MOLA Lab Challenge

The MOLA lab is looking for motivated MS students to hire them for a project related to software engineering and natural language processing.

UPDATES:

- The submissions will be checked **in the order of their timestamp** and on a **rolling basis**. The final submission deadline is **Friday, Jan 13th, 11:59 PM PST**.
- This position is open to **all MS students**, regardless of whether they are international students or not.

Required Skills

- 1+ years of experience with Python and JavaScript (ES5 features, promises)
- 1+ years of experience with web development and familiarity with technologies and concepts like HTML5, CSS3, DOM, JSON, REST API, CORS, Node.js, npm, Web Storage API
- Experience with NumPy, Pandas, Matplotlib libraries of python
- Experience with MongoDB (querying and indexing data)
- Experience and familiarity with Git (commit, merge, pull)

Bonus Skills

Having experience with or passing the following courses is a plus:

- Experience with Deep Learning and training/fine-tuning language models
- CSCI 571 Web Technologies
- CSCI 544 Applied Natural Language Processing / CSCI 662 Advanced Natural Language Processing
- Experience with TypeScript

Task Description

In this challenge, you are required to make a Chrome Extension that is supposed to work **only** on the **Twitter** website. By clicking the extension icon, the extension will read the tweets presented in the timeline. After detecting **English** tweets, it will calculate the sentiment score of these tweets, and based on the score, it will show three emojis (😊, 😐, ☹️) **next to the date of the tweet**.

Part 1:

In the first part, you will make a web server with two REST API endpoints. Your REST API endpoints should use the exact **same method (POST) and URL** given below. Please deploy your web server to AWS, Google Cloud Platform, or Microsoft Azure.
Note: Make sure that you are using the **free tier** of these cloud computing platforms.

`POST /api/language-detection`

Body format is a JSON array with a number of JSON objects, each with the key “tweet_text” and the actual text of the tweet.

```
[
 {
   "tweet_text": "Stats on Twitter World Cup"
 },
 {
   "tweet_text": "As the saying goes, be careful what you wish, as you might get it"
 },
 {
   "tweet_text": "شب یلدا مبارک! ❤️"
 }
]
```

This API endpoint is supposed to return a boolean which determines if the tweet is English or not. You can use any off-the-shelf language detection NLP model, but you must integrate the language detection model with your web server. The output format can be like this:

```
[
 {
   "tweet_text": "Stats on Twitter World Cup",
   "is_english": true
 },
 {
   "tweet_text": "As the saying goes, be careful what you wish, as you might get it",
   "is_english": true
 },
 {
   "tweet_text": "شب یلدا مبارک! ❤️",
   "is_english": false
 }
]
```

Note: It is okay if your endpoint cannot detect other languages like Farsi in the example above, but it has to detect the English tweets correctly.

`POST /api/sentiment-score`

Body format is a JSON array having **only the English tweets** detected in the last step:

```
[
 {
   "tweet_text": "Stats on Twitter World Cup"
 },
 {
   "tweet_text": "As the saying goes, be careful what you wish, as you might get it"
 }
]
```

This API endpoint is supposed to return the sentiment score related to each tweet, and also the final detected mood of the tweet. The output format can be like this:

```
[
 {
   "tweet_text": "Stats on Twitter World Cup",
   "sentiment_score": {
     "positive": 0.07268287241458893,
     "neutral": 0.863078773021698,
     "negative": 0.0642382949590683
   },
   "detected_mood": "NEUTRAL"
 },
 {
   "tweet_text": "As the saying goes, be careful what you wish, as you might get it",
   "sentiment_score": {
     "positive": 0.05127052217721939,
     "neutral": 0.7015827894210815,
     "negative": 0.24714668095111847
   },
   "detected_mood": "NEUTRAL"
 }
]
```

Part 2:

In this part, you should create a Chrome Extension with **Manifest V3**. You can use the official documentation of the Chrome Developers website here.
Your extension is supposed to work **only** on the **Twitter** website. When on the Twitter website, the extension will be enabled by clicking the Chrome extension icon, and:

1. It will gather all tweet texts visible in the Twitter tab.

Hint: Take a look at the DOM on the Twitter website and define a pattern to extract the tweet texts.

2. Sends all tweet texts to the language detection API endpoint.
3. After getting the response from the first API, it will **filter** the **English tweets** and sends them to the sentiment score API endpoint.
4. After getting the sentiment scores from the second endpoint, you must **add one of the** 😊, 😐, ☹️ emojis **next to the date of the tweet**. (See the below screenshot)
   Hint: You should manipulate the DOM and add the emoji as a new element with the **same style** as the tweet's date.

As you can see next to the date of the tweet, there is a separator point (.) and the sentence saying “Detected Mood: 😐”

Evaluation

The following points will be used to evaluate your submission:

- After pulling your GitHub repository, your extension can be loaded into chrome without any problems
- Your API endpoints are in AWS, GCP, or Azure, and they work using a simple postman request
- Your first API endpoint performance
- Your second API endpoint performance
- Chrome extension performance on Twitter
- Proper DOM modification on Twitter
- The style (CSS properties) of the “Detected Mood: ” should be the same as the tweet’s date
- Your GitHub repository commits have self-explanatory commit messages and proper timing of commits

Submission

Please consider the following before starting the project:

- Make sure you are pushing every file of the challenge (from the endpoints code to your extension code) to your GitHub repository
- Make sure your GitHub repository is **private**
- Make sure you use Git properly and use best practices (good commit messages, commit changes in your project over time, use git workflows)

- **Add hejabi@usc.edu as a collaborator to your private GitHub repository so I can access your repository.**

You should submit your information in the Google Form at this link. Make sure you upload the following:

- Your up-to-date resume
- The link to your first API endpoint
- The link to your second API endpoint
- The link to your **private GitHub** repository

If you have any questions, please contact Parsa Hejabi.

Good Luck!