```
class StdOutListener(StreamListener):
    def __init__(self, api=None):
        super(StdOutListener, self).__init__()
        self.num_alltweets = 0
        self.num_print = 2
        self.count = 0
        self.tstart = time.time()
        print( tf_filter, Sf_max, Sf_save_num, Sf_save_sec, tf_text)

    def on_status(self, status):
        self.num_alltweets += 1
        self.count += 1

        if self.num_alltweets <= Sf_max :
            ft_save.write(status.text+'\n')

            time_past = (time.time() - self.tstart)

            if self.count >= Sf_save_num or time_past > Sf_save_sec :
                print('self.count',self.count,'Sf_save_num',Sf_save_num,
                      'time_past',time_past,'Sf_save_sec',Sf_save_sec)
                self.count = 0
                self.tstart = time.time()

                tf_filter_str = '_'.join(map(str, tf_filter))
                createWordFile(self.num_alltweets, 'F_'+ tf_filter_str )
            return True
        else:
            print("Time limit or Max Tweet Recieved")
            return False

    def on_error(self, status):
        print('on_error(self, status):', status,self )
        return True

def getTweetsByTrack():
    #This handles Twitter authetification and the connection to Twitter Streaming API
    try:
        auth = OAuthHandler(consumer_key, consumer_secret)
        auth.set_access_token(access_token, access_token_secret)
        if tfscr_name != '' :
            api = tweepy.API(auth)                    ## Get all tweeets for give tfscr_name on the setup file
            tw_tot = getTweetsByScreenName(api)
        else:
            l = StdOutListener()                      ## Get new tweets
            stream = Stream(auth, l)
            if tf_filter == ['']:
                stream.sample()
            else:
                stream.filter(track = tf_filter )
            ft_save.close()
```