Memotion Analyzes

Bardia Karimi Zandi, Mohammad Mohammadi

January 2020

Abstract

Information on social media comprises of various modalities such as textual, visual and audio. NLP and Computer Vision communities often leverage only one prominent modality in isolation to study social media. However, computational processing of Internet memes needs a hybrid approach.In the last few years, the growing ubiquity of Internet memes on social media platforms such as Facebook, Instagram, and Twitter has become a topic of immense interest. Memes, one of the most typed English words (Sonnad, 2018) in recent times. Memes are often derived from our prior social and cultural experiences such as TV series or a popular cartoon character .These digital constructs are so deeply ingrained in our Internet culture that to understand the opinion of a community, we need to understand the type of memes it shares .This project is about recognizing the text of a meme and from the picture and the text , machine should choose the subject of the picture .

1 Introduction

This paper is the implementation of Memotion Analysis [8MA] task of SemEval2020 [SEM]. As we discussed in Abstract section, we define two tasks on this problem:

1.1 A. Sentiment Classification

Given an Internet meme, the first task is to classify it as positive, negative or neutral meme.

1.2 B. Humor Classification

Given an Internet meme, the system has to identify the type of humor expressed. The categories are sarcastic, humorous, and offensive meme. If a meme does not fall under any of these categories, then it is marked as a other meme. A meme can have more than one category.

2 Proposed Method

A deep neural network is trained for the tasks defined in the introduction.we elaborate the steps required for the training of this neural network in the following sub-sections:

2.1 Pre-Processing

The data was initially in the form of CSV format. The CSV file consists of: image name, image URL, OCR extracted caption, corrected OCR, humour level, sarcasm level, offense level, overall sentiment. First of all we downloaded images files from their URL field in the CSV. some of the URLs were corrupted ,more over minority of them could not be converted to jpeg format by tensorflow. The total number of images available for forming the train dataset was 5,847 and for forming the test dataset was 831.[code]

For task A we organized the images in three classes. Unfortunately there was not equal number of images provided for each category and therefore the data was not properly segmented. Hence the data was not proper for training deep neural networks.

For task B we could use all of the data for each of the categories. Each meme can belong to any one or more categories or it can belong to none of them which we mark as other memes.

2.2 Tokenize Input Texts

We used Glove [GLO] word embedding for converting texts to embedding vectors.

2.3 Builing Dataset Using Tensorflow

Inputs of the network are images with their coresponding OCR. the output of the network differs based on each task.

next step was normalizing photos , all the photos were resized to 200x200 pixels and division by 255.

2.4 The Architecture of The Neural Network

Overally we have used Tensorflow [TF] Keras [KRS] library and we made our network using Keras functional API. We have mainly used two types of layers to build this network:

- Convolutional layers for learning images.
- Long Short-Term Memory layers for learning texts.

The architecture of the networks for each task is as below:

