Software Test Description

Group No. 28 : Prevention of Cyber Troll & Sarcasm System on Social Networking using Machine Learning with Bilingual Analytics

Tejas Karia, Priya Mane, Jeet Mehta, Pratik Merchant

October 2020

Contents

1 Introduction

1.1 System Overview

The virtual social media managing platform will help the user to analyse the comments received on the posts created by the user. The current system used for monitoring social media platforms is manually analysing comments and deleting negative comments or manually blocking a regular spammer or hate promotor. This web application will eliminate the effort required to manually screen thousands of comments and block hundreds of spam users. The user has to register to the application and then provide permission to the web application for accessing his account by authenticating via the OAuth platform. The application then retrieves the comments on the posts. The suitable algorithms for sentiment analysis and sarcasm are applied on the comments and displays the aggregated results on the panel home. The user can search for specific posts and view analytics for those posts. The platform can also provide auto-replies for positive comments and reports/blocks negative comments. The application also blocks spam users. Hence, this application will highly contribute towards reducing online hate and help influencers to manage their social media profiles.

1.2 Test Approach

1.2.1 Testing Method

Blackbox testing will be used to perform testing. Black Box is a testing method in which internal structure, design, implementation of the item being tested is not known to the tester. This method basically tests the functional requirements of the project. White box testing will be performed to test internal structure, design and coding of software to verify flow of input-output and to improve design, usability and security.

1.2.2 Testing Strategies

The testing strategies used for our project are alpha testing. Alpha testing is performed by testers who are usually internal people of the team. As usually alpha testing involves both white-box and black-box testing, we will consider black-box testing as it is well suited for our project. In our project this testing will be done by our testers.

2 Test Plan

2.1 Features To Be Tested

Software modules that need to be tested are:

1. User Login: TC-1

2. Authorization of the web application: TC-2
3. User Dashboard: TC-3
4. Blocking/Reporting of malicious accounts: TC-4

2.2 Features Not To Be Tested

Automatic reply to comments: The testing of this feature is out of the scope of this project because it is a part of the services provided by the individual API of each social media platform.

2.3 Testing Tools and Environment

Manual approach as well as computerized approach would be used for testing of the application. The web application would be extensively used during the testing of this application. An active internet connection might be required during the testing of certain features. Microsoft Visual Code, CLI (Command line interface) and certain online open source testing tools will be used to perform the testing.

3 Test Cases

3.1 User Login: TC-1

3.1.1 Purpose

To authenticate any user.

3.1.2 Inputs

1. Username
2. Password

3.1.3 Expected output and Pass/Fail Criteria

It depends on the details entered by the user. After validating the inputs given by the user, then the user will be granted access to the application.

3.1.4 Test Procedure

If an account is not created then the user will have to create an account and then login.
Constraint: Username and password used at the time of Sign Up should match.

3.2 Authorization of the web application: TC-2

3.2.1 Purpose

Retrieval of comments and perform other actions using the respective user ID of the user.

3.2.2 Inputs

API key provided to the developer.

3.2.3 Expected output and Pass/Fail Criteria

A key will be generated to the developer. A failure will occur if and only if the application decides to dismiss the developer's request to generate an API key.

3.2.4 Test Procedure

This module needs to be tested manually.
Constraint: Developer should have created an account for this.

3.3 User Dashboard: TC-3

3.3.1 Purpose

To test the application for is its usability and accessibility and implement the changes suggested by the tool

3.3.2 Inputs

URL of the web application

3.3.3 Expected output and Pass/Fail Criteria

The output would be in the form of a report that would grade the website on various parameters. The passing criteria would be to get a decent score on all the parameter grading.

3.3.4 Test Procedure

Use an open source tool for testing the overall usability of the web application.
Constraint: The web application should be able to access the user's data.

3.4 Blocking/Reporting of malicious accounts: TC-4

3.4.1 Purpose

The sentiment value generated by the algorithm will help us in identifying whether an account is abiding by the rules and guidelines of the application.

If the values indicate that an account is posting vindictive comments then the account will be blocked.

3.4.2 Inputs

User ID of the person posting hate comments.

3.4.3 Expected output and Pass/Fail Criteria

The output obtained will be blocking of the account who is spreading hate on the website. If the blocking is done correctly then it's a success.

3.4.4 Test Procedure

This will be tested manually with the help of the backend team.
Constraint: Only accounts violating the policies and guidelines of the social media platforms must be blocked.