What is NLP?

NLP is a form of AI that gives computers the ability to read, understand and interpret human language. It helps computers to measure sentiments and determine which part of human language are important.

Text Cleaning in Python

In [1]:
```
import warnings
warnings.filterwarnings('ignore')
```

Creating bunch of sentences. I have intentionally corrupt the sentence.

In [6]:
```
#Creating bunch of sentences. i have intentionally corrupt the sentence.
raw_docs = ["I am writing some very basic english sentences",
"I'm just writing it for the demo PURPOSE to make audience understand the basi
cs .",
"The point is to _learn HOW it works_ on #simple # data."]
```

In [2]:
```
import nltk
```

In [3]:
```
nltk.download()
```

```
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.
xml
```

Out[3]:
```
True
```

Step 1

Convert everything into lower case

In [8]:
```
#import string
raw_docs = [doc.lower() for doc in raw_docs]
print(raw_docs)
```

```
['i am writing some very basic english sentences', "i'm just writing it for t
he demo purpose to make audience understand the basics .", 'the point is to _
learn how it works_ on #simple # data.']
```

Step 2 - Tokenization

The process of breaking down a text paragraphs in smaller chunks such as words or sentence is called Tokenization.

In [14]:
```
#word tokenize
from nltk.tokenize import word_tokenize
tokenized_words = [word_tokenize(doc) for doc in raw_docs]
print('tokenized_words:', tokenized_words)
print()
#Sentence tokenization
from nltk.tokenize import sent_tokenize
sent_tokenize = [sent_tokenize(doc) for doc in raw_docs]
print('tokenize_sentence:', sent_tokenize)
```

```
tokenized_words: [['i', 'am', 'writing', 'some', 'very', 'basic', 'english',
'sentences'], ['i', "'m", 'just', 'writing', 'it', 'for', 'the', 'demo', 'pur
pose', 'to', 'make', 'audience', 'understand', 'the', 'basics', '.'], ['the',
'point', 'is', 'to', '_learn', 'how', 'it', 'works_', 'on', '#', 'simple',
'#', 'data', '.']]

tokenize_sentence: [['i am writing some very basic english sentences'], ["i'm
just writing it for the demo purpose to make audience understand the basics
."], ['the point is to _learn how it works_ on #simple # data.']]
```

Step 3 - Punctuation Removal

In [19]:
```
#import string
import re
regex = re.compile('[%s]' % re.escape(string.punctuation))

tokenized_docs_no_punctuation = []

for review in tokenized_words:
    new_review = []
    for token in review:
        new_token = regex.sub(u'', token)
        if not new_token == u'':
            new_review.append(new_token)

    tokenized_docs_no_punctuation.append(new_review)

print(tokenized_docs_no_punctuation)
```

```
[['i', 'am', 'writing', 'some', 'very', 'basic', 'english', 'sentences'],
['i', 'm', 'just', 'writing', 'it', 'for', 'the', 'demo', 'purpose', 'to', 'm
ake', 'audience', 'understand', 'the', 'basics'], ['the', 'point', 'is', 't
o', 'learn', 'how', 'it', 'works', 'on', 'simple', 'data']]
```

Step 4 - Removing Stopwords

Stop words are considered as noise in the text. We would not want these words taking up space in our database or taking up the processing time.

In [35]:
```
nltk.download('stopwords')
from nltk.corpus import stopwords

tokenized_docs_no_stopwords = []

for doc in tokenized_docs_no_punctuation:
    new_term_vector = []
    for word in doc:
        if not word in stopwords.words('english'):
            new_term_vector.append(word)

    tokenized_docs_no_stopwords.append(new_term_vector)

print(tokenized_docs_no_stopwords)
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\admin\AppData\Roaming\nltk_data...

[['writing', 'basic', 'english', 'sentences'], ['writing', 'demo', 'purpose',
'make', 'audience', 'understand', 'basics'], ['point', 'learn', 'works', 'sim
ple', 'data']]

[nltk_data]   Package stopwords is already up-to-date!
```

Step 5- Stemming and Lemmantization

Lemmatization is a process of converting a word to its base form. The difference between stemming and lemmatization is, lemmatization considers the context and converts the words to its meaningful baseform, whereas stemming just removes the last few characters often leading to incorrect meaning and spelling errors.

In [40]:
```
nltk.download('wordnet')
from nltk.stem.porter import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer

porter = PorterStemmer()
wordnet = WordNetLemmatizer()

preprocessed_docs = []

for doc in tokenized_docs_no_stopwords:
    final_doc = []
    for word in doc:
        #final_doc.append(porter.stem(word))
        final_doc.append(wordnet.lemmatize(word))

    preprocessed_docs.append(final_doc)

print(preprocessed_docs)
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\admin\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\wordnet.zip.

[['writing', 'basic', 'english', 'sentence'], ['writing', 'demo', 'purpose',
'make', 'audience', 'understand', 'basic'], ['point', 'learn', 'work', 'simpl
e', 'data']]
```

(Source : Unfold Datascience youtube channel)