10/7/2021

# Assignment 6: Natural Language

## The instructions—about this Word file template, the evaluation matrix, no more than 4 pages, and appendices—are as before. We recommend that you start with a rough draft and refine it as you become familiar with the ways in which a NL API can help.

## Purpose

In a one or two sentences, explain the overall purpose of an application (we’ll call it Natlang) that accepts a sentence of natural language input, and outputs one or two sentences in response. Natlang should account for sentiment in the input. Minimize the amount of non-NL work performed by Natlang. It is acceptable if the potential Natlag *outputs* are hard-coded, not created on-the-fly. Part 2 makes this part specific.

Natlang will process sentences mentioning a given company in an effort to complete brand name monitoring. It will provide a running score which will provide users (investors) with the analysis of whether to buy, sell, or ignore/hold a company’s stock.

## Specifications

In one sentence each as below, specify the requirements (different specific capabilities) for Natlang. Natlang should account for sentiment in the input.

**2.1 Requirement 1**: Determine if a stock is a Buy/Sell.

Natlang shall calculate, given a company’s current sentiment, if their stock should be bought, sold, or held.

**2.2 Requirement 2**: Relay Sentiment to Investors.

Natlang shall forward all sentiment analysis information to the users in the form of a buy/sell/neutral rating.

**2.3 Requirement 3**: Keep a Running Sentiment.

Natlang must calculate its ratings daily and notify users if a company in their watchlist enters either a buy or sell zone.

**2.4 Requirement 3**: Maintain an Investor’s Watchlist.

Natlang shall keep track of its users’ stock watchlist.

Continue if Natlang has more requirements.

## I/O Instances

Implement Natlang and show 3-6 input/output examples from your execution, as unrelated as possible. Indicate which requirement(s) (1.2, 1.2, or 1.3) are satisfied by each. State what NL API you used, hyperlinked (e.g., [TextRazr](https://www.textrazor.com/)) as below.

3.0 **Natural Language API Used**

I used [SocialSentiment](https://socialsentiment.io/) API.

3.1 **I/O Example 1**:

**Tests of Requirement(s)** 1

**Input and Natlang output (screenshot from execution):**

User checks a stocks rating.

Text

Description automatically generated

3.2 **I/O Example 2**:

**Tests of Requirement(s)** 1,3.

**Input and Natlang output (screenshot from execution):**

When a new stock is added, create a new rating and add it to the list.

Text

Description automatically generated

3.2 **I/O Example 3**:

**Tests of Requirement(s)** 1, 3, 4.

**Input and Natlang output (screenshot from execution):**

When a company goes to a buy or sell, notify any users that have that in their watchlist.

Text

Description automatically generated

Continue with up to 3 more examples in the above form.

## Sentiment Analysis Code

Show and explain the (commented) code pertaining (only) to sentiment analysis.

The API I was using actually handles scraping all of Twitter for specific stock tickers and does a total sentiment analysis grade for a given stock. I convert the score given to a specific ticker/company and convert that into a buy or sell or hold rating (buy if above 25, sell if below -25, hold otherwise). The numerical score is fed into a stock object daily when it queries the API.

Text

Description automatically generated

## Key Code

In a page or less, show and explain key (commented) code for the application other than sentiment analysis.

The most important code to my project were the two classes I implemented: Investor and Stock. These classes stored most the information pertaining to users and the actual stock objects.

## Source Code

Include, or point to the complete source code.

Your responses replace this.

# Evaluation

