Sentiment Analysis of Memes

Dhanraj Bhedi[1], Liyana Faisel[2], Deepika Thokala[3], Vedprakash Gupta[4] and Vinay pothu[5]

[1] Marwadi University, Gujarat
[2] Vidya Academy of Science & Technology, Kerala
[3] RVR & JC College of Engineering, Andhra Pradesh
[4] Techno India Njr Institute of Technology, Udaipur
[5] Vaagdevi College of Engineering, Telangana

{dhanraj0709,liyanafaisel,deepikampc1,vgstardeeg1998,pothuvinay248}@gmail.com

Abstract. Now a days, people share a lot of content on social media in the for memes to express their feelings. In this project we are try to predict meaning of that memes like it is positive, negative or neutral. For this we have to use the concept of image captioning and optical character recognition to retrieve data from memes and predict that meaning. And also, we used MS COCO data set for training image captioning model Coming to result we got 75% accuracy for our dataset.

Keywords: Sentimental Analysis, memes, optical character recognition, image captioning.

1 Introduction

Sentiment analysis is the classification of emotions that refers to the use of natural language processing (NLP), rule-based modeling and text analysis. It is mostly used for reviews and surveys response, social media and healthcare materials to voice of the customer material. Categorizing of polarity is the basic task of sentiment analysis of a text which can be present at a document or sentence. It is also called opinion mining. Now-a-days, people share a lot of memes on social media in the form of image and text format. When we receive a meme from a sender, we cannot tell about the sender emotions that it is happy, angry or neutral. This makes sentiment analysis more predominant.

Positive: These sentiments are present in positive words or sentences. Some text with positive sentiment indicates happiness, kindness, passion etc. For example: He is so brilliant, this view is so beautiful, etc.
Negative: Negative sentiments are present in negative words or sentences in a textual content or visual content. Some negative sentiment indicates hate, sadness, violent etc.

Negative sentiment has the chance to report, block a post. For example: This view is horrible, she is my enemy, etc.
Neutral: There can be some sentences which have not any emotion that is called neutral sentiment. For example: Here is a book on the desk, childhood is the time to play, etc.

This paper focus on classifying the sentiment of visual content memes accordingly usage of image-based memes in social media. As a part of this project, we aim to predict the emotional category of an image falls into three positive, neutral and negative. We do this by convolutional neural network for the task of emotion prediction and sentiment analysis. I have dataset of 6992 observations, with 60% positive, 31% neutral and 9% labeled as negative. Dataset is quite small and unbalanced so first of all I balanced it by unsampled technique and then applied different algorithm to train the model.

2 Related Work

There are only a few publications on Multi-modal Sentiment analysis i.e., for both text and images. Both [1] and [2] employed textual and visual contents for sentiment analysis, and used these texts and images independently to extract the features in a supervised manner to combine the results of prediction using a n-gram textual feature and mid-level visual features [3].

Visual sentiment analysis forecast the sentiment of an image and deals with classifying its polarity. One method of it is to represent the image in terms of SIFT (Scale Invariant Feature Transform) feature which detects and describe the local features in the image and color. Then employing classification algorithms like SVM or Naive Bayes [4,5] that weights the input feature so that output separates one class a positive and other class a negative value. In our project, Image Captioning is used to get the data of image and to convert the caption into vectors, we used the pre-trained GLOVE model.

Paper [6] implemented the text quality based on annotation which counts the whole annotations generated and determined the total sentiment score.

Sentiment accuracy can be calculated using Naive Bayes classifier, k-nearest neighbor and random forest. [7] judges sentiment polarity (positive, negative or neutral) of text data and analyze its sentiment score. A text is split into single sentence and words.

Inspired by these works on both textual and visual models, we rely on deep learning techniques to extract features of memes (text and images). Our work used supervised learning for sentiment analysis. Methodology Because we have the memes as our input, we have to consider the visual content as well as textual content in our sentiment analysis. In this Problem, we have identified the two possible approach to solve this problem. 1.Unimodel classification 2. Multimodel classification.

2.1 Unimodel Classification

In unimodel classification, we have used the VGG19 model [8] [11]. VGG-ImageNet is pretrained model which trained on more than millions of images. Here, we have used the technique called transfer learning to train the model for our dataset [10]. Transfer learning is technique of storing knowledge of one problem to use in different problem.

Here this is the architecture of VGG19 model where i have changed the fully connected layer.

```
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 224, 224, 3)       0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 224, 224, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 224, 224, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 112, 112, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 112, 112, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 112, 112, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 56, 56, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 56, 56, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 56, 56, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 28, 28, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 28, 28, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 28, 28, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 14, 14, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 14, 14, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 7, 7, 512)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 25088)             0
_________________________________________________________________
dense_1 (Dense)              (None, 3)                 75267
=================================================================
Total params: 14,789,955
Trainable params: 75,267
Non-trainable params: 14,714,688
_________________________________________________________________
```

Fig. 1. Architecture of VGG19 model

Also, our dataset is unbalanced. So, we have to balanced it. For that we have explored many techniques such as up sampling, down sampling etc. and we have figured out that the up sampled data provides us more accuracy in compare to others.

```
positive    4160
neutral     2201
negative     631
Name: overall_sentiment, dtype: int64
```

Fig. 2. Unbalanced dataset

Using this method, we obtain a testing accuracy of 70% on the sentiment analysis task. In this work, we have observed that standard data augmentation techniques like random cropping and mirroring of images increase the accuracy of model. We used Adam optimization for all the networks, and the base learning rate at a start value of 0.001 gave better accuracies.

2.2 Multimodel Classification

In multimodel classification, we have used two models for the sentimental analysis. First for the image caption and second for text classification based on caption and textual content [8].

