples, the software differentiates and determines how different word arrangements affect the final sentiment score.

Pros and cons

ML sentiment analysis is advantageous because it processes a wide range of text information accurately. As long as the software undergoes training with sufficient examples, ML sentiment analysis can accurately predict the emotional tone of the messages. However, a trained ML model is specific to one business area. This means sentiment analysis software trained with marketing data cannot be used for social media monitoring without retraining.

Hybrid

Hybrid sentiment analysis works by combining both ML and rule-based systems. It uses features from both methods to optimize speed and accuracy when deriving contextual intent in text. However, it takes time and technical efforts to bring the two different systems together.

What are the different types of sentiment analysis?

Businesses use different types of sentiment analysis to understand how their customers feel when interacting with products or services.

Fine-grained scoring

Fine-grained sentiment analysis refers to categorizing the text intent into multiple levels of emotion. Typically, the method involves rating user sentiment on a scale of 0 to 100, with each equal segment representing very positive, positive, neutral, negative, and very negative. Ecommerce stores use a 5-star rating system as a fine-grained scoring method to gauge purchase experience.

Aspect-based

Aspect-based analysis focuses on particular aspects of a product or service. For example, laptop manufacturers survey customers on their experience with sound, graphics, keyboard, and touchpad. They use sentiment analysis tools to connect customer intent with hardware-related keywords.

Intent-based

Intent-based analysis helps understand customer sentiment when conducting market research. Marketers use opinion mining to understand the position of a specific group of customers in the purchase cycle. They run targeted campaigns on customers interested in buying after picking up words like discounts, deals, and reviews in monitored conversations.

Emotional detection

Emotional detection involves analyzing the psychological state of a person when they are writing the text. Emotional detection is a more complex discipline of sentiment analysis, as it goes deeper than merely sorting into categories. In this approach, sentiment analysis models attempt to interpret various emotions, such as joy, anger, sadness, and regret, through the person's choice of words.

What are the challenges in sentiment analysis?

Despite advancements in natural language processing (NLP) technologies, understanding human language is challenging for machines. They may misinterpret finer nuances of human communication such as those given below.

Sarcasm

It is extremely difficult for a computer to analyze sentiment in sentences that comprise sarcasm. Consider the following sentence, Yeah, great. It took three weeks for my order to arrive. Unless the computer analyzes the sentence with a complete understanding of the scenario, it will label the experience as positive based on the word great.

Negation

Negation is the use of negative words to convey a reversal of meaning in the sentence. For example, I wouldn't say the subscription was expensive. Sentiment analysis algorithms might have difficulty interpreting such sentences correctly, particularly if the negation happens across two sentences, such as, I thought the subscription was cheap. It wasn't.

Multipolarity

Multipolarity occurs when a sentence contains more than one sentiment. For example, a product review reads, I'm happy with the sturdy build but not impressed with the color. It becomes difficult for the software to interpret the underlying sentiment. You'll need to use aspect-based sentiment analysis to extract each entity and its corresponding emotion.

**Build Model for sentiment classification**

We will build different models to classify sentiments.

1. Naive-Bayes classifier

2. TF-IDF VECTORIZER

Now we will discuss one by one and also will see the comparison. Let’s start

**First will discuss the Naive Bayes classifier**

**Naive-Bayes Model For Sentiment Classification**

Naive-Bayes classifier is widely used in Natural language processing and proved to give better results. It works on the concept of Baye’s theorem.

Assume that we would like to predict whether the probability of a document is positive given that the document contains a word awesome. This can be computed if the probability of the word awesome appearing in a document given that it is positive sentiment is multiplied by the probability of the document being positive.

P(doc = +ve | word = awesome) = P(word = awesome | doc = +ve) \* P(doc = +ve)

The posterior probability of the sentiment is computed from the prior probabilities of all the words it contains. The assumption is that the occurrences of the words in a document are considered independent and they do not influence each other. So, if the document contains N-words and words represented as w1, w2, w3………wn then

P(doc = +ve | word = w1, w2, w3………wn) =

sklearn.naive\_bayes provides a class BernoulliNB which is a Naive -Bayes classifier for multivariate BernoulliNB models. BernoulliNB is designed for binary features, which is the case here.

