MemeSem:A Multi-modal Framework for Sentimental Analysis of Meme via Transfer Learning

Raj Ratn Pranesh [1] **Ambesh Shekhar** [1]

Abstract

In the age of the internet, Memes have grown to be one of the hottest subjects on the internet. But despite their huge growth, there is not much attention given towards meme sentimental analysis. In this paper, we present MemeSem- a multimodal deep neural network framework for sentiment analysis of memes via transfer learning. Our proposed model utilizes VGG19 pre-trained on ImageNet dataset and BERT language model to learn the visual and textual feature of the meme and combine them together to make predictions. We have performed a comparative analysis of MemeSem model with various baseline models. For our experiment, we prepared a dataset consisting of 10,115 internet memes with three sentiment classes- (Positive, Negative and Neutral). Our proposed model outperforms the baseline multimodals and independent unimodals based on either images or text. On an average MemeSem outperform the unimodal and multimodal baseline by 10.69% and 3.41%.

1. Introduction

Sentimental analysis is one of the highly researched topics in Natural Language Processing. Many research works are available where the labeled dataset, primarily textual, is used for the sentiment analysis of social conversations such as social media posts, discussions, comments, customer review of products in order to determine deeper context of people's perception and behaviour online. For an instance, authors(Peirson et al., 2018) utilize labeled dataset for the sentimental analysis of tweets, authors(Asghar et al., 2015) discuss about various techniques to analyze the user sentiment over YouTube video using the comments posted by the users.

In the last few years, the research community has developed a growing surge to study the granularity of the digital constructs called memes and the increasing multi-modalities of social media. Shifman(Shifman, 2014) investigates about internet memes and their importance in digital culture. The influx of internet memes contributing substantially to the network overload (Peirson et al., 2018) has drawn our attention towards the analysis of meme sentiments.

A large group of people on social media platforms such as Facebook, Instagram, and Twitter use memes to convey or express their feeling and opinion instead of just using text. For example, people use hate memes instead of hate speech in online social media. Putting memes in comment section is also a very common practice. With the rapidly growing popularity of meme in social media it is a very crucial and interesting task to classify the memes and provide a framework for filtering the meme on internet.

The **motivation** of our work to leverage the multimodal aspect of memes in MemeSem lies as follows: (i) different modalities of multimodal documents i.e. meme carry different aspects, (ii) meme contain text and image which complement each other and feature extracted from both modalities would be essential for the prediction, (iii) analyzing the legitimate genre of Internet meme require a very strong multimodal framework, models solely based on visual or text feature would not be enough to completely understand the meme sentiment.

Challenges: The most challenging task is to determine the type of sentiment in a given meme (and the concept of humour and sarcasm in general) which could be difficult for humans to trace and distinguish themselves,let alone machines. So it should be noted that this task might be a long way from getting solved.Another challenge faced when working with memes, is the use of texts and images together. In order for the networks to work with this kind of complex problems, it is required for them to be deep and complex enough. This also leads us to another challenge, which is extracting the text out of the image and prepossessing them so that they can be used as input in the deep learning model . The last hurdle of this task is that in the absence of sufficient training data, augmentation techniques can't be used to generate new instances.

[1] Birla Institute of Technology, Mesra, India. Correspondence to: Raj Ratn Pranesh <raj.ratn18@gmail.com>.

Accepted at the 4^{th} *Lifelong Learning Workshop at ICML 2020*, Vienna, Austria, PMLR 108, 2020. Copyright 2020 by the author(s).

The main **contributions** of the paper are- (i) design, explore and compare performance of various mulimodal framework for meme sentimental analysis, (ii) present MemeSem: a transfer learning based multimodal framework for sentimental analysis of meme. The objective of proposed multimodal architecture of MemeSem is to detect the sentiment class(positive, negative or neutral) to which a given meme belongs. It combines the visual and linguistic feature of meme to make predictions. MemeSem multimodal architecture consists of Bidirectional Encoder Representations from Transformers(Devlin et al., 2018) to extract contextual features and VGG-19(Simonyan & Zisserman, 2014) pre-trained on ImageNet(Russakovsky et al., 2015) to learn the image features of meme. The representations generated from visual and linguistic models are then merged to provide the final meme vector representation. This vector is then used for the sentimental analysis task.

The paper is divided in the following sections. Section 2 provides brief related works done in the field of meme analysis using multimodal data. In Section 3, we discuss about the methodology we adopted to develop MemeSem. Followed by Section 4 which contains information of dataset used for the experiment. Section 5 details overview of baseline models and the experiment setup. In the following Section 6, we put together the experiment results and conduct a detailed analysis. Finally, with Section 7 we conclude the paper.

2. Related Work

Sentimental analysis in multimedia domain is a highly researched topic. Social media platforms are flooded with textual, visual and multimodal data. That being said, very less attention has been given to the multimodal and visual sentiment analysis as compared to text based sentimental analysis. Daniel Miller and Jolynna Sinanan (Miller & Sinanan) talked about the how important the images are, as a source of information for precisely inferring sentimental states of users because of their omnipresence on social media platforms as a medium for people to express themselves. Authors(You et al., 2015) have demonstrated sentimental analysis on Flicker dataset with domain transfer from Twitter using convolutional neural networks. However, studies(Gong et al., 2014),(Guillaumin et al., 2010) have shown that textual features combined with image can significantly improve the image annotation task.

This brings us to the multimodal data called memes. In the paper(Bauckhage, 2011), author talks about presence of memes on internet and conjecture that the main source of internet memes spread is social media platforms. One of the important works on meme analysis and clustering in social media is (Ferrara et al., 2013) where authors performed the task of meme detection by clustering messages from various large streams of social data such as Twitter. Meme dataset used here was limited to text such as messages, tweets, trending hashtags.

Modern memes are images with embedded text. At the time of meme sentimental analysis one should focus on extracting features from both modalities. Work on multimodal dataset such as(Hu & Flaxman, 2018) where sentimental analysis is done using Tumblr posts consist of images and their caption. Our prime inspiration for sentimental analysis of meme was derived from this work(Sabat et al., 2019). Successful results in multimodal analysis of meme for detecting offensive memes on internet have been done by the authors.

Although above mentioned works[13, 12] are very valuable and address the task of multimodal sentimental analysis very well, there are a few weaknesses and limitations possessed by these previous works. In the paper(Hu & Flaxman, 2018), experiments are done on Tumblr posts which are not exactly memes. A notable difference is that, in memes, texts are embedded in image but in Tumblr posts, images have separate captions. In paper(Sabat et al., 2019), the author carried out binary classification for detecting either meme is offensive or not, without providing a complete generic sentimental analysis for internet memes.

In order to overcome these shortcomings, we have designed MemeSem. Memesem takes meme as input, analyses and extracts the visual and linguistic features, followed by concatenating both features to produce a multimodal vector representation of meme, which is then finally used to classify the meme into positive, negative or neutral sentiment. Hence, MemeSem provides a framework for classification and filtering of large scale dataset of internet memes.

3. Methodology

In this section, we have outlined our transfer learning based multimodal architecture, basically divided into three submodules. The first sub-module involves a language model that contributes to the contextual text features extraction. The second sub-module is responsible for the extraction of visual features from a meme image. Finally, at last we have a fusion sub-module where feature representation received from first and second sub-modals are combined together to obtain a meme feature vector.

3.1. Textual feature extractor

In our multimodal framework, we have used a pretrained BERT$_{base}$(Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) to extract high quality textual feature vector. BERT is a powerful language model which represent the sentences or words by taking ac-

