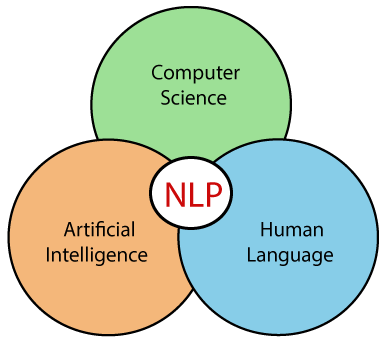
**Natural Language Processing (NLP)**  
Language is very important when we want to communicate with each other. Every human can talk and tell others what they need and listen with language. These languages can be anything like English, Spanish, Hindi, Malayalam, etc… We can express our ideas to others in this medium. Language is one of the critical components of human intelligence.

Every day we interact with humans, but how about non-human? We daily use our mobile phones, computers, cars, etc… How do these machines understand what we want to express with them? This is where we use “Natural Language Processing”. Machines cannot understand our natural language and hence it is much required to process the language in such a way it can understand and the vice-versa is also true.

**What is NLP?**

NLP stands for **Natural Language Processing**, which is a part of **Computer Science, Human language,** and **Artificial Intelligence**. It is the technology that is used by machines to understand, analyse, manipulate, and interpret human's languages. It helps developers to organize knowledge for performing tasks such as **translation, automatic summarization, Named Entity Recognition (NER), speech recognition, relationship extraction,** and **topic segmentation**.

  
  
**Understanding natural language processing**Programming languages, SQL Queries, and Scripting are other sets of structured language which we use. These languages are designed in such a way they are non-ambiguous in nature and can be processed easily. These are expressed by some rules or Grammars. These grammars are well defined and can be used while processing a language.

When we talk, mostly we try to use some languages and they may be structured. But mostly, unstructured. We don’t have many difficulties when we use ambiguities in our language while speaking and the unstructured nature isn’t difficult for us to process.   
  
**NLP Tasks**   
  
Several NLP tasks break down human text and voice data in ways that help the computer make sense of what it's ingesting(the process of absorbing information). Some of these tasks include the following:

* **Speech recognition**, also called speech-to-text, is the task of reliably converting voice data into text data. Speech recognition is required for any application that follows voice commands or answers spoken questions. What makes speech recognition especially challenging is the way people talk—quickly, slurring(ಅಸ್ಪಷ್ಟ ಧ್ವನಿ) words together, with varying emphasis and intonation(tone), in different accents, and often using incorrect grammar.
* **Part of speech tagging**, also called grammatical tagging, is the process of determining the part of speech of a particular word or piece of text based on its use and context. Part of speech identifies ‘make’ as a verb in ‘I can make a paper plane,’ and as a noun in ‘What make of car do you own?’
* **Word sense disambiguation** is the selection of the meaning of a word with multiple meanings  through a process of semantic analysis that determine the word that makes the most sense in the given context. For example, word sense disambiguation helps distinguish the meaning of the verb 'make' in ‘make the grade’ (achieve) vs. ‘make a bet’ (place).
* **Named entity recognition,**or NEM, identifies words or phrases as useful entities. NEM identifies ‘Kentucky’ as a location or ‘Fred’ as a man's name.
* **Co-reference resolution** is the task of identifying if and when two words refer to the same entity. The most common example is determining the person or object to which a certain pronoun refers (e.g., ‘she’ = ‘Mary’),  but it can also involve identifying a metaphor or an idiom in the text  (e.g., an instance in which 'bear' isn't an animal but a large hairy person).
* **Sentiment analysis**attempts to extract subjective qualities—attitudes, emotions, sarcasm, confusion, suspicion—from text.
* **Natural language generation**is sometimes described as the opposite of speech recognition or speech-to-text; it's the task of putting structured information into human language.

**NLP use cases**   
Natural language processing is the driving force behind machine intelligence in many modern real-world applications. Here are a few examples:

* **Spam detection:**You may not think of spam detection as an NLP solution, but the best spam detection technologies use NLP's text classification capabilities to scan emails for language that often indicates spam or phishing. These indicators can include overuse of financial terms, characteristic bad grammar, threatening language, inappropriate urgency, misspelled company names, and more. Spam detection is one of a handful of NLP problems that experts consider 'mostly solved' (although you may argue that this doesn’t match your email experience).
* **Machine translation:**Google Translate is an example of widely available NLP technology at work. Truly useful machine translation involves more than replacing words in one language with words of another.  Effective translation has to capture accurately the meaning and tone of the input language and translate it to text with the same meaning and desired impact in the output language. Machine translation tools are making good progress in terms of accuracy. A great way to test any machine translation tool is to translate text to one language and then back to the original.
* **Virtual agents and chatbots:** [Virtual agents](https://www.ibm.com/in-en/cloud/watson-assistant) such as Apple's Siri and Amazon's Alexa use speech recognition to recognize patterns in voice commands and natural language generation to respond with appropriate action or helpful comments. [Chatbots](https://www.ibm.com/in-en/topics/chatbots" \o "Follow link) perform the same magic in response to typed text entries. The best of these also learn to recognize contextual clues about human requests and use them to provide even better responses or options over time. The next enhancement for these applications is question answering, the ability to respond to our questions—anticipated or not—with relevant and helpful answers in their own words.
* **Social media sentiment analysis:**NLP has become an essential business tool for uncovering hidden data insights from social media channels. Sentiment analysis can analyze language used in social media posts, responses, reviews, and more to extract attitudes and emotions in response to products, promotions, and events–information companies can use in product designs, advertising campaigns, and more.
* **Text summarization:**Text summarization uses NLP techniques to digest huge volumes of digital text and create summaries and synopses for indexes, research databases, or busy readers who don't have time to read full text.

NLP approaches – rule based, statistical   
**What is rule-based approach in NLP?**A rule-based NLP system simply **follows these rules to categorise the language it's analysing**. As you can imagine, if the rule doesn't exist, the system will be unable to 'understand' the human language and thus will fail to categorise it. Unfortunately, this means accuracy is dependent on the rules provided.

**1. Rule-based**

Rule-based approaches are the oldest approaches to NLP. Why are they still used, you might ask? It's because they are tried and true, and have been proven to work well. Rules applied to text can offer a lot of insight: think of what you can learn about arbitrary text by finding what words are nouns, or what verbs end in -ing, or whether a pattern recognizable as Python code can be identified. [Regular expressions](https://en.wikipedia.org/wiki/Regular_expression) and [context free grammars](https://en.wikipedia.org/wiki/Context-free_grammar) are textbook examples of rule-based approaches to NLP.

Rule-based approaches:

* tend to focus on pattern-matching or parsing
* can often be thought of as "fill in the blanks" methods
* are low precision, high recall, meaning they can have high performance in specific use cases, but often suffer performance degradation when generalized.

What are the statistical approaches to NLP tasks?   
Statistical NLP comprises all quantitative approaches to automated language processing, including **probabilistic modeling, information theory, and linear algebra**.

Statistical approaches are characterized by:

* training data - in this case, a corpus with markup
* feature engineering - word type, surrounding words, capitalized, plural, etc.
* training a model on parameters, followed by fitting on test data (typical of machine learning systems in general)
* inference (applying model to test data) characterized by finding most probable words, next word, best category, etc.
* "semantic slot filling"

How to use dictionary?   
Dictionary in NLP means **a list of all the unique words occurring in the corpus**. If some words are repeated in different documents, they are all written just once as while creating the dictionary   
  
**Corpus**  
Corpus in NLP means the collection of text or other digital datasets across languages. Example – Wikipedia is a text corpus of an online encyclopedia.   
  
**A dictionary is a very important tool for anyone who is learning a new language. With a good dictionary you can do the following:**

* look up the meaning of an English word you see or hear
* find the English translation of a word in your language
* check the spelling of a word
* check the plural of a noun or past tense of a verb
* find out other grammatical information about a word
* find the synonym or antonym of a word
* look up the collocations of a word.( collocation refers to **a natural combination of words that are closely affiliated with each other**. Some examples are "pay attention", "fast food", "make an effort", and "powerful engine".)
* check the part of speech of a word
* find out how to say a word
* find out about the register of a word
* find examples of the use of a word in natural language

Commonly used NLP tools & libraries

**NLP Libraries**

* Natural Language Toolkit (NLTK) NLTK is one of the leading platforms for building Python programs that can work with human language data. ...
* Gensim. ...
* CoreNLP. ...
* spaCy. ...
* TextBlob. ...
* Pattern. ...
* PyNLPl.

What are the tools of NLP?

Basically, you can start using NLP tools through SaaS (software as a service) tools or open-source libraries.

SaaS tools are ready-to-use and powerful cloud-based solutions that can be implemented with low or no code. SaaS platforms often offer pre-trained NLP models that can be used code-free, and APIs that are geared more towards those who want a more flexible, low-code, option, e.g. professional developers, or those learning to code, who want to simplify their work.

To build your own NLP models with open-source libraries, you’ll need time to build infrastructures from scratch, and you’ll need money to invest in devs if you don’t already have an in-house team of experts.

Now that you have an idea of what’s available, tune into our list of top SaaS tools and NLP libraries.

**The NLP Tools**

* MonkeyLearn | NLP made simple.
* Aylien | Leveraging news content with NLP.
* IBM Watson | A pioneer AI platform for businesses.
* Google Cloud NLP API | Google technology applied to NLP.
* Amazon Comprehend | An AWS service to get insights from text.
* NLTK | The most popular Python library.

