**Text Blob**

* Text Blob is a python library and offers a simple API to access its methods and perform basic NLP tasks.
* A good thing about TextBlob is that they are just like python strings. So, you can transform and play with it same like we did in python
* TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.
* The simplest way to install TextBlob is by PyPI:

$ pip install -U textblob  
$ python -m textblob.download\_corpora

* TextBlob support following [text processing](http://textprocessing.org/) features, including:

Word Tokenization  
Sentence Tokenization  
Part-of-speech tagging  
Noun phrase extraction  
Sentiment analysis  
Word Pluralization  
Word Singularization  
Spelling correction  
Parsing  
Classification (Naive Bayes, Decision Tree)  
Language translation and detection powered by Google Translate  
Word and phrase frequencies  
n-grams  
Word inflection (pluralization and singularization) and lemmatization  
JSON serialization  
Add new models or languages through extensions  
WordNet integration

Tokenization

Tokenization refers to splitting a large paragraph into sentences or words. Typically, a token refers to a word in a text document. Tokenization is pretty straight forward with TextBlob. All you have to do is import the TextBlob object from the textblob library, pass it the document that you want to tokenize, and then use the sentences and words attributes to get the tokenized sentences and attributes

[Word Tokenization](http://textanalysisonline.com/textblob-word-tokenize)

You can tokenize Text Blobs into words

[Sentence Tokenization](http://textanalysisonline.com/textblob-sentence-segmentation)

TextBlob can be used to segment sentence from text paragraph

[Part-of-Speech Tagging](http://textanalysisonline.com/textblob-pos-tagging)

You can use TextBlob to get POS Tagging result, which can be accessed through the tags property.

Part-of-speech tagging or grammatical tagging is a method to mark words present in a text on the basis of its definition and context. In simple words, it tells whether a word is a noun, or an adjective, or a verb, etc. This is just a complete version of noun phrase extraction, where we want to find all the the parts of speech in a sentence.

Parsing

Use the [parse()](https://textblob.readthedocs.io/en/dev/api_reference.html#textblob.blob.TextBlob.parse) method to parse the text.

WordNet Integration

You can access the synsets for a [Word](https://textblob.readthedocs.io/en/dev/api_reference.html#textblob.blob.Word) via the [synsets](https://textblob.readthedocs.io/en/dev/api_reference.html" \l "textblob.blob.Word.synsets" \o "textblob.blob.Word.synsets) property or the [get\_synsets](https://textblob.readthedocs.io/en/dev/api_reference.html" \l "textblob.blob.Word.get_synsets" \o "textblob.blob.Word.get_synsets) method, optionally passing in a part of speech.

You can access the definitions for each synset via the [definitions](https://textblob.readthedocs.io/en/dev/api_reference.html#textblob.blob.Word.definitions) property or the [define()](https://textblob.readthedocs.io/en/dev/api_reference.html#textblob.blob.Word.define) method, which can also take an optional part-of-speech argument.

[Noun Phrase Extraction](http://textanalysisonline.com/textblob-noun-phrase-extraction)

You can get noun phrases by accessing through the noun\_phrases property in TextBlob

Noun phrase extraction, as the name suggests, refers to extracting phrases that contain nouns. To find noun phrases, you simply have to use the noun\_phrase attributes on the TextBlob object.

[Sentiment Analysis](http://textanalysisonline.com/textblob-sentiment-analysis)

Sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials

Sentiment analysis is basically the process of determining the attitude or the emotion of the writer, i.e., whether it is positive or negative or neutral.

The *sentiment* function of textblob returns two properties, polarity, and subjectivity.

Polarity is float which lies in the range of [-1,1] where 1 means positive statement and -1 means a negative statement. Subjective sentences generally refer to personal opinion, emotion or judgment whereas objective refers to factual information. Subjectivity is also a float which lies in the range of [0,1].

Words Lemmatization

Lemmatization refers to reducing the word to its root form as found in a dictionary.

To perform lemmatization via TextBlob, you have to use the Word object from the textblob library, pass it the word that you want to lemmatize and then call the lemmatize method.

