LITMUS-Reddit Final Report
CS6365 Spring 2021

Eric Hsieh
Georgia Institute of Technology
hsieh.eric@gatech.edu

Dongsuk Lim
Georgia Institute of Technology
dlim46@gatech.edu

1 Introduction

Our project for Spring 2021 of CS 6365 is LITMUS-Reddit. We modeled our project's general pipeline after the original LITMUS project [1]. This semester, we proposed and completed a project on discerning disinformation with Reddit on topics regarding the COVID-19 vaccine. We created a dataset by compiling Reddit post titles that were related to the COVID-19 vaccine, and developed a machine learning model that could separate concrete and accurate facts from disinformation from this data. We then integrated our model and data with a front-end to demonstrate our model. While we may not be able to trace the source of it all, our model was successful in identifying

Additionally, we hope that our work can contribute to LITMUS/EDNA by providing both a new potential source of data as well as a working model that can distinguish misinformation in this new data source.

2 Motivation

The project was motivated from polarized ongoing public debate on topics related to COVID-19 vaccine. COVID-19 is the first pandemic that has occurred where the large scale of modern media has been used to provide helpful information [2], but there has been significant amounts of misinformation spread online.

We chose Reddit as our main study because it is a supposed to be a decentralized, user-run media forum where anyone can post and vote. We believe that disinformation runs rampant on Reddit due to the lack of fact-checking and moderation, combined with the anonymity of its users. In the wake of the COVID-19 pandemic, there have been many incidents where users have spread strong disinformation and smear campaigns against the COVID-19 vaccine. We want to identify disinformation on Reddit by contrasting with given facts from trustworthy news sources, such as the Centers for Disease Control and Prevention (CDC) and the World Health Organization (WHO).

3 Architecture and Frameworks

The envisioned system architecture is comprised of several frameworks and components. The main focus of this project is the classification of disinformation, and all components exist to create a suitable pipeline to ensure the classification runs smoothly.

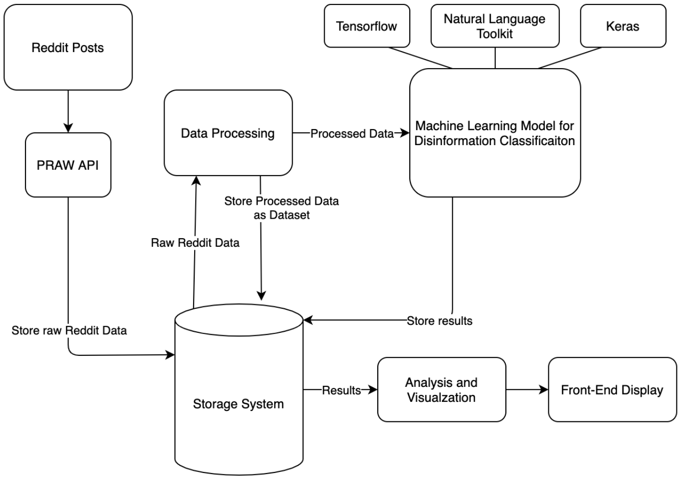


Figure 1: Proposed System Architecture

For this project, we used Python. Python supports the PSAW API to pull Reddit information, as well as machine learning and NLP frameworks like TensorFlow and Natural Language Toolkit. We first pull data from Reddit through API before cleaning the raw data via data prepossessing. With the processed data, we feed it into our models. We then save the best model and integrate it with a frontend for application usage.

4 Implementation

This section will cover detailed implementation of the LITMUS-Reddit project. The source code can be found at this Github link: https://github.com/hsieheric/LITMUS-Reddit

4.1 Source Data

Our Reddit post titles were pulled from four main subreddits, in descending order of size:

- https://www.reddit.com/r/Coronavirus/
- https://www.reddit.com/r/COVID19/
- https://www.reddit.com/r/CoronavirusNewYork/
- https://www.reddit.com/r/CoronavirusFOS/

These four subreddits contain a significant amount of posts pertaining to the COVID-19 vaccine. They contained a mix of general discussion, science-based discussion, location-based discussion, and controversial discussions, respectively. We hoped that this could create a broad domain of discussion for our classification.

4.1.1 Data Gathering and Preparation

For data gathering, we start by using the PSAW Reddit API, which is based off of PRAW. This API allows us to retrieve comments and posts from selected SubReddits, using filters to narrow our data pulling. After these posts are identified, we look for numbers and facts that match up with the correct information from WHO and CDC sources and label the data as misinformation or not misinformation. This labeled data is used as training data for our supervised learning models.

Our data gathering pipeline was inspired by the one used for LITMUS; we followed a very similar structure from gathering our data to preprocessing it before feeding it in to our model. The following is a juxtaposition of the pipeline used in LITMUS and our LITMUS inspired social filtering pipeline:

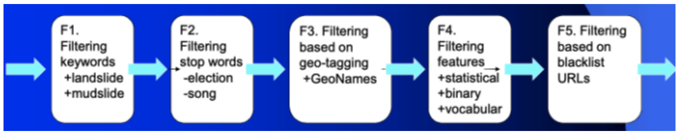

F1. Filtering keywords +landslide +mudslide
F2. Filtering stop words -election -song
F3. Filtering based on geo-tagging +GeoNames
F4. Filtering features +statistical +binary +vocabular
F5. Filtering based on blacklist URLs

Figure 2: LITMUS Pipeline

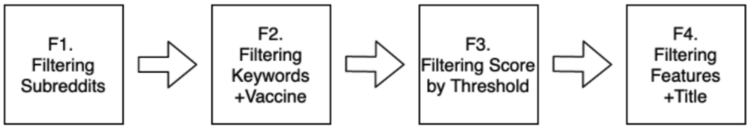

F1. Filtering Subreddits
F2. Filtering Keywords +Vaccine
F3. Filtering Score by Threshold
F4. Filtering Features +Title

Figure 3: Our LITMUS-inspired Pipeline

We first filter by SubReddit, then by keywords, and finally by score, which ensures that we get relevant data that has adequate visibility. Next, we proceeded to label the data by hand. We had initially spent some time reading about the COVID-19 vaccine from CDC and WHO sources, and used this information to try and impartially label the post title data as misinformation and not misinformation. One problem here is that our labeling is directly correlated to our personal bias; there is a good chance that not all of our data is labelled correctly, or if there even is a 'correct' label for many of these points. Regardless, we tried our best to create impartial data.

Compared to LITMUS, our data is much broader since it just covers COVID-19 vaccines instead of landslides. Afterwards, in our data preprocessing step, we filter out irrelevant features to just retain the title for encoding and the labels for classification.

4.1.2 Data Analysis

An important topic that was discussed in class was True Novelty, which was also very prominent in our project. True Novelty refers to the novelty of our data, or the data beyond the facts; we have known data that we have gathered, but we also have other domains like disinformation, known unknowns, and unknown unknowns.

Our project focuses mostly on this concept of anti-information, and finding out ways to discern this from facts. As humans, we label our data with the intention of categorizing fact from misinformation. Here, we are prone to our own biases regarding the misinformation, but it is still known. The spreaders of misinformation will try and hide fake news inside known facts, utilizing techniques like half-truths, white lies, or out-of-context quotes. By building up a dataset that has these types of misinformation labeled correctly, we hope that our model will be able to learn the nuances and false information within them.

However, what we have to be careful of during our labeling are known unknowns. Known unknowns are where we know that the data is unknown, meaning that it is extremely easy to misunderstand or agree with false information, since there is no concrete facts about it. Additionally, those that spread misinformation also know the data is unknown, so they will use that to their advantage. Good examples of this that we found in our data would be the impact of the vaccine on mortality; there is a lot of misinformation spread of people who got the vaccine and died soon after. However, most of these cases are people that did not die of COVID-19, or who took the vaccine as a last-minute attempt to save them from COVID-19, but were unsuccessful.

We also visualized our misinformation and not misinfomration datasets in word clouds:

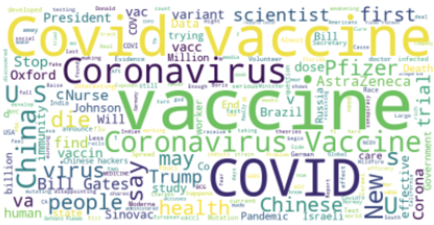


Figure 4: Misinformation Word Cloud

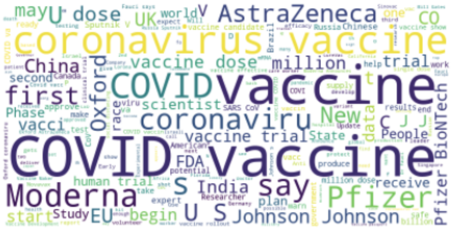


Figure 5: Not Misinformation Word Cloud

The ‘Misinformation’ Word Cloud focuses on fewer terms than the ‘Not-Misinformation’ Cloud, showing that misinformation tends to focus on fewer topics that mostly likely had a greater effect. The words that we noticed were more prominent in the Misinformation cloud were ‘Bill Gates’, ‘China’, ‘Chinese’, ‘Trump’. While the same words were shown in the Not Misinformation cloud, this meant that non-misinformation titles did not focus so heavily on these topics.

From the misinformation subsection of our data, we noticed that a lot of them were composed of out-of-context quotes, misquotes, half-truths, and white lies. For example, if the post title is ‘Bill Gates says 700,000 people may die from using the Covid 19 vaccine’, in reality, Bill Gates said that the side effects of the vaccine may impact up to 700,000 people. However, the intention behind the ‘misquote’ is clear, as this title aims to spark outrage and doubt about the vaccine.

If anything, our data analysis proves that there is sizable amounts of misinformation being spread through Reddit. Therefore, our project should produce meaningful results that can be applied to LITMUS/EDNA.

4.1.3 Data Processing

In order to prepare the data for classification, we have to perform data processing on the text. We use NLP techniques here like normalizaiton, tokenization, and word embeddings in order to make the text readable by the machine.

For all of our models, we used used vectorization after normalization, using SKLearn’s CountVectorizer or Gensim’s Word2Vec. This converts the titles into tokens and vectors that can be used as inputs to the machine learning algorithms. We also used word embeddings, which are very similar, if not the same concept as word vectors, but from the Keras Tokenizer library instead. One important note was the length of the word embeddings. Machine learning models expect a static input size, so each input must be preprocessed to be a specific length. We chose embeddings of length 50, as post titles should not be ridiculously long. This means that all inputs must be encoded to a length of 50 max, or padded until such.

Word embeddings are especially effective with NLP classification tasks. Word embedding algorithms learn word associations from the input text and encodes it in a vector-based numerical format, taking into account features such as word location and context, among others. As seen in our Word2Vec clustering in section 4.2.1, the distance between two words demonstrates how similar they are in similarity scores. For our machine learning model, using word embeddings like Word2Vec means that we be certain the model will be able to learn the importance of these words based on their calculated scores.

4.2 Model and Classification

In this section, we present four main models that we developed to classify disinformation from Reddit post titles. We used SKLearn, TensorFlow and Keras for quick prototyping, as well as Natural Language Toolkit various NLP techniques to preprocess and augment our data for classification. The NLP techniques include vectorization, tokenizing, padding and word embeddings in order to transform our textual data. We divide the entire dataset to training and testing datasets with the ratio of 9 to 1 respectively. The results are the following:

Testing Accuracy of Models on Our Classification Task				
Model	Gaussian Naive Bayes	Logistic Regression	KNN	**Word Embeddings + Neural Network**
Accuracy	73.06%	85.38%	83.46%	**89.40%**

The neural network with word embeddings was able to achieve close to 90% accuracy. As opposed to our baseline models of Gaussian Naive Bayes, Logistic Regression, and K-Nearest Neighbors, the neural network was able to learn more from the dataset to make correct classifications. The neural network was composed of four layers: an embedding layer of to size 50, followed by a flatten layer, a dense layer of size 64 with ReLU, and a classification dense layer of size 1 with softmax.

We also tried variants of the neural network, such as using LSTMs instead of Dense layers, as well as using different tokenizers and word embeddings. However, the combination of word embeddings of size 50 with a single-layer dense neural network achieved the best results. We theorize that this is not necessarily our neural network, but may be attributed to the Keras Tokenizer word embeddings versus the SKLearn word vectors. Our data also differed in its preprocessing for the baseline models versus our word embedding neural network.

We also think that the reason LSTMs did not perform as well is due to our training data; we have a lot more not misinformation classes than misinformation, so the data itself is skewed. While our model is not guaranteed to be the best classifier, our preprocessing is still very effective.

4.2.1 Classification Challenges: Borderline Data

Unsupervised learning algorithms such as Word2Vec or K-Nearest Neighbors are good for classification problems that can be solved by grouping similar data points together. However, misinformation is designed to bypass unsupervised learning through what we call borderline data. Borderline data rides the line between misinformation and not misinformation and aims to be classified closer to not misinformation.

This is reflected in our work with unsupervised learning on our dataset. We ran Gensim Word2Vec (word2vec-google-news-300) on our data, which uses a pretrained model to perform clustering. We visualized this clustering in the following images:

In our Word2Vec visualization, there are two distinct clusters. The images zoom in on the smaller cluster on the left to show the text. These clusters try and map each word vector to similar points. However, these two groups are not indicative of misinformation or not misinformation, since the smaller one is more for words regarding COVID-19, while the larger cluster is for everything

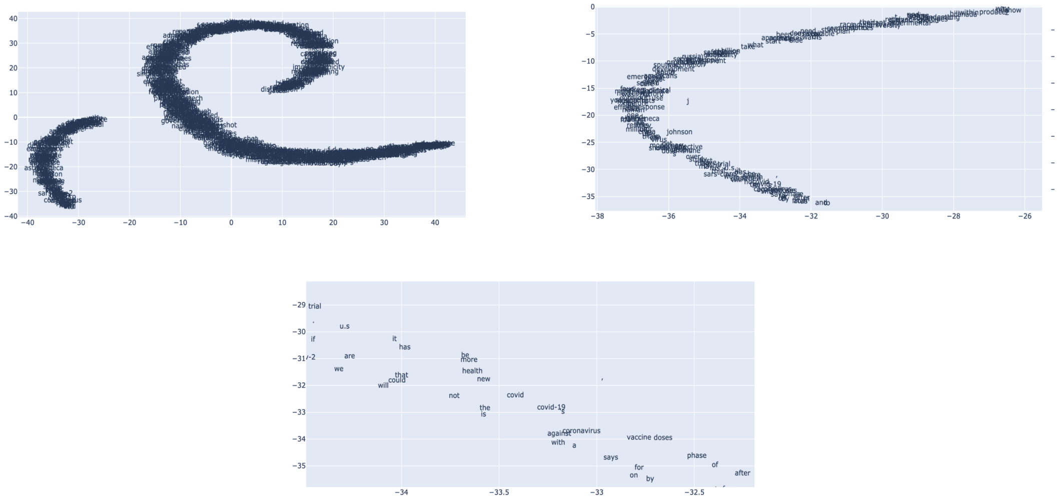


Figure 6: Word2Vec Clusters

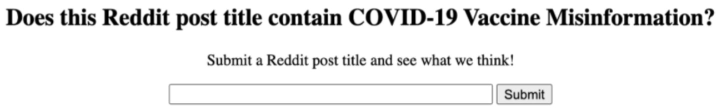
else. Thus, this unsupervised learning from Word2Vec was not able to create a distinction with misinformation and not misinformation.

If we wanted to get a better division of our titles, we may try training Word2Vec on our general corpus instead of using the pretrained model by Google News. The issue here is that we do not have enough total data for unsupervised learning to make a larger distinction; if we could bring our distribution of classes to 50/50, then the Word2Vec model may have a better chance at using unsupervised learning to create clusters of correct classes.

Additionally, we did not try traditional unsupervised learning techniques like K-Means Clustering on our data. However, we theorize that it would perform poorly compared to our supervised learning with labeled data. This is because of the inclusion of borderline data; unless we created a third class for borderline data and labeled it manually, the model would probably be unable to make the distinction that it was misinformation, since it would look and measure very close to not misinformation in the word vector space.

4.3 Front-End Application

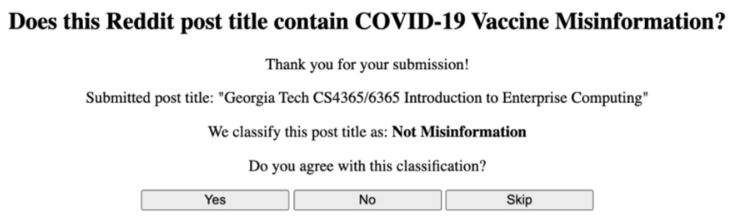
In this section, we discuss our application. The purpose of our frontend was to both test our classification model and potentially gather more data from users.The following is two screenshots of our application: one of the home page and the other after a successful classification has been run.



Does this Reddit post title contain COVID-19 Vaccine Misinformation?

Submit a Reddit post title and see what we think!

Figure 7: Homepage



Does this Reddit post title contain COVID-19 Vaccine Misinformation?

Thank you for your submission!

Submitted post title: "Georgia Tech CS4365/6365 Introduction to Enterprise Computing"

We classify this post title as: **Not Misinformation**

Do you agree with this classification?

Figure 8: Results Page

The frontend was implemented using the Python Flask framework for our backend and setting up python subprocess calls to run the best-performing model. We can then take a user input and perform the appropriate transformations of tokenizing and encoding to match it up with our frozen model before performing classification on the processed input data and displaying the results.

The user input must be transformed in exactly the same way as the training data before being fed into the model. Therefore, since we settled on using the Word Embeddings and Neural Network model, we must replicate that same embedding process for our input data. This involves tokenizing on the same corpus as our input training data.

Because we are running a frozen model, we do not need to train the model each time the server is run or an input is given. Without the frozen model, the web application would not be feasible as it would have to train each time. After the classification, we prompt the user for additional input on whether or not this classification is accurate, as seen in the results page in Figure 8. We then save the input title, our model's classification, and the user's opinion on our classification as inputs to a CSV. We can use this additional data to adjust our model and classification algorithm, as well as add the user's post title input as more data.

5 Deliverables

Our deliverables for this project include:

1. Source code for our data gathering and analysis
2. Classification Model
3. Web Application
4. Collected dataset
5. Presentation and Demonstration of our work
6. Results and analysis in a paper

The source code, classification model, web application, and dataset can be found at this github link: https://github.com/hsieheric/LITMUS-Reddit

The presentation and demonstration of our work can be found here: youtube link

6 Related Works

Hussain et al. introduced a similar work for detecting COVID-19 misinformation on social media. The paper introduced the COVIDLIES dataset for misconception detection on Twitter. The dataset contains 6761 COVID-19 related tweets, identified and annotated by researchers from the UCI School of Medicine. Hussain et al. formulated the task of detecting misinformation as retrieving relevant misconceptions, and classifying whether the tweet supports or refutes it. The paper demonstrated that it is feasible to detect the stance of tweets towards misconceptions using models trained on existing datasets. [3]

Another NLP based COVID-19 misinformation work is presented by Serrano et al. The paper introduces a simple NLP methodology for detecting COVID-19 misinformation videos on YouTube by leveraging user comments. The transfer-learning-pre-trained models were used to generate a multi-label classifier that can catogorize misleading content. Showing significantly improved accuracy of misinformation video detection (89.4%), the paper demonstrates that leveraging large quantities of user comments is effective in the prediction of COVID-19 misinformation videos. [4]

7 Conclusion

In this project, we investigated Reddit for misinformation and create a model that would be able to classify this misinformation correctly. NLP and LITMUS techniques were applied to Reddit post titles regarding the COVID-19 vaccine to classify them into either misinformation or not misinformation classes. As an application, we developed a COVID-19 Reddit post title classification and reporting system, which is a novel approach that includes looking out for known and unknowns in our work.

Reddit has real-time characteristics and a vast user pool that distinguishes it from other social media. Therefore, utilizing such immense information brings many benefits in our world of data. It is hoped that this paper provides some insights and trajectory for future integration of NLP techniques with Reddit data, as well as providing a new source of data and methods of misinformation classification to help assist LITMUS/EDNA.

7.1 Future Work

One important step is to improve the quantity and quality of the training data. We felt that the distribution of our data was skewed too heavily towards the Not Misinformation classification. With a better balance, we may be able to get more meaningful results with more powerful models, such as swapping out our Dense Neural Network for an LSTM to take advantage of the temporal nature of text.

Additionally, the issue with hand-labelling data is that our classification works are prone to human bias and errors. One could see there would be a potential increase of our classification accuracy with having impartial labelling of the data along with more data.

In terms of expansion, we would like to expand our system to classify misinformation regarding any COVID-19 related information on Reddit, not limited to just the COVID-19 vaccine. This would require finer tuning of our model and improvements of our dataset.

Finally, we would like to to improve the UI with better frontend Javascript library such as React and Angular as well as connect our entire application to a persistent database in a cloud environment for potential traffic. This would allow us to host our work as a full web application.

References

[1] Calton Pu **andothers**. “Beyond Artificial Reality: Finding and Monitoring Live Events from Social Sensors”. **in**: *ACM Trans. Internet Technol.* 20.1 (**march** 2020). ISSN: 1533-5399. DOI: 10.1145/3374214. URL: https://doi.org/10.1145/3374214.

[2] World Health Organization. “Managing the COVID-19 Infodemic: Promoting Healthy Behaviours and Mitigating the Harm from Misinformation and Disinformation”. **in**: (2020). URL: www.who.int/news/item/23-09-2020-managing-the-covid-19-infodemic-promoting-healthy-behaviours-and-mitigating-the-harm-from-misinformation-and-disinformation.

[3] Tamanna Hossain **andothers**. “COVIDLies: Detecting COVID-19 Misinformation on Social Media”. **in**: *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*. Online: Association for Computational Linguistics, **december** 2020. DOI: 10.18653/v1/2020.nlpcovid19-2.11. URL: https://www.aclweb.org/anthology/2020.nlpcovid19-2.11.

[4] Juan Carlos Medina Serrano, Orestis Papakyriakopoulos **and** Simon Hegelich. “NLP-based Feature Extraction for the Detection of COVID-19 Misinformation Videos on YouTube”. **in**: *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*. Online: Association for Computational Linguistics, **july** 2020. URL: https://www.aclweb.org/anthology/2020.nlpcovid19-acl.17.