



CSC 215 Project 3 Report

Deadline: 3:00 PM, March 27, 2023

Contents

1. Problem Statement
2. Methodology
3. Experimental Results and Analysis
4. Task Division and Project Reflection
 - a. Task Division
 - b. Challenges
 - c. Learnings
5. Additional Features

Members: Aayush Shukla (ID: 302777098), Shubham Ghule (ID: 302784482), Rajvee Modi (ID: 219712052)

- **Problem Statement:** The problem statement is to develop a system that can verify whether a given multimedia item (image or video) related to an event matches the content described in the accompanying tweet. The primary objective is to provide a binary decision on the veracity of the multimedia item in terms of reflecting the reality of the event. The scope of the project is to focus on events with international news potential, and the system needs to handle multimedia items that are potentially manipulated or edited.
- **Methodology:** The first step is to prepare the data for training and testing. We will use the data in the "devset" folder to train our models and test the model using the data in the "testset" folder. We will remove the label "humor" as we are only focusing on "fake" and "real" tweets. We will also remove all the tweets that have no corresponding images. To encode the text, we will use pre-trained Word2Vec. This will allow us to create a numerical representation of the textual data, which can be used as input for the neural network. We load images in the same order as they are represented in the labeled set so that all the labeled tweets match their corresponding images. We will use EarlyStopping when training neural networks using TensorFlow. This will help prevent overfitting and improve the generalization performance of our models. We will train neural network models using TensorFlow. We will use pre-trained embeddings as input to the neural network and train the model on the labeled data to predict whether a tweet is real or fake. Finally, we will evaluate the trained models on the test data by calculating the Recall, Precision, and F1 score for real and fake news, respectively. This will help us measure the performance of our models and compare them to each other. We will print out these scores for both real and fake news to analyze the results. Note that we will not perform hyper-parameter tuning, as it is not required for this project. The focus is on building a functional model that can perform the binary classification task of verifying the multimedia item's reality in the way described in the tweet.
- **Experimental result and analysis:** The following table shows the result

```
... Found 1962 unique tokens.
20/20 [=====] - 16s 794ms/step
Classification Report for Real Data:
```

	precision	recall	f1-score	support
real	0.68	0.18	0.28	213
micro avg	0.68	0.18	0.28	213
macro avg	0.68	0.18	0.28	213
weighted avg	0.68	0.18	0.28	213

```
Classification Report for Fake Data:
```

	precision	recall	f1-score	support
fake	0.69	0.96	0.80	414
micro avg	0.69	0.96	0.80	414
macro avg	0.69	0.96	0.80	414
weighted avg	0.69	0.96	0.80	414

- Task division and Project Reflection

- Task Division:

Sr no.	Task	Assigned to
1.	Data preparation	Aayush, Rajvee
2.	Text encoding	Shubham
3.	Neural network architecture design	Rajvee
4.	Model training	Aayush
5.	Model evaluation	Shubham
6.	Compare models	Rajvee
7	Multi output LSTM Model	Shubham
8.	Report	Aayush, Shubham, Rajvee

- **Challenges:** Following are the challenges we faced while working on this project:

- Analysis of dataset
- Data quality issues & Data preprocessing
- Using the entire dataset caused insufficient memory error. To resolve this, we are taking 35% of the dataset.
- Performance metrics
- Concatenation of 2 models

- **Learnings:** The tweet data and image data may come in different formats and structures, and cleaning and preprocessing the data is an essential step before merging and analysis. Concatenating two models can improve the overall performance of the analysis, but it's important to carefully evaluate the models and ensure that they are compatible and complementary.

- **Additional Features:**