### 1. MonkeyLearn

[MonkeyLearn](https://monkeylearn.com/) is a user-friendly, NLP-powered platform that helps you gain valuable insights from your text data.

To get started, you can try one of the [pre-trained models](https://app.monkeylearn.com/main/explore/), to perform text analysis tasks such as sentiment analysis, topic classification, or keyword extraction. For more accurate insights, you can build a [customized machine learning model](https://app.monkeylearn.com/main/module-create/wizard/choose-module-type/) tailored to your business.

Once you’ve trained your models to deliver accurate insights, you can connect your text analysis models to your favorite apps (like Google Sheets, Zendesk, Excel or Zapier) using our [integrations](https://monkeylearn.com/integrations/) (no coding skills needed!), or through [MonkeyLearn’s APIs, available in all major programming languages.](https://monkeylearn.com/api/v3/" \t "_blank)

### 2. Aylien

[Aylien](https://aylien.com/) is a SaaS API that uses deep learning and NLP to analyze large volumes of text-based data, such as academic publications, real-time content from news outlets and social media data. You can use it for NLP tasks like text summarization, article extraction, entity extraction, and sentiment analysis, among others.

### 3. IBM Watson

[IBM Watson](https://www.ibm.com/watson) is a suite of AI services stored in the IBM Cloud. One of its key features is Natural Language Understanding, which allows you to identify and extract keywords, categories, emotions, entities, and more.

 It’s versatile, in that it can be tailored to different industries, from healthcare to finance, and has a trove of documents to help you get started.

### 4. Google Cloud

The [Google Cloud Natural Language API](https://cloud.google.com/natural-language) provides several pre-trained models for sentiment analysis, content classification, and entity extraction, among others. Also, it offers AutoML Natural Language, which allows you to build customized machine learning models.

As part of the Google Cloud infrastructure, it uses Google question-answering and language understanding technology.

### 5. Amazon Comprehend

[Amazon Comprehend](https://aws.amazon.com/comprehend/) is an NLP service, integrated with the Amazon Web Services infrastructure. You can use this API for NLP tasks such as sentiment analysis, topic modeling, entity recognition, and more.

For those that work in healthcare, there’s a specialized variant: the Amazon Comprehend Medical, which allows you to perform advanced analysis of medical data using Machine Learning.

### 6. NLTK

The [Natural Language Toolkit (NLTK)](https://www.nltk.org/) with Python is one of the leading tools in NLP model building. Focused on research and education in the NLP field, NLTK is bolstered by an active community, as well as a range of [tutorials for language processing](https://pythonprogramming.net/tokenizing-words-sentences-nltk-tutorial/), sample datasets, and resources that include a comprehensive [Language Processing and Python handbook](http://www.nltk.org/book/ch01.html).

Although it takes a while to master this library, it’s considered an amazing playground to get hands-on NLP experience. With a modular structure, NLTK provides plenty of components for NLP tasks, like tokenization, tagging, stemming, parsing, and classification, among others.

### 7. Stanford Core NLP

[Stanford Core NLP](https://stanfordnlp.github.io/CoreNLP/) is a popular library built and maintained by the NLP community at Stanford University. It’s written in Java ‒ so you’ll need to install JDK on your computer ‒ but it has APIs in most programming languages.

The Core NLP toolkit allows you to perform a variety of NLP tasks, such as part-of-speech tagging, tokenization, or named entity recognition. Some of its main advantages include scalability and optimization for speed, making it a good choice for complex tasks.

### 8. TextBlob

[TextBlob](https://textblob.readthedocs.io/en/dev/) is a Python library that works as an extension of NLTK, allowing you to perform the same NLP tasks in a much more intuitive and user-friendly interface. Its learning curve is more simple than with other open-source libraries, so it’s an excellent choice for beginners, who want to tackle NLP tasks like sentiment analysis, text classification, part-of-speech tagging, and more.

### 9. SpaCy

One of the newest open-source Natural Language Processing with Python libraries on our list is [SpaCy](https://spacy.io/usage/spacy-101" \t "_blank). It’s lightning-fast, easy to use, well-documented, and designed to support large volumes of data, not to mention, boasts a series of [pretrained NLP models](https://spacy.io/models" \t "_blank) that make your job even easier. Unlike NLTK or CoreNLP, which display a number of algorithms for each task, SpaCy keeps its menu short and serves up the best available option for each task at hand.

This library is a great option if you want to prepare text for deep learning, and excels at extraction tasks. For the moment, it’s only available in English.

