Only text? only image? or both? Predicting sentiment of internet memes

Pranati Behera, Mamta, Asif Ekbal
Department of Computer Science and Engineering
Indian Institute of Technology Patna, India
pranati5079@gmail.com, {mamta_1921cs11, asif}@iitp.ac.in

Abstract

Nowadays, the spread of Internet memes on online social media platforms such as Instagram, Facebook, Reddit, and Twitter is very fast. Analyzing the sentiment of memes can provide various useful insights. Meme sentiment classification is a new area of research that is not explored yet. Recently SemEval provides a dataset for meme sentiment classification. As this dataset is highly imbalanced, we extend this dataset by annotating new instances and use a sampling strategy to build a meme sentiment classifier. We propose a multi-modal framework for meme sentiment classification by utilizing textual and visual features of the meme. We found that for meme sentiment classification, only textual or only visual features are not sufficient. Our proposed framework utilizes textual as well as visual features together. We propose to use the attention mechanism to improve meme classification performance. Our proposed framework achieves macro F1 and accuracy of 34.23% and 50.02%, respectively. It increases the accuracy by 6.77% and 7.86% compared to only textual and visual features, respectively.

1 Introduction

The rapid growth of users on social media platforms leads to new ways of spreading information. Meme nowadays has become one of the most popular words for social media. A meme is an idea, the way in which a person behaves in response to a particular situation or a manner that spreads from one person to another within a culture. Spreading of memes on social media platforms such as Facebook, Instagram, Reddit, and Twitter is very fast.

Sentiment analysis is a growing field of Natural Language Processing (NLP), aiming to identify the polarity of opinion. Sentiment can be positive, negative or neutral (Pang and Lee, 2005). Sentiment analysis has a vast number of applications in real life, including the product's recommendation to a user based on opinions provided by other users (Pang et al., 2002), in political uses (Bakliwal et al., 2013), etc. Memes play an important role in handling various political battles or public relations on social media platforms.

The most common practice in sentiment analysis is finding the sentiment of textual content crawled from Twitter, product reviews, hotel reviews, etc. Existing literature has mostly addressed the problem of sentiment analysis primarily using textual contents (Xu et al., 