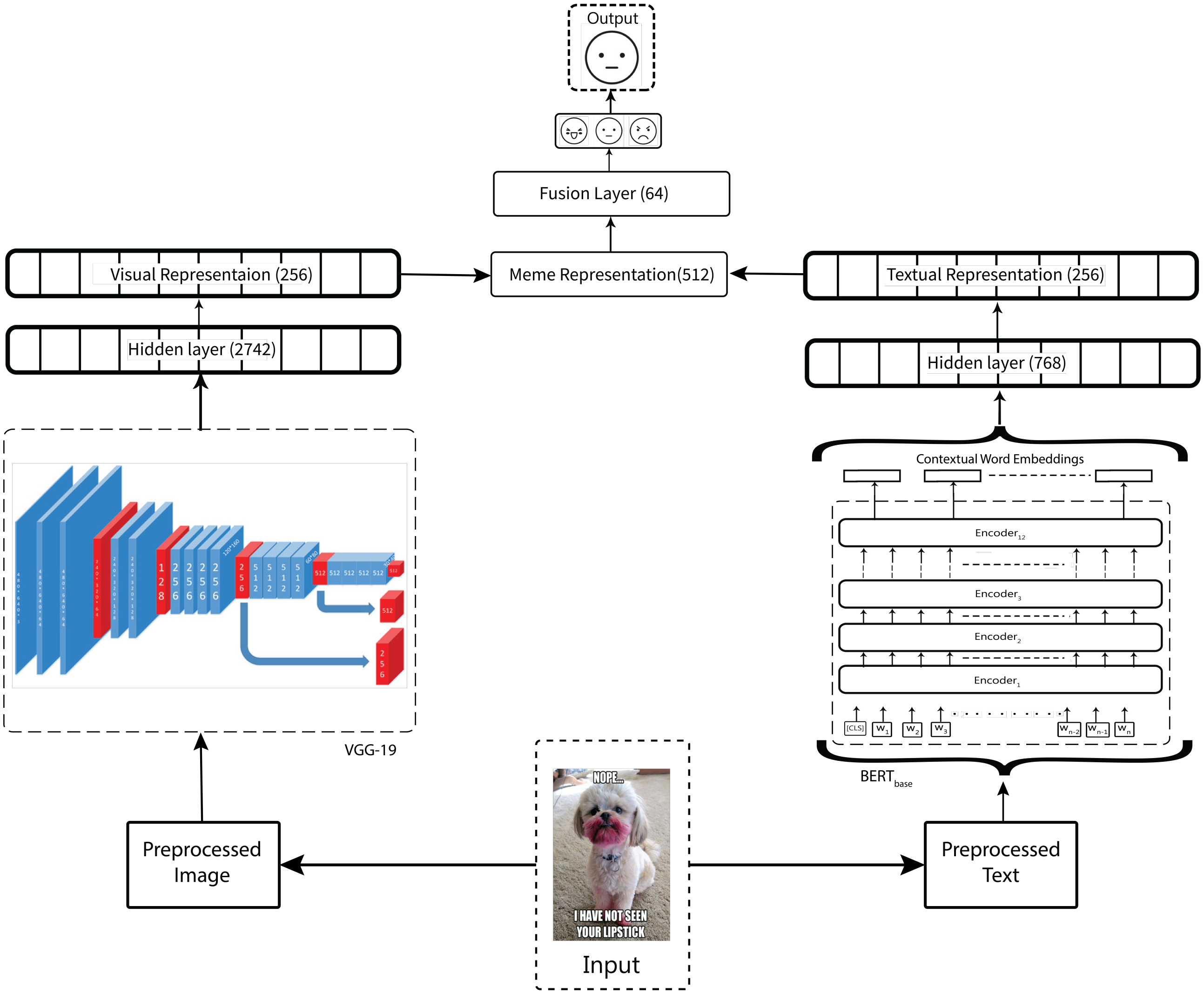


Figure 1. A detailed structure of MemeSem architecture with visual sub-modal(left) and textual sub-modal(right).

count of contextual and semantic meaning present in them. In this phase we used BERT$_{base}$ uncased model with 12 encoding layers, 768 hidden state, 12 attention heads and 110 million parameters. We supply a sequence of words as inputs which goes through all the stacked encoding layers, extracting essential features from the context.