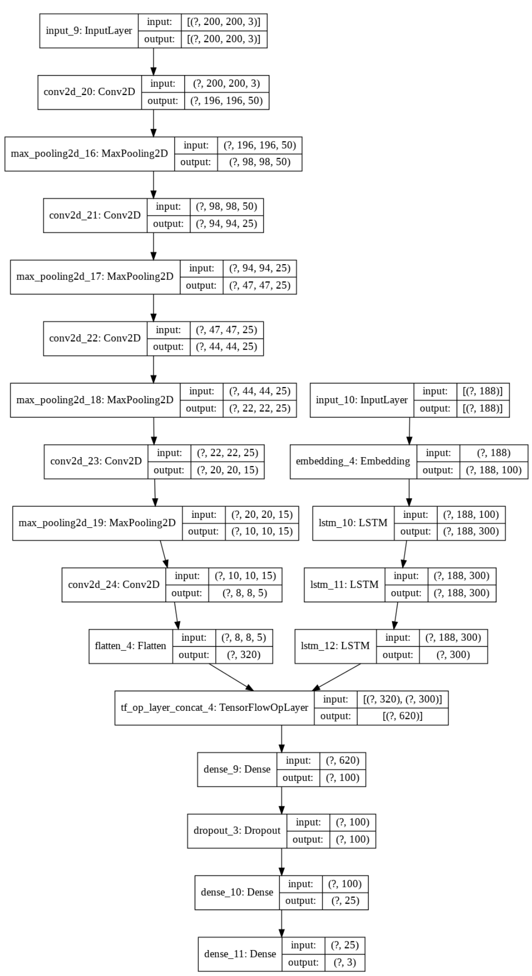

Figure: Model for task A

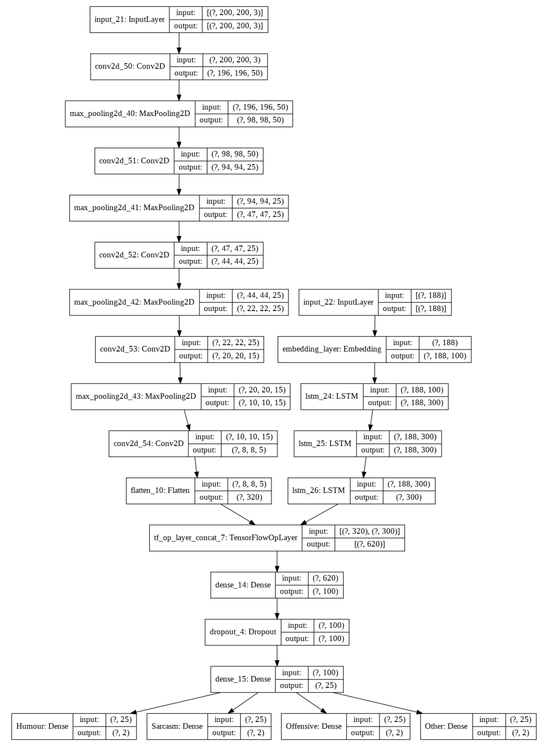

Figure: Model for task B

3 Results

Best result for task A after 40 epochs with Adam [ADM] optimizer and categorical cross entropy loss function:

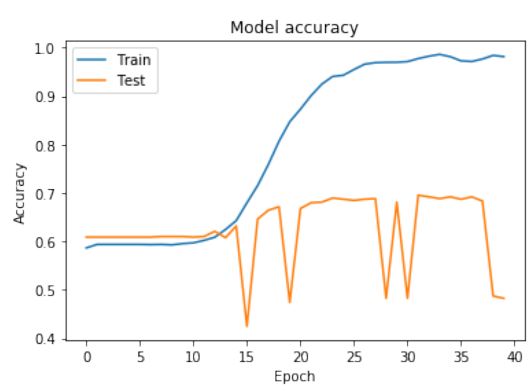

Figure: Accuracy for task A

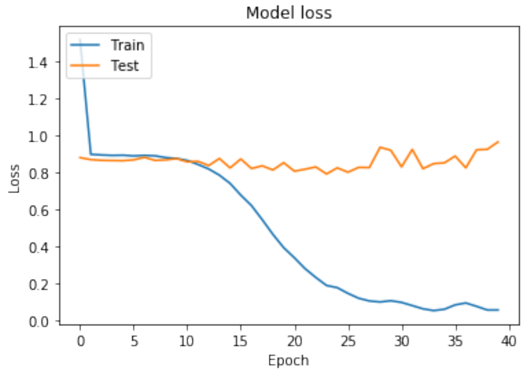

Figure: Loss for task A

	precision	recall	f1-score	support
Positive	0.68	0.97	0.80	506
Negative	0.88	0.21	0.34	67
Neutral	0.80	0.29	0.42	258
micro avg	0.70	0.70	0.70	831
macro avg	0.78	0.49	0.52	831
weighted avg	0.73	0.70	0.64	831
samples avg	0.70	0.70	0.70	831

Figure: Classification report for task A

Best result for task B after 40 epochs with Adam [ADM] optimizer and binary cross entropy loss function on each category:

