Title

Emil Rask Maagaard - s142955

Abstract

****
We describe a lightweight webservice that performs online topic mining with sentiment analyze using standard components of Python. It can analyze a small corpus on a few hundred small documents in a few hundred milliseconds.
****

The report describes a limited application that performs ïlive Twitter sentiment analysis. It can analyze the sentiment of tweets, .. however quite limited performance ..

INTRODUCTION

For sentiment analyzing of tweets with a given hashtag, several issues arise. First of all there is the data mining part; a great amount of tweets are needed. This will have to be queried from Twitter, and therefore use the Twitter API. The free Twitter API has certain limits, that makes this cumbersome.

Secondly there is the sentiment analyzing part.

How this is solved...

```
from sentimentanalyzer import SentimentAnalyzer
from tweetfetcher import TweetFetcher

class TSA(object):
    """docstring for TwitterSentimentAnalyzer"""

    sa = SentimentAnalyzer()
    tweet_fetcher = TweetFetcher()

    output_modes = ["hours", "days", "weeks"]
    output_mode = output_modes[0]

    analyzed_tweets = None
    output_bins = None

    def __init__(self):
        super(TSA, self).__init__()
        self.sa.load_classifier()
        # self.tweet_fetcher = TweetFetcher()

    def set_output_mode(self, mode="hours"):
        if (mode == "hours") | (mode == "days") | (mode == "weeks"):
            self.output_mode = mode
        else:
            try:
                self.output_mode = self.output_modes[mode]
            except Exception, e:
                DLOG("Output mode not set correctly: " + str(e))
                self.output_mode = "hours"
```

SUBSECTION 1

Sub title

Conclusion