For Image captioning, I have used 2 models. For feature extraction we have used transfer learning on Inception V3 and these features used as input of Recurrent neural network to predict the Image caption. We have used the transfer learning for feature extraction using Inception V3 pretrained model.

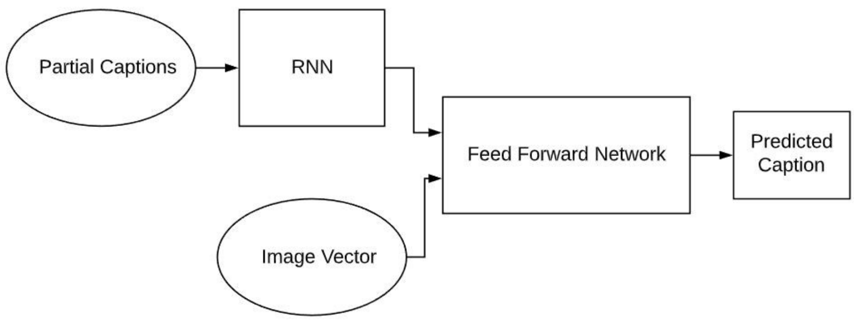

Fig. 3. Structure of multimodel classification

Here, we have captions as an input of RNN algorithm. So, we have to clean it before sending to model. Therefore, we have to do cleaning like lower casing, remove special characters, remove punctuations, remove emojis etc. Also, the recurrent neural network is not trained on words. Therefore, we have to convert the words to vectors. For that, I have used the pre-trained GLOVE embedding to map every word (index) to a 200-long vector. Now use this corresponding caption vector to image feature to train the image captioning model.

This is a model summary for Image captioning.

```
Model: "model_1"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            (None, 43)           0
__________________________________________________________________________________________________
input_1 (InputLayer)            (None, 2048)         0
__________________________________________________________________________________________________
embedding_1 (Embedding)         (None, 43, 200)      341400      input_2[0][0]
__________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 2048)         0           input_1[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 43, 200)      0           embedding_1[0][0]
__________________________________________________________________________________________________
dense_1 (Dense)                 (None, 256)          524544      dropout_1[0][0]
__________________________________________________________________________________________________
lstm_1 (LSTM)                   (None, 256)          467968      dropout_2[0][0]
__________________________________________________________________________________________________
add_1 (Add)                     (None, 256)          0           dense_1[0][0]
                                                                 lstm_1[0][0]
__________________________________________________________________________________________________
dense_2 (Dense)                 (None, 256)          65792       add_1[0][0]
__________________________________________________________________________________________________
dense_3 (Dense)                 (None, 1707)         438699      dense_2[0][0]
==================================================================================================
Total params: 1,838,403
Trainable params: 1,838,403
Non-trainable params: 0
__________________________________________________________________________________________________
```

Fig. 4. Summary of image captioning model

Second model is text classification. For that we have to use optical character recognition (OCR) to extract textual content from the image. Then we combined the OCR text and caption with respect to memes. Now for text classification, we have used the fastText. fastText is a library created by Facebook's AI Research lab for word embeddings and text classification. Here We have used the supervised learning algorithm of fastText to train the text classification model.

3 Results

There are so many datasets available for captioning the image, like Flickr, MS COCO, etc. But for the purpose of this case study, I have used the MS COCO dataset for train the image captioning model.

As we have two different approaches and we have 2 different results. In unimodel, we have got 70% test accuracy and 0.47 test loss. In multimodel, we have 75% accuracy for our dataset.

4 Conclusion

From both approaches, we can identify that the multimodel learning having more accuracy than the unimodel because it is based on both content visual as well as textual. Future work for this work is to increase the number of classes in dataset, optimize the optical character recognition to extract maximum and accurate data from memes and try to predict more accurate captions for the memes.

References

1. M. Wang, D. Cao, L. Li, S. Li, and R. Ji: Microblog sentiment analysis based on cross-media bag-of-words model, pages 76:76–76:80 (2014)
2. D. Cao, R. Ji, D. Lin, and S. Li: A cross-media public sentiment analysis system for microblog. Multimedia Systems, pages 1–8 (2014).
3. D. Borth, T. Chen, R. Ji, and S.-F. Chang: Sentibank: large-scale ontology and classifiers for detecting sentiment and emotions in visual content. In ACM MM, pages 459–460 (2013).
4. Stefan Siersdorfer, Enrico Minack, Fan Deng, and Jonathon S. Hare: Analyzing and predicting sentiment of images on the social web. In Proceedings of the 18th International Conference on Multimedia 2010, Firenze, Italy. ACM, 715–718 (2010).
5. Jianbo Yuan, Sean McDonough, •anzeng You, and Jiebo Luo: Sentribute: image sentiment analysis from a mid-level perspective. In Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining, WISDOM 2013, Chicago, IL, USA. ACM, 10:1–10:8 (2013)
6. Schukla: A Sentiment analysis of document based on annotation, CORR Journal, Vol. abs/1111.1648 (2011).
7. Esuli A., & Sebastiani, F.: SentiWordNet: A High-Coverage Lexical Resource for Opinion Mining, Kluwer Academic Publishers. Printed in the Netherlands, (2006).
8. Douwe Kiela, Hamed Firooz∗, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, Davide Testuggine: The Hateful Memes Challenge: Detecting Hate Speech in Multimodel Memes (2020).
9. Jean H. French: Image Based Memes as sentiment Predictors (2017)
10. Stuti Jindal, Sanjay Singh: Image sentiment analysis using deep convolutional neural networks with domain specific fine tuning (2015)
11. Namita Mittal, Divya Sharma, Manju Lata Joshi: Image Sentiment Analysis Using Deep Learning (2018)