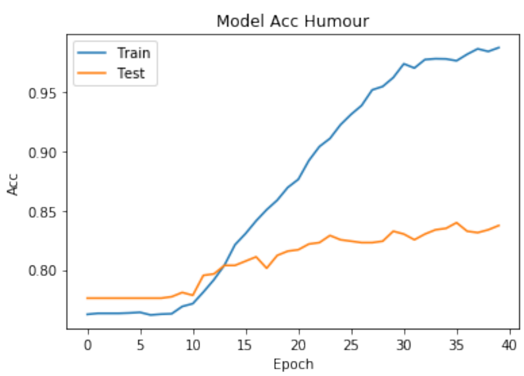

Figure: Accuracy for humour

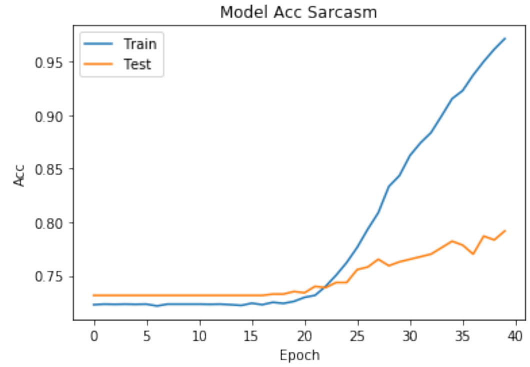

Figure: Accuracy for sarcasm

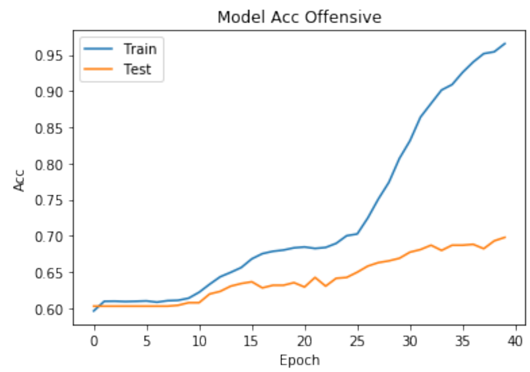

Figure: Accuracy for sarcasm

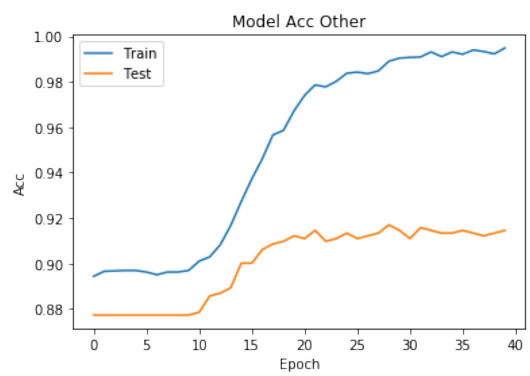

Figure: Accuracy for others

Figure: Loss for task B

	precision	recall	f1-score	support
Humour	0.83	0.99	0.90	645
Not Humour	0.90	0.31	0.46	186
micro avg	0.84	0.84	0.84	831
macro avg	0.87	0.65	0.68	831
weighted avg	0.85	0.84	0.80	831
samples avg	0.84	0.84	0.84	831

	precision	recall	f1-score	support
Sarcasm	0.87	0.26	0.41	223
Not Sarcasm	0.79	0.99	0.87	608
micro avg	0.79	0.79	0.79	831
macro avg	0.83	0.62	0.64	831
weighted avg	0.81	0.79	0.75	831
samples avg	0.79	0.79	0.79	831

	precision	recall	f1-score	support
Offensive	0.67	0.97	0.80	501
Not Offensive	0.88	0.28	0.42	330
micro avg	0.70	0.70	0.70	831
macro avg	0.77	0.63	0.61	831
weighted avg	0.75	0.70	0.65	831
samples avg	0.70	0.70	0.70	831

	precision	recall	f1-score	support
Other	0.94	0.32	0.48	102
Not Other	0.91	1.00	0.95	729
micro avg	0.91	0.91	0.91	831
macro avg	0.93	0.66	0.72	831
weighted avg	0.92	0.91	0.90	831
samples avg	0.91	0.91	0.91	831

Figure: Classification report for task B

You can find the source code of this paper over here.

4 Discussion

In this section, we will discuss the proposed method to solve the problem. First we define the baseline for each task:

- Task A:

$$Baseline = \max\{\frac{Positive}{Total\ Data}, \frac{Negative}{Total\ Data}, \frac{Neutral}{Total\ Data}\}$$

Baseline accuracy for this task is 0.6089 , however, our network's accuracy is 0.6955 hence we improved the accuracy by approximately 0.10 .

- Task B:

$$Baseline = Max\{\frac{Offensive}{Total\ Data}, \frac{Not\ Offensive}{Total\ Data}\}$$

$$Baseline = Max\{\frac{Sarcasm}{Total\ Data}, \frac{Not\ Sarcasm}{Total\ Data}\}$$

$$Baseline = Max\{\frac{Humour}{Total\ Data}, \frac{Not\ Humour}{Total\ Data}\}$$

$$Baseline = Max\{\frac{Other}{Total\ Data}, \frac{Not\ Other}{Total\ Data}\}$$

Baseline accuracy for each class of this task is 0.7761, 0.7316, 0.6028 and 0.8772 for humor, sarcasm, offensive and others ,respectively. Average accuracy for baseline is 0.7469, however, our network's accuracy is averagely 0.8104 hence we improved the accuracy by approximately 0.07 .

One of the merits of our network is the fact that we did not use pre-trained networks such as Inception or VGG, therefore, our training process uses less resources and is faster by a fraction of time. We have used a network consisting of some CNN and LSTM layers instead which is way faster than those massive pre-trained networks.

References

[8MA] 8MA. *Task 8: Memotion Analysis.* URL: https://competitions.codalab.org/competitions/20629.

[ADM] ADM. *Adam.* URL: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.

[GLO] GLO. *Glove.* URL: http://nlp.stanford.edu/data/glove.6B.zip.

[KRS] KRS. *Keras.* URL: https://keras.io.

[SEM] SEM. *SemEval-2020 International Workshop on Semantic Evaluation.* URL: http://alt.qcri.org/semeval2020/index.php?id=tasks#.

[TF] TF. *TensorFlow.* URL: https://www.tensorflow.org.