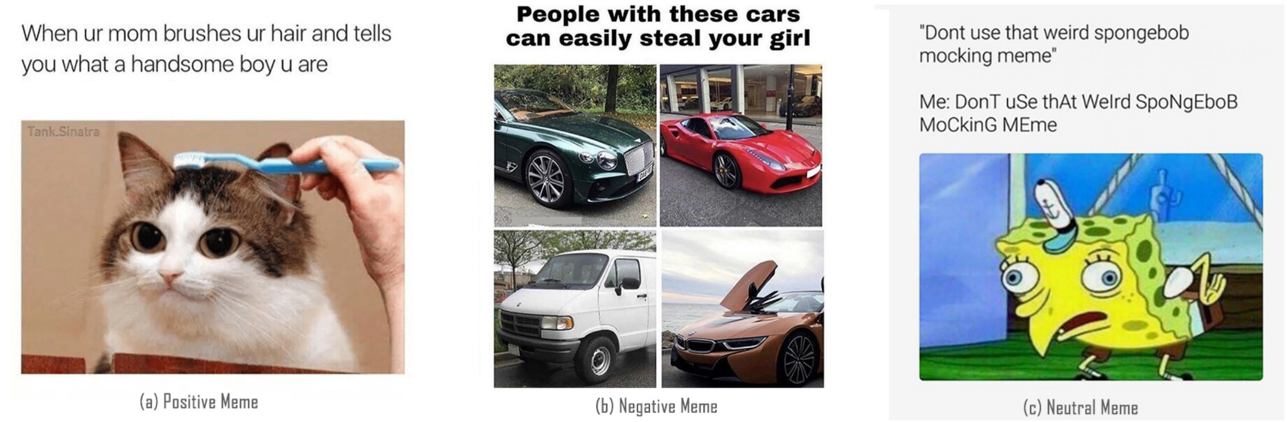
In the Figure 1, as we can see encoders are stacked over each other followed by a sequence of tokens represented as w_i. The input flows up through the stacks. Each of the tokens pass through a self-attention applied by each layer and through the feed forward network passes it's results, and then give it to the encoder present next. On the right part of the figure 1 a detail architecture of the BERT language model is shown. The [CLS] token stands for classification and it represents sentence-level classification and w$_i$ represents the sequence of word tokens that are supplied as input to the BERT for feature extraction. The textual feature extractor outputs g_i the contextual word embeddings of size 768(hidden_size in BERT) for every token , passes to a fully connected layer of dimension 768. The output vectors are then supplied to a fully-connected layer of size 256 giving a final textual representation dimensions of 256. These text features hold meaningful information and are denoted as T_f.

$$T_f = \sigma(W_t \cdot g_i) \tag{1}$$

where W_t is the weight matrix of the fully connected layer in the textual feature extractor.

3.2. Visual Feature Extractor

This module deals with visual feature of image from meme. Images contain thousands of pixel values in several color


When ur mom brushes ur hair and tells you what a handsome boy u are
Tank.Sinatra
People with these cars can easily steal your girl
"Dont use that weird spongebob mocking meme"
Me: DonT uSe thAt WeIrd SpoNgEboB MoCkinG MEme

(a) Positive Meme (b) Negative Meme (c) Neutral Meme

Figure 2. An instance of our dataset (i)Positive Meme- shows affection (ii)Negative Meme- shows inappropriate thinking (iii)Neutral Meme- shows no meaning

channels, edges between two image regions, interest points and regions, ridges and their correlation and relationship characterizes the class and enables drawing a separation. These are some important features of an image. MemeSem utilizes the pretrained VGG networks (Simonyan & Zisserman, 2014) to extract essential features from the image. We used VGG19 pre-trained on ImageNet (Russakovsky et al., 2015) dataset. The pre-processed meme image with the size of (224,224,3) is passed as an input and a 4096 dimensional vector denoted as V_g is extracted from the last second layer of VGG19. The visual feature vector is then passed through a pair of fully connected layers with hyper-parameters as referred in Table 2. It is then passed through two fully connected layers of dimension 2742 and 256. This module outputs the final visual feature V_f.

$$V_f = \sigma(W_I \cdot V_g) \tag{2}$$

where V_g represents the final visual feature output from the pre-trained VGG-19 and W_I represents the fully connected layer weight matrix present in the visual sub-modal.

3.3. Multimodal fusion

We combine the output vectors from the two separate modalities(textual and visual) using concatenation method to obtain the required meme feature representations. The output from concatenation layer is passed down to a pair of fully connected layers of weights W_F for meme's sentiment classification.

$$\beta_i = relu(W_F \cdot [T_f \cdot V_f]) \tag{3}$$

$$\hat{y} = softmax(\beta_i) \tag{4}$$

4. Dataset

In this paper we prepare a dataset of 10,115 memes which were weakly labeled into three classes Positive, Negative and Neutral. Our modality works on two fundamental features, one is image for visual analysis and another one is text for textual analysis. To train our multi-model we need sentiment label associated to each meme. Each row of the dataset contains pre-processed OCR extracted text and path to the image file in the directory along with labels. There are fundamentally 3 types of label as shown in figure2: Positive(1), Neutral(2) and Negative(0). Differentiation of Overall sentiment between memes has been done during the extraction of the dataset.

