PROJECT ANALYSIS

Week 3 Assignment

ABSTRACT

This document was created for UMUC Course, CMSC 495, and analyzes aspects of the (TNC)

Group 3 Members
Name: Christiano, Andrew
Name: Fernandez, Yrume
Name: Orwick, Brian
Name: Sell, Julia
Class: CMSC 495 - Current Trends and Projects in Computer Science
Professor: Dr. Hung Dao
Due: 10 September 2018

Version Control

Revision #	Date	Name	Descriptions	Contact Info
TNC_0001	9/4/2018	Brian Orwick	Created	Orwick12@outlook.com
TNC_0002	9/5/2018	Yrume Fernandez	Revisions	Yrume.fernandez@gmail.com
TNC_0003	9/6/2018	Andrew Christiano	Revisions	ajchristiano91@gmail.com
TNC_0004	9/6/2018	Julia Sell	Revisions	selljm14@gmail.com
TNC_0005	9/7/2018	Andrew Christiano, Brian Orwick, Julia Sell, Yrume Fernandez	Finalized	ajchristiano91@gmail.com orwick12@outlook.com selljm14@gmail.com yrume.fernandez@gmail.com

Table of Contents

1. Outside Systems

The Trusted News Code (TNC) software connects to various news websites using various python libraries. This software requires internet connection. The following is a list of trusted sources that are hardcoded for TNC to check, and a level 0 Data Flow Diagram (DFD):

- https://www.foxnews.com
- https://www.usatoday.com
- https://www.cnn.com
- https://www.bbc.com
- http://www.apnews.com

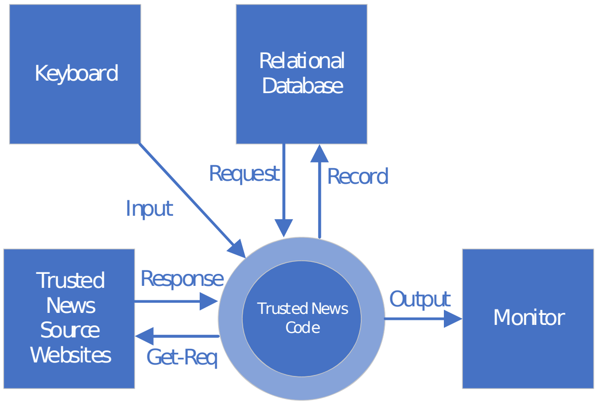

Figure 1 - Architectural Context Diagram (ACD) External Interfaces

2. Input Data

TNC does not require any input data from a user. A user will navigate to the TNC Server Web Page and receive statistical information on articles posted within the list of trusted news sources. TNC takes input data from web pages that are part of a hardcoded list of trusted news websites. This data is then parsed and structured for storage and retrieval using the SQLite Data Base that is part of the system. The retrieval of the data from the news websites will occur asynchronously in separate threads.

3. Output Data

The data that has been aggregated by the TNC will presented from the SQLite Data Base (DB). This database will house all the required information to statistically analyze the trustworthiness of posted articles. Trustworthiness, as presented by the TNC, is the verifiability of news story across multiple news sources. A trustworthy story is prevalent across organizations and is more likely to be published multiple times by many different people. The following provides information on the proposed structure of the database table used during execution of the TNC:

NEWS ARTICLE (TABLE NAME)			
ID (TEXT)	NUMBER (INT)	NAME (TEXT)	COUNT (INT)
Web-Page URL	Article Number	Article Name	Article Count

Figure 2 - Architectural Context Diagram (ACD) External Interfaces

4. Data Processing

Information captured and stored within the TNC is used to identify the trustworthiness of an article. Initially the TNC software, using the newspaper library, connects and downloads a copy of newspaper articles housed within the source websites. This information is captured and stored for subsequent processing within the SQLite DB. After each site, within the list of trusted sites, is captured and stored the TNC software parses and iterates through the contents of each articles sequentially comparing each word within articles. As the system compares each article a counter is incremented when common words are found, and a percentage of common words is calculated. If two articles have over 70%-word commonality (> 70), then both articles are assumed related. The system adds a count to each article identified as related, and articles acquiring a higher count of related articles are assumed more trustworthy than lesser related articles. All data is stored in database for storage and retrieval.