The steps involved in using the Naive-Bayes model for sentiment classification are as follows:

Split the dataset into train and validation sets,

Build the Naive-Bayes model,

Find model accuracy.

we will discuss these in the following subsections.

Split the dataset into train and validation sets

Split the dataset into a 70:30 ratio for creating training and test datasets using the following code.

from sklearn.model\_selection import train\_test\_split

train\_x, test\_x, train\_y, test\_y = train\_test\_split(train\_ds\_features, train\_data.Sentiment,

test\_size = 0.3, random\_state = 42)

Build Naive-Bayes Model

Build Naive-Bayes model using the training set.

from sklearn.naive\_bayes import BernoulliNB

nb\_clf = BernoulliNB()

nb\_clf.fit(train\_x.toarray(), train\_y)

Make a prediction on Test case

The predicted class will be the one that has the higher probability based on Naive-Baye’s Probability calculation. Predict the sentiments of the test dataset using predict() method.

test\_ds\_predicted = nb\_clf.predict(test\_x.toarray())

Finding Model Accuracy

Let us print the classification report.

from sklearn import metrics

print(metrics.classification\_report(test\_y,test\_ds\_predicted))

finding model | sentiment classification

The Model is classifying with very high accuracy. Both average precision and recall is about 98% for identifying positive and negative sentiment documents. Let us draw the confusion matrix.

cm = metrics.confusion\_matrix(test\_y, test\_ds\_predicted)

sn.heatmap(cm, annot=True, fmt = '.2f')

In the confusion matrix, the rows represent the actual number of positive and negative documents in the test set, whereas the columns represent what the model has predicted. Label 1 means positive sentiment and label 0 means negative sentiment.

confusion matrix | sentiment classification

As per the model prediction, that there are only 13 instances that are wrongly classified as a negative sentiment document and there are only 26 negative sentiment documents classified wrongly as positive sentiment documents. Rest all have been classified correctly.

The next section will discuss TD-IFD Vectorizer Model.

**TF-IDF VECTORIZER**

TfidfVectorizer is used to create both TF Vectorizer and TF-IDF Vectorizer. It Takes a parameter to use \_idf to create TF-IDF vectors. If use \_idf set to false, it will create only TF vectors and if it is set to True, it will create TF-IDF vectors.

from sklearn.feature\_extraction.text import TfidfVectorizer

tfidf\_vectorizer = TfidfVectorizer(analyzer = stemmed\_words, max\_features = 1000)

feature\_vector = tfidf\_vectorizer.fit(train\_data.Text)

train\_ds\_features = tfidf\_vectorizer.transform(train\_data.Text)

features = feature\_vector.get\_feature\_names()

Tf-IDF is continuous values and these continuous values associated with each class can be assumed to be distributed according to Gaussian distribution. So Gaussian Naive-Bayes can be used to classify these documents. We will GaussianNB, which implements the Gaussian Naive\_bayes algorithm for classification.

from sklearn.naive\_bayes import GaussianNB

train\_x, test\_x, train\_y, test\_y = train\_test\_split(train\_ds\_features, train\_data.Sentiment,

test\_size = 0.3, random\_state = 42)

nb\_clf = GaussianNB()

nb\_clf.fit(train\_x.toarray(), train\_y)

test\_ds\_predicted = nb\_clf.predict(test\_x.toarray())

print(metrics.classification\_report(test\_y,test\_ds\_predicted))

performance report | sentiment classification

cm = metrics.confusion\_matrix(test\_y, test\_ds\_predicted)

sn.heatmap(cm, annot=True, fmt = '.2f')

confusion matrix NB

The precision and recall seem to be pretty much the same. The accuracy is very high in this example as the dataset is clean and carefully curated. But it may not be the case in the real world.

**What Is One-Shot Learning in AI?**

One-shot learning is a machine learning-based (ML) algorithm that compares the similarities and differences between two images. The goal is simple: verify or reject the data being scanned. In human terms, it’s a computer vision approach for answering one question, is this person who they claim to be?

Unlike traditional computer vision projects, where models are trained on thousands of images or videos or detect objects from one frame to the next, one-shot classification operates using a limited amount of training set data to compare images against.

In many ways, one-shot algorithms are simpler than most computer vision models. However, other computer visions and ML models are given access to vast training set databases to improve accuracy and training outputs in comparison to one-shot or few-shot learning algorithms.

**One-Shot Learning in Context: N-Shot Learning**

In computer vision, a “shot” is the amount of data a model is given to solve a one-shot learning problem. In most cases, computer vision and any type of algorithmic model are fed vast amounts of data to train, learn from, and generate accurate outputs.

Similar to unsupervised or self-supervised learning, n-shot is when a model is trained on a scarce or very limited amount of data.

N-shot is a sliding scale, from zero examples (hence, zero-shot) up to few-shot; no more than 5 examples to train a model on. One-shot is when a model only has one example to learn from and make a decision on: E.g. when a passport scanner sees the image on a passport, it has to answer one question: “Is this image the same as the image in the corresponding database?”

N-shot learning (and every approach on this sliding scale; zero, one, or few-shot) is when a deep learning model with millions or billions of parameters is trained to deliver outputs, usually in real-time, with a limited amount of training examples.

In N-shot learning, we have:

N = labeled examples of;

K = each class (N∗K)

For every K, there is an example in the Support Set: S

Plus, there’s a Query Set: Q

**One-Shot vs. Few-Shot Learning**

Few-shot learning is one of the subsets of N-shot learning and is similar to few-shot with one key difference. Few-shot means a deep learning or machine learning model has more than one image to make a comparison against instead of only one image; hence that approach is one-shot.

Few-shot learning works the same way as one-shot, using deep learning networks and training examples to produce accurate outputs from a limited amount of comparative images. Few-shot can be trained on anywhere from two to five images. It still operates as a comparison-based approach, except the CV model making the comparison has access to slightly more data.

The few-shot learning concept was first introduced in 2019 at the Conference on Computer Vision and Pattern Recognition with the following paper: Meta-Transfer Learning for Few-Shot Learning.

This state-of-the-art data science research paved the way for new meta-learning transfer and training methods for deep neural networks. With these, Prototypical Networks are combined with the most popular deep learning models to generate impressive results quickly.

**One-Shot vs. Zero-Shot Learning**

One-shot is when a deep learning model is given one image to make a split-second decision on: a yes or no answer, in most cases.

As explained above, few-shot involves a slightly wider range of images, anywhere from two to five.

Zero-shot is when a model is expected to categorize unknown classes without any training data. In this case, we are asking a deep learning model to make a cognitive leap based on the metadata description of images. Imagine you’d never seen a lion in real life, online, in a book, or on TV. Someone describes a lion to you and then drops you in a nature reserve. Do you think you’d spot a lion compared to all of the other animals in the reserve?

That’s the same for zero-shot learning. Providing algorithmic models are trained on enough descriptions, then zero-shot is an effective method when data is scarce.

**One-Shot vs. Less Than One-Shot Learning**

Less than one-shot is another version of one-shot, similar to zero-shot whereby: “models must learn N new classes given only M < < N examples, and we show that this is achievable with the help of soft labels. We use a soft-label generalization of the k-Nearest Neighbors classifier to explore the intricate decision landscapes that can be created in the “less than one”-shot learning to set. We analyze these decision landscapes to derive theoretical lower bounds for separating N classes using M < N soft-label samples and investigate the robustness of the resulting systems.”

Image-based datasets for this approach are given soft labels, and then deep neural networks are trained on these descriptions to produce accurate outcomes.