### 10. GenSim

## [Gensim](https://github.com/RaRe-Technologies/gensim) is a highly specialized Python library that largely deals with topic modeling tasks using algorithms like Latent Dirichlet Allocation (LDA). It’s also excellent at recognizing text similarities, indexing texts, and navigating different documents. NLP Libraries

### 1. Natural Language Toolkit (NLTK)

NLTK is one of the leading platforms for building Python programs that can work with human language data. It presents a practical introduction to programming for language processing. NLTK comes with a host of text processing libraries for sentence detection, tokenization, lemmatization, stemming, parsing, chunking, and POS tagging.

NLTK provides easy-to-use interfaces to over 50 corpora and lexical resources. The tool has the essential functionalities required for almost all kinds of natural language processing tasks with Python.

### 2. Gensim

Gensim is a Python library designed specifically for “topic modeling, document indexing, and similarity retrieval with large corpora.” All algorithms in Gensim are memory-independent, w.r.t., the corpus size, and hence, it can process input larger than RAM. With intuitive interfaces, Gensim allows for efficient multicore implementations of popular algorithms, including online Latent Semantic Analysis (LSA/LSI/SVD), Latent Dirichlet Allocation (LDA), Random Projections (RP), Hierarchical Dirichlet Process (HDP) or word2vec deep learning.

Gensim features extensive documentation and Jupyter Notebook tutorials. It largely depends on NumPy and SciPy for scientific computing. Thus, you must install these two Python packages before installing Gensim.

**FYI:** [Free Deep Learning Course](https://www.upgrad.com/blog/deep-learning-free-online-course/)!

### 3. CoreNLP

Stanford CoreNLP comprises of an assortment of human language technology tools. It aims to make the application of linguistic analysis tools to a piece of text easy and efficient. With CoreNLP, you can extract all kinds of text properties (like named-entity recognition, part-of-speech tagging, etc.) in only a few lines of code.

Since CoreNLP is written in Java, it demands that Java be installed on your device. However, it does offer programming interfaces for many popular programming languages, including Python. The tool incorporates numerous Stanford’s NLP tools like the parser, [sentiment analysis](https://www.upgrad.com/blog/sentiment-analysis-what-is-it-and-why-does-it-matter/), bootstrapped pattern learning, part-of-speech (POS) tagger, named entity recognizer (NER), and coreference resolution system, to name a few. Furthermore, CoreNLP supports four languages apart from English – Arabic, Chinese, German, French, and Spanish.

### 4. spaCy

spaCy is an open-source NLP library in Python. It is designed explicitly for production usage – it lets you develop applications that process and understand huge volumes of text.

spaCy can preprocess text for Deep Learning. It can be be used to build natural language understanding systems or information extraction systems. spaCy is equipped with pre-trained statistical models and word vectors. It can support tokenization for over 49 languages. spaCy boasts of state-of-the-art speed, parsing, named entity recognition, convolutional neural network models for tagging, and deep learning integration.

### ****5.**** TextBlob

TextBlob is a Python (2 & 3) library designed for processing textual data. It focuses on providing access to common text-processing operations through familiar interfaces. TextBlob objects can be treated as Python strings that are trained in Natural Language Processing.

TextBlob offers a neat API for performing common NLP tasks like part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, language translation, word inflection, parsing, n-grams, and WordNet integration.

### 6. Pattern

Pattern is a text processing, web mining, natural language processing, machine learning, and network analysis tool for Python. It comes with a host of tools for data mining (Google, Twitter, Wikipedia API, a web crawler, and an HTML DOM parser), NLP (part-of-speech taggers, n-gram search, sentiment analysis, WordNet), ML (vector space model, clustering, SVM), and network analysis by graph centrality and visualization.

Pattern can be a powerful tool both for a scientific and a non-scientific audience. It has a simple and straightforward syntax – the function names and parameters are chosen in a way so that the commands are self-explanatory. While Pattern is a highly valuable learning environment for students, it serves as a rapid development framework for web developers.

*Get the*[*Machine Learning Certification*](https://www.upgrad.com/machine-learning-course/?utm_source=BLOG&utm_medium=BODY&utm_campaign=DV_ML_BLOG_BODY_VP_73362)*online from the World’s top Universities. Earn Masters, Executive PGP, or Advanced Certificate Programs to fast-track your career.*

### 7. PyNLPl

Pronounced as ‘pineapple,’ PyNLPl is a Python library for Natural Language Processing. It contains a collection of custom-made Python modules for Natural Language Processing tasks. One of the most notable features of PyNLPl is that it features an extensive library for working with FoLiA XML (Format for Linguistic Annotation).

PyNLPl is segregated into different modules and packages, each useful for both standard and advanced NLP tasks. While you can use PyNLPl for basic NLP tasks like extraction of n-grams and frequency lists, and to build a simple language model, it also has more complex data types and algorithms for advanced NLP tasks.