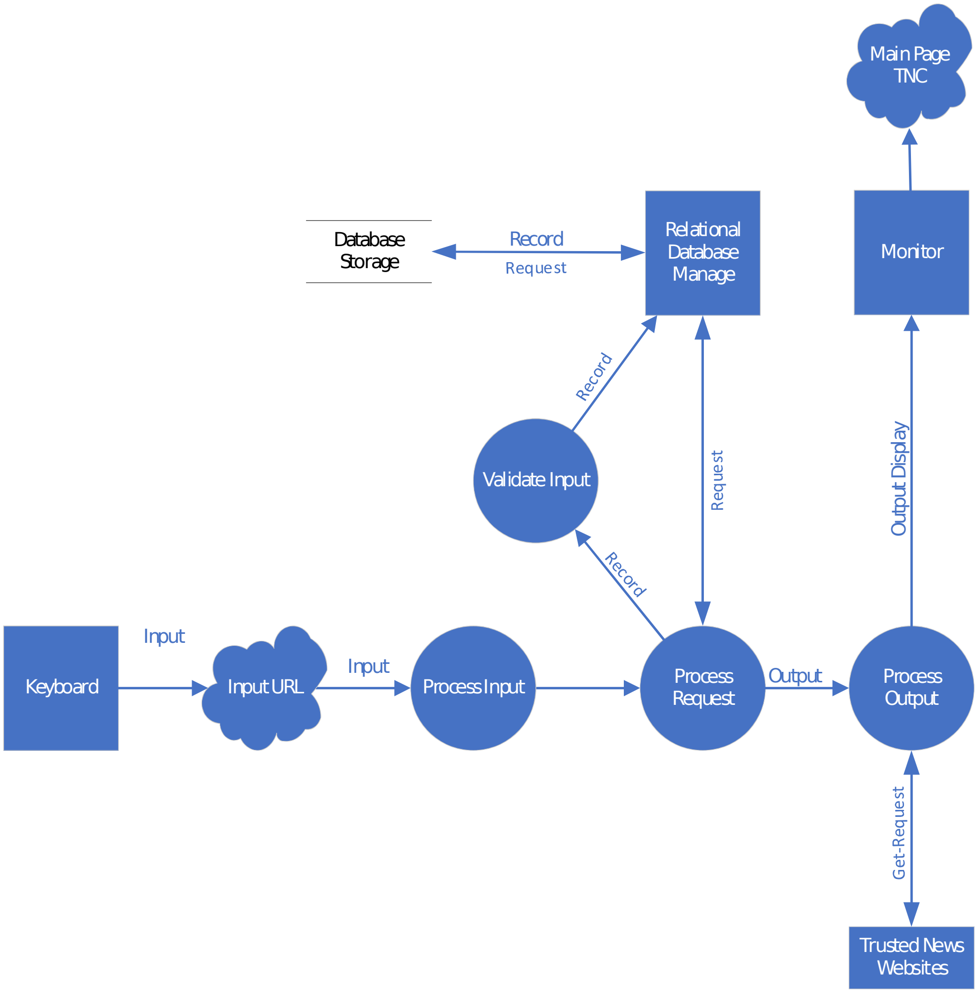

Figure 3 - Data Flow Diagram (DFD) Interfaces

5. Subsystem Requirements

The TNC Software requires connections to local database (SQLite DB) which provides a structure to maintain information for analysis in a persistent manner. This DB also provides the capability of running analysis on articles while the system is offline. It also requires access to local systems networking services to retrieve the required information from external websites while online.

6. Data Interfaces

The proposed Data Interfaces are as follows:

1. Web
    a. routes()
    b. queryDB()
2. Scraper
    a. get_articles()
    b. parse()
    c. download()
    d. generate_news()
3. DB
    a. create_table()
    b. db_insert()
    c. db_update()
    d. db_query()
    e. mass_update()

Below is a flow diagram that shows how the different classes will interact:

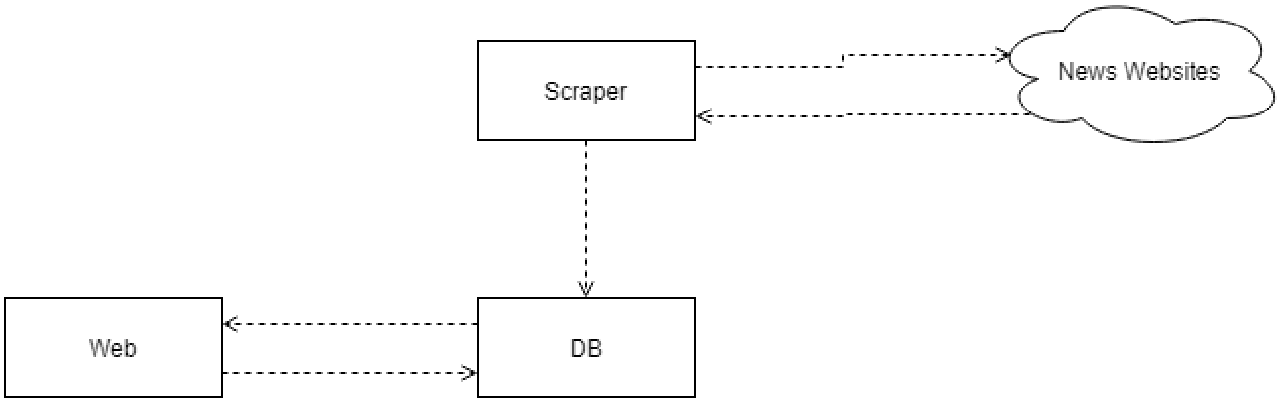

Figure 4 - Class Interaction Diagram

7. Potential Risk and Mitigation

Potential risks when operating the system are:

1. The system may fail to retrieve information from trusted websites (due to https certificate error, no internet connection, or other unknown circumstances)
    a. Mitigation: System will check for error code 200 to ensure website is available prior to submitting get-request for html.
2. System may fail to provide adequate information if fake-news goes viral and a number of articles are found on the internet based on false information.
    a. Mitigation: System will have a fake-news option that will lower the trustworthiness of the article.

8. Future Enhancements

1. Sentiment analysis:
    a. Attempt to show if there is any correlation between rate of occurrence of certain words and their impact on the trustworthiness of an article
    b. Create visualizations to illustrate vocabulary, determine the relationships between word choice and trustworthiness.
2. Search functionality:
    a. Provide a way for users to search through all content to discover the trustworthiness of a specific article, or specifically tagged articles.
3. User related:
    a. Provide a way for users to have a dialogue about articles
    b. Provide a way for users to propose additional articles for analysis
    c. Provide a way for users to propose additional websites for automated analysis
    d. Present users with their history, to show them their implicit biases
4. Advanced Tracking:
    a. Present metadata regarding publishing dates to expose when stories are posted on different websites
    b. Track if the same exact story is published by a different author(s) (plagiarism)
    c. Track if the same exact story is published by the same author(s)
    d. Track the trustworthiness of individual author(s)