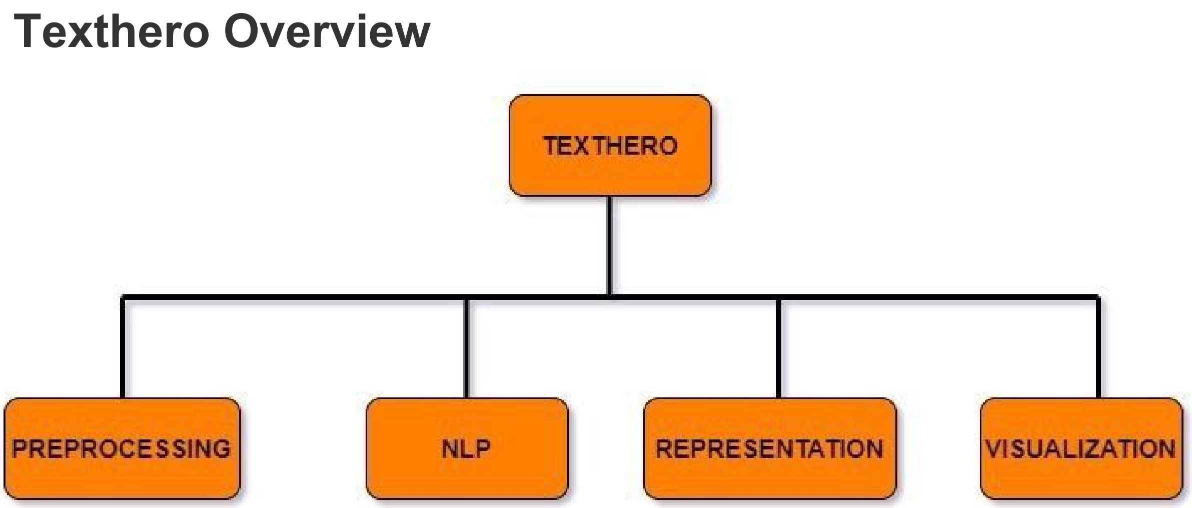
# TextHero : A Python ToolKit for Text Processing

## Intro

Texthero is a simple Python toolkit that helps you work with a text-based dataset. It provides quick and easy functionalities that let you preprocess, represent, map into vectors, and visualize text data in just a couple of lines of code.Texthero is designed to be used on top of pandas, so it makes it easier to preprocess and analyze text-based Pandas Series or Dataframes.If you are working on an NLP project, Texthero can help you get things done faster than before and gives you more time to focus on important tasks.

**NOTE:** The Texthero library is still in the beta version. You might face some bugs and pipelines might change. A faster and better version will be released and it will bring some major changes.

## Description



Texthero has four useful modules that handle different functionalities that you can apply in your text-based dataset.

1. **Preprocessing**
   This module allows for the efficient pre-processing of text-based Pandas Series or DataFrames. It has different methods to clean your text dataset such as lowercase(), remove_html_tags() and remove_urls().

2. **NLP**
   This module has a few NLP tasks such as named_entities, noun_chunks, and so on.
3. **Representation**
   This module has different algorithms to map words into vectors such as TF-IDF, GloVe, Principal Component Analysis(PCA), and term_frequency.
4. **Visualization**
   The last module has three different methods to visualize the insights and statistics of a text-based Pandas DataFrame. It can plot a scatter plot and word cloud.

Like any other library, we first need to install texthero using **pip install texthero.**

## 1. Importing required libraries

We will be importing texthero for text processing and pandas for loading the dataset and manipulating it.

```
import pandas as pd
```

```
import texthero as hero
```

## 2. Loading the dataset

The dataset we will be using here can be downloaded from Kaggle. This dataset contains certain attributes which we will analyze but we will mainly focus on the ‘content’ column.

```
df = pd.read_csv(‘text.csv’)
```

```
df
```

## 3. Processing the dataset

We can see that our dataset contains a sentiment analysis of tweets of different authors. We will focus on the tweets and will try and apply different functions used for text processing using Texthero.

We will start by cleaning the text in the ‘content’ column which is the tweets by the users. We will clean the text and store it in a new column.

## 1. Preprocessing the Text

### • Cleaning the text

```
df['clean_content'] = hero.clean(df['content'])
```

```
df[‘clean_content’].head()
```

The clean function has certain defined properties which like, it removes all stopwords, punctuations, digits, whitespaces, etc. Also, it converts the text into all lowercase. We can use all these functions separately according to our wish.

### • Tokenize the text

Tokenize function returns a pandas series where each row contains a list of tokens

```
hero.tokenize(df['clean_content'])
```

### • Stemming

Stemming means removing the end of words with a heuristic process. Stem function makes use of two NLTK stemming algorithms known as Snowball Stemmer and Porter Stemmer.

```
hero.stem(df['clean_content'], stem=’snowball’)
```

## 2. Visualize the Cleaned Text

There are many ways of visualizing the textual data, here we will use ‘Wordcloud’ to visualize the cleaned data we created.

```
hero.visualization.wordcloud(df['clean_content'], width= 250,
height = 150, max_words=200, background_color='WHITE')
```

Similarly, we can visualize the most frequently used words or the top used words using the top_words visualization by TextHero.

```
hero.visualization.top_words(df['clean_content'])
```

### 3. NLP Operations on Text

Now we will implement some of the NLP operations provided by TextHero on our data.

- Named Entities

Named entities function returns a Pandas Series where each row contains a list of tuples containing information regarding the given named entities. We will be using the spacy as a package here.

```
hero.named_entities(df['clean_content'], package='spacy')
```

- Noun Chunks

It returns a group of consecutive word that belongs together. As our dataset is pretty large so we will analyze the noun chunks in only 100 rows.

```
hero.noun_chunks(df['clean_content'][:100])
```

## Conclusion

Although the library is still in beta, I see a promising future for Texthero and hope it gets the love it deserves. It makes cleaning and preparing text in Pandas dataframes a breeze.

I hope a few more visualization options get added, but the scatter plot is a great start. If you’re interested in learning more about Natural Language Processing, check out my other articles covering some basic and advanced topics.