Words can be lemmatized by the lemmatize method, but notice that the Text Blog lemmatize method is inherited from [NLTK Word Lemmatizer](http://textanalysisonline.com/nltk-wordnet-lemmatizer), and the default POS Tag is “n”, if you want lemmatize other pos tag words, you need specify it

Words Inflection and Lemmatization

*Inflection* is a process of *word* formation in which characters are added to the base form of a *word* to express grammatical meanings. Word inflection in TextBlob is very simple, i.e., the words we tokenized from a textblob can be easily changed into singular or plural.

TextBlob library also offers an in-build object known as Word. We just need to create a word object and then apply a function directly to it

We can also use the tags to inflect a particular type of words

[Spelling Correction](http://textanalysisonline.com/textblob-spelling-correction)

Spelling correction is a cool feature which TextBlob offers, we can be accessed using the correct function

Word objects also have a spellcheck(),Word.spellcheck()  method that returns a list of (word, confidence) tuples with spelling suggestions

 Translation and Language Detection

Text Blob’s translation and language detection feature are based on Google’s API

One of the most powerful capabilities of the TextBlob library is to translate from one language to another. On the backend, the TextBlob language translator uses the [Google Translate API](https://cloud.google.com/translate/docs/)

To translate from one language to another, you simply have to pass the text to the TextBlob object and then call the translate method on the object. The language code for the language that you want your text to be translated to is passed as a parameter to the method.

Finally, using the detect\_language method, you can also detect the language of the sentence.

### N-grams

A combination of multiple words together are called N-Grams. N grams (N > 1) are generally more informative as compared to words, and can be used as features for language modelling.  N-grams can be easily accessed in TextBlob using the *ngrams* function, which returns a tuple of n successive words.

The [TextBlob.ngrams()](https://textblob.readthedocs.io/en/dev/api_reference.html#textblob.blob.TextBlob.ngrams) method returns a list of tuples of n successive words.

### Get Word and Noun Phrase Frequencies

### There are two ways to get the frequency of a word or noun phrase in a [TextBlob](https://textblob.readthedocs.io/en/dev/api_reference.html#textblob.blob.TextBlob). The first is through the word\_counts dictionary.

If you access the frequencies this way, the search will not be case sensitive, and words that are not found will have a frequency of 0.

The second way is to use the count() method.

To find the frequency of occurrence of a particular word, we have to pass the name of the word as the index to the word\_counts list of the TextBlob object.

Another way is to simply call the count method on the words attribute, and pass the name of the word whose frequency of occurrence is to be found

It is important to mention that by default the search is not case-sensitive. If you want your search to be case sensitive, you need to pass True as the value for the case\_sensitive parameter,

Like word counts, noun phrases can also be counted in the same way.

### Get Start and End Indices of Sentences

Use sentence.start and sentence.end to get the indices where a sentence starts and ends within a [TextBlob](https://textblob.readthedocs.io/en/dev/api_reference.html#textblob.blob.TextBlob).

### Converting to Upper and Lowercase

Text Blob objects are very similar to strings. You can convert them to upper case or lower case, change their values, and concatenate them together as well.

Text Classification

TextBlob also provides basic text classification capabilities. However, if you have a really limited data and you want to quickly develop a very basic text classification model, then you may use TextBlob. For advanced models, I would recommend machine learning libraries such as Scikit-Learn or Tensorflow.

The first thing we need is a training dataset and test data. The classification model will be trained on the training dataset and will be evaluated on the test dataset.

You can also make a prediction by passing our classifier to the classifier parameter of the TextBlob object. You then have to call the classify method on the TextBlob object to view the prediction.

Finally, to find the accuracy of your algorithm on the test set, call the accuracy method on your classifier and pass it the test\_data that we just created.

To find the most important features for the classification, the show\_informative\_features method can be used. The number of most important features to see is passed as a parameter.

Pros:

1. Since, it is built on the shoulders of NLTK and Pattern, therefore making it simple for beginners by providing an intuitive interface to NLTK.
2. It provides language translation and detection which is powered by Google Translate ( not provided with Spacy).

Cons:

1. It is little slower in the comparison to spacy but faster than NLTK. (Spacy > TextBlob > NLTK)
2. It does not provide features like dependency parsing, word vectors etc. which is provided by spacy.