Internet has abundant of resources and that's why we searched through a wide variety of websites to find useful resources which might help in getting us meaningful data. We came across google-image-downloader[1] and PRAW[2]. Google Images Download is a python program to search keywords or key-phrases on google images and download these images. PRAW is a Python Reddit API Wrapper that allows for simple access to Reddit's API. We use these python packages to get access to our data and download them, and as for different sentiment images, we searched keywords like racist memes, Jews meme and similar types of keywords for google-image-download, whereas for PRAW we gone through sub-reddit pages and found pages like r/DarkMemes[3] for negative sentiment memes, r/wholesomememes[4] for positive sentiment memes and r/antimeme[5] for neutral sentiment memes and downloaded

[1] https://github.com/hardikvasa/google-images-download

[2] https://github.com/praw-dev/praw

[3] https://www.reddit.com/r/darkmemes/

[4] https://www.reddit.com/r/wholesomememes/

[5] https://www.reddit.com/r/antimeme/

Table 1. Class distribution of meme dataset

CLASS	NUMBER OF MEME
POSITIVE	3,490
NEGATIVE	3,300
NEUTRAL	3,325
TOTAL	10,115

posts with high ratings to our local system. Table 1 reports the number of total memes collected for each class.

5. Experiment

In this section, we elaborate the pre-processing pipeline, the baseline models and sequential experimental setup.

5.1. Preprocessing pipeline

Our dataset contains processed texts, path to the meme image and labels for training. We need to further process the data to pass these through our modalities in the architecture.

5.1.1. TEXTUAL DATA

We need to extract the textual information from memes in order to process them separately. We use Tessaract-OCR[6]. Tessaract-OCR is an LSTM(Schmidhuber & Hochreiter, 1997) network based OCR engine which helps in detection and recognition of texts embedded in an image. Tessaract-OCR extracts the embedded texts from the meme dataset. We stored these texts in a comma separated value(CSV) files for further operations. Some of the extracted texts had abnormalities such as incomplete or missing words, therefore they were manually corrected and annotated, and acronyms were replaced with their complete meaningful words.

5.1.2. VISUAL DATA

For image pre-processing, we first import memes from directory, followed by reshaping the size. Since we are using VGG-19 pretrained network, we need to reshape our image shape according to VGG-19's input_shape which is (224,224,3). These images are converted into arrays and every pixel value is then normalised before passing in the model.

5.1.3. TRAIN AND VALIDATION DATA

After preprocessing the textual and visual data, we prepare the training data as well as validation data to perform the experiments. The processed meme data was spilt into training and validation set consisting of 80% and 20% of the data respectively.

5.2. Experimental Setup of Baselines

For a comprehensive evaluation of MemeSem, we compare our model with the unimodal and multimodal baseline methods. All the models are trained on the training set and evaluated on the validation data. Following are the models:

5.2.1. SINGLE MODALITY MODELS

To compare the multi-modals we experimented with two uni-modal models exclusively based on textual and visual feature of meme as described below-

- **Textual:** This model uses textual data present in posts to classify meme's sentiment. BERT expects input in specific format. BERT works on contextual data therefore each word is represented as a n-dimensional vector with its respective ID from tokenizer's dictionary along with segement ID to distinguish between sentences and an input mask to distinguish between tokens and padded sequence. Individual post's pre-processed data are then fed into a BERT to extract textual features. Textual features are then fed into a 32-dimensional fully connected layer with a softmax activation function coupled with the BERT that is responsible for making final predictions.

- **Visual:** The visual model is based on VGG19(Simonyan & Zisserman, 2014). It takes images of the meme as input and then classifies the meme's sentiment. The re-sized images after pre-processing are fed into the pre-trained VGG-19 network to extract the visual features from the fully-connected layer. As we did in the textual model, the visual features are then passed into a 32-dimensional fully connected layer in order to make the final class prediction of the meme.

5.2.2. MULTI-MODALITY MODELS

To compare MemeSem, we experimented with two distinct multimodal leveraging both modalities of meme. For visual feature extraction we used VGG19(Simonyan & Zisserman, 2014) and for textual feature we used Long short-term memory(Schmidhuber & Hochreiter, 1997) and Bidirectional Long short-term memory as described below-

- **VGG19 + LSTM:** This baseline model consists of VGG19 and LSTM(Schmidhuber & Hochreiter, 1997) for image and text processing respectively. For visual features, we extract the output of size 4096 from a VGG19(pre-trained on ImageNet(Russakovsky et al., 2015)). Visual vectors are then passed through a pair

[6] https://github.com/tesseract-ocr/tesseract

Table 2. Overview of Hyperparameters

Parameter	Value
Dropout	0.5
# of dense layers (text)	2
# of neurons in dense layer	768,256
# of dense layers(image)	2
# of neurons in dense layer	2742,256
# of neurons in concatenation layer	64
Sequence length	42 words
Batch size	64
Optimizer	Adam
Learning Rate	0.00002
Total	10,115

of fully connected layers with size 2742 and 256 respectively. For textual feature, we load the word embeddings to the LSTM, using embedding weights from GloVe(Pennington et al., 2014) applied in the embedding layer. Each word embedding has a vector dimension of 300. These word embeddings are then fed to the LSTM network of hidden size of 300, followed by fully connected layer of size 256. The 256-dimensional visual and textual vectors are then concatenated and resultant feature vector of 512 is passed through a hidden layer of size 64 coupled with a softmax function to make final predictions. We use L2-regularizer on the weights to prevent overfitting, also used Adam(Kingma & Ba, 2014) optimizer to get the optimal parameters for the model.

VGG19 + BiLSTM: This baseline model is consist of VGG19 and Bi-LSTM for image and text processing respectively. For visual feature, we extract the output of size 4096 from a VGG19(pre-trained on ImageNet(**?**)). Visual vectors are then passed through a pair of fully connected layers of size 2742 and 256 respectively. For textual features, we load word embedding to the Bi-LSTM by using embedding weights from GloVe(**?**) applied in the embedding layer. Each word embedding has a vector dimension of 300. These word embedding are then fed to the Bi-LSTM network of each hidden size of 300, followed by fully connected layer of size 256. The 256-dimensional visual and textual vectors are then concatenated and resultant feature vector of 512 is passed through a hidden layer of size 64 coupled with a softmax function to make final predictions. We use L2-regularizer on the weights to prevent overfitting, also used ADAM(**?**) optimizer to get the optimal parameters for the model.

5.3. Experimental Setup of MemeSem

At first we start by filtering and pre-processing the meme data as described in the above section. In the text modality,the max sequence length of processed text data was 52 but only 3% texts were greater than 42, so we decided to set max length limit to 42. If the sequence of tokens are greater than limit, then the sequence will be truncated otherwise padded from right respectively. So, the processed textual data consists of sequence of English words from which a maximum input sequence of 42 tokens are fed in the model. For the visual modality, all the images are resized to (224x224x3) according to the VGG19's input size standard.

For the extraction of textual features out of meme text data we used pre-trained BERT$_{base}$ uncased model available on Tensorflow Hub[7]. The tokenized and padded input sequence are passed through 12-encoding layers of the BERT model resulting in text embedding of dimension 768. Output from BERT is passed through a pair of fully connected layers with number of neurons of 768 and 384 respectively. For the visual features extraction, the meme images were resized by the pre-processing pipeline and then passed to the VGG19 pre-trained on ImageNet(Russakovsky et al., 2015) to produce a feature vector of 4096 which is extracted and passed through the pair of fully connected layers of number of neurons of 2742 and 256 respectively. The 256-dimensional and 256- dimensional vectors from both the modalities(textual and visual) are concatenated and passed into a pairs of fully connected layer of sizes 36 and finally followed by a classification layer of size 3 with softmax activation function. All the dense layer present in the architecture has a RELU activation function and a dropout layer with a probability of 0.5 to avoid overfitting. The MemeSem model was trained on a batch size of 64 and Adam(Kingma & Ba, 2014) optimizer for training the model.

5.4. Hyperparameter Setting for MemeSem

In the table 2 we have listed all the hyperparameters used for the experiment for MemeSem and baseline models. The selection of hyperparameters for the model is completely based on the experiments. We perform several iterations with all possible and distinct sets of hyperparameter. After analyzing the model performance on various configurations of hyperparameter we selected the one giving the most optimum result on our dataset. The training code, dataset and model used for experiment are publicly available for reproducibility purpose.[8].

6. Result and Discussion

In this section we discuss and analyse the experiment results. Table 3 shows the accuracy %, precision, recall and F1-score

[7] https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H768_A-12/1

[8] https://github.com/AmbiTyga/MemSem

Table 3. Models performance scores(in %)

MODEL	*Accuracy*	*P*	*R*	*F1*
VGG19	55.45	56.06	54.46	55.24
BERT$_{base}$	57.41	57.12	58.25	57.67
VGG19 + LSTM	62.34	63.03	59.09	61.01
VGG19 + BI-LSTM	64.08	64.17	60.32	62.18
MEMESEM(VGG19 + BERT$_{base}$)	**67.12**	**68.02**	**64.21**	**66.06**

obtained by two unimodals(VGG19 and BERT$_{base}$) and three multimodals(VGG19 + LSTM, VGG19 + Bi-LSTM and VGG19 + BERT$_{base}$). We observed that performance of text-only model was slightly better(around **3%**) then image-only model. This might be because meme's text gives more detailed context of the meme as compared to meme's image, therefor it is much easier for the model to understand and recognize the sentiment behind the meme.

All the multimodals significantly outperform the unimodal establishing the fact that fusion of features extracted from different modalities through transfer learning greatly increase the model's performance. Our model MemeSem reported an accuracy of **67.12** which was on an average **10.69%** more than unimodals. Other baseline multimodal also outperform the unimodals by **5.91%**(VGG19 + LSTM) and **8.65%**(VGG19 + Bi-LSTM) for on an average.

In multimodal setup we experimented with different textual feature extraction model. Using Bi-LSTM for extracting textual feature had a slight improvement(around **5.48%**) as compared to LSTM. The major improvement was noticed while using BERT for textual feature extraction. MemeSem outperform the baseline multimodal(VGG19 + LSTM) by **4.78%** and multimodal(VGG19 + Bi-LSTM) by **2.04%**.

7. Conclusion

In this paper we presented a multi-modal framework which leverages transfer learning for detecting and classifying internet memes in different sentiment classes based on the textual and visual feature. In our study we compared various model both single and multi-modal for the task and finally we can conclude that the presence of both visual and linguistic feature plays a crucial role in machine's learning in identifying the sentiment behind the memes. We also showed that using a powerful pre-trained language model(Devlin et al., 2018) for textual feature extraction improves the contextual understanding of deep learning model and gives better results in a multi-modal setup. The experiments results shows that one application of MemeSem can potentially be the identification and filtration of internet memes on social media platforms but it is still far from perfect and there is still room for improvement in the proposed design. Future work and possible experiments that can be done such as: (i) Experimenting with various models for textual and visual feature extraction, (ii) Increasing the dataset size would definitely improve the performance, (iii) Providing additional features to the model such as Sarcasm, Humor present in meme would also enhance the performance.

References

Asghar, M. Z., Ahmad, S., Marwat, A., and Kundi, F. M. Sentiment analysis on youtube: A brief survey. *arXiv preprint arXiv:1511.09142*, 2015.

Bauckhage, C. Insights into internet memes. In *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Ferrara, E., JafariAsbagh, M., Varol, O., Qazvinian, V., Menczer, F., and Flammini, A. Clustering memes in social media. In *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, pp. 548–555. IEEE, 2013.

Gong, Y., Ke, Q., Isard, M., and Lazebnik, S. A multi-view embedding space for modeling internet images, tags, and their semantics. *International journal of computer vision*, 106(2):210–233, 2014.

Guillaumin, M., Verbeek, J., and Schmid, C. Multimodal semi-supervised learning for image classification. In *2010 IEEE Computer society conference on computer vision and pattern recognition*, pp. 902–909. IEEE, 2010.

Hu, A. and Flaxman, S. Multimodal sentiment analysis to explore the structure of emotions. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 350–358, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Miller, D. and Sinanan, J. Forthcoming. visualising facebook.

Peirson, V., Abel, L., and Tolunay, E. M. Dank learning: Generating memes using deep neural networks. *arXiv preprint arXiv:1806.04510*, 2018.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Sabat, B. O., Ferrer, C. C., and Giro-i Nieto, X. Hate speech in pixels: Detection of offensive memes towards automatic moderation. *arXiv preprint arXiv:1910.02334*, 2019.

Schmidhuber, J. and Hochreiter, S. Long short-term memory. *Neural Comput*, 9(8):1735–1780, 1997.

Shifman, L. *Memes in digital culture*. MIT press, 2014.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

You, Q., Luo, J., Jin, H., and Yang, J. Robust image sentiment analysis using progressively trained and domain transferred deep networks. In *Twenty-ninth AAAI conference on artificial intelligence*, 2015.