Reference:

- https://www.youtube.com/watch?v=eFdPGpny_hY&t=98s (https://www.youtube.com/watch?v=eFdPGpny_hY&t=98s)
- Added some notes and made some changes

In [1]:
```
import sys,tweepy,csv,re
#the following for stentiment analysis
from textblob import TextBlob
import matplotlib.pyplot as plt
```

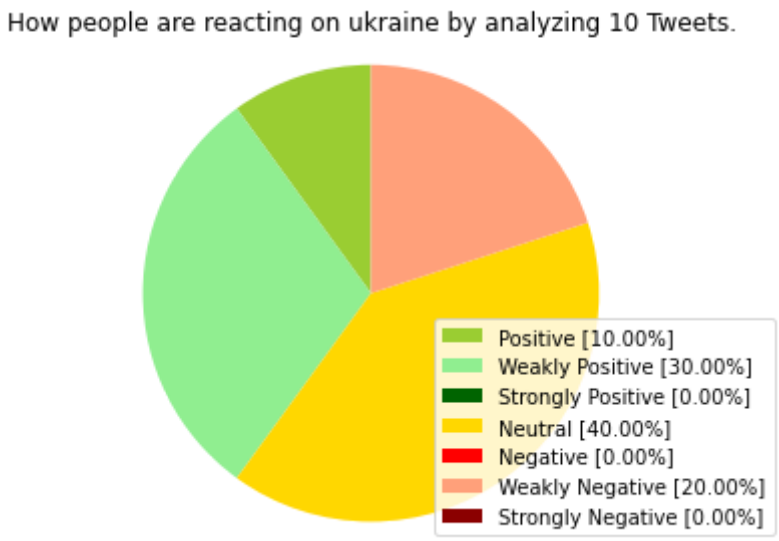
In [ ]:
```
from credentials import *
#create a file called credentials.py make sure it is in the same folder as
# the credentials file will look like this
#ACCESS_TOKEN = 'xxx'
#ACCESS_SECRET = 'xx'
#CONSUMER_KEY = 'xx'
#CONSUMER_SECRET = 'xx'
```

In [3]:
```
class SentimentAnalysis:

    def __init__(self):
        self.tweets = []
        self.tweetText = []

    def DownloadData(self):
        # authenticating

        auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
        auth.set_access_token(ACCESS_TOKEN, ACCESS_SECRET)

        twitter_api = tweepy.API(auth)


        # input for term to be searched and how many tweets to search
        searchTerm = input("Enter Keyword/Tag to search about: ")
        NoOfTerms = int(input("Enter how many tweets to search: "))

        # searching for tweets
        self.tweets = tweepy.Cursor(twitter_api.search_tweets, q=searchTerm

        # Open/create a file to append data to
        csvFile = open('result.csv', 'a')

        # Use csv writer
        csvWriter = csv.writer(csvFile)


        # creating some variables to store info
        polarity = 0
        positive = 0
        wpositive = 0
        spositive = 0
        negative = 0
        wnegative = 0
        snegative = 0
        neutral = 0

         # iterating through tweets fetched
        for tweet in self.tweets:
            #Append to temp so that we can store in csv later. I use encode
            self.tweetText.append(self.cleanTweet(tweet.text).encode('utf-8
            # print (tweet.text.translate(non_bmp_map))    #print tweet's t
            analysis = TextBlob(tweet.text)
            # print(analysis.sentiment)  # print tweet's polarity
            polarity += analysis.sentiment.polarity  # adding up polarities

            if (analysis.sentiment.polarity == 0):  # adding reaction of ho
                neutral += 1
            elif (analysis.sentiment.polarity > 0 and analysis.sentiment.po
                wpositive += 1
            elif (analysis.sentiment.polarity > 0.3 and analysis.sentiment.
                positive += 1
            elif (analysis.sentiment.polarity > 0.6 and analysis.sentiment.
                spositive += 1
```

```
            elif (analysis.sentiment.polarity > -0.3 and analysis.sentiment
                wnegative += 1
            elif (analysis.sentiment.polarity > -0.6 and analysis.sentiment
                negative += 1
            elif (analysis.sentiment.polarity > -1 and analysis.sentiment.p
                snegative += 1
        # Write to csv and close csv file
        csvWriter.writerow(self.tweetText)
        csvFile.close()

        # finding average of how people are reacting
        positive = self.percentage(positive, NoOfTerms)
        wpositive = self.percentage(wpositive, NoOfTerms)
        spositive = self.percentage(spositive, NoOfTerms)
        negative = self.percentage(negative, NoOfTerms)
        wnegative = self.percentage(wnegative, NoOfTerms)
        snegative = self.percentage(snegative, NoOfTerms)
        neutral = self.percentage(neutral, NoOfTerms)

        # finding average reaction
        polarity = polarity / NoOfTerms

        # printing out data
        print("How people are reacting on " + searchTerm + " by analyzing "
        print()
        print("General Report: ")

        if (polarity == 0):
            print("Neutral")
        elif (polarity > 0 and polarity <= 0.3):
            print("Weakly Positive")
        elif (polarity > 0.3 and polarity <= 0.6):
            print("Positive")
        elif (polarity > 0.6 and polarity <= 1):
            print("Strongly Positive")
        elif (polarity > -0.3 and polarity <= 0):
            print("Weakly Negative")
        elif (polarity > -0.6 and polarity <= -0.3):
            print("Negative")
        elif (polarity > -1 and polarity <= -0.6):
            print("Strongly Negative")

        print()
        print("Detailed Report: ")
        print(str(positive) + "% people thought it was positive")
        print(str(wpositive) + "% people thought it was weakly positive")
        print(str(spositive) + "% people thought it was strongly positive")
        print(str(negative) + "% people thought it was negative")
        print(str(wnegative) + "% people thought it was weakly negative")
        print(str(snegative) + "% people thought it was strongly negative")
        print(str(neutral) + "% people thought it was neutral")

        self.plotPieChart(positive, wpositive, spositive, negative, wnegati

    def cleanTweet(self, tweet):
        # Remove Links, Special Characters etc from tweet
        return ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z \t]) | (\w +:\
```

```
    # function to calculate percentage
    def percentage(self, part, whole):
        temp = 100 * float(part) / float(whole)
        return format(temp, '.2f')

    def plotPieChart(self, positive, wpositive, spositive, negative, wnegat
        labels = ['Positive [' + str(positive) + '%]', 'Weakly Positive ['
                  'Negative [' + str(negative) + '%]', 'Weakly Negative ['
        sizes = [positive, wpositive, spositive, neutral, negative, wnegati
        colors = ['yellowgreen','lightgreen','darkgreen', 'gold', 'red','li
        patches, texts = plt.pie(sizes, colors=colors, startangle=90)
        plt.legend(patches, labels, loc="best")
        plt.title('How people are reacting on ' + searchTerm + ' by analyzi
        plt.axis('equal')
        plt.tight_layout()
        plt.show()

if __name__== "__main__":
    sa = SentimentAnalysis()
    sa.DownloadData()
```

```
Enter Keyword/Tag to search about: ukraine
Enter how many tweets to search: 10
How people are reacting on ukraine by analyzing 10 tweets.

General Report:
Weakly Positive

Detailed Report:
10.00% people thought it was positive
30.00% people thought it was weakly positive
0.00% people thought it was strongly positive
0.00% people thought it was negative
20.00% people thought it was weakly negative
0.00% people thought it was strongly negative
40.00% people thought it was neutral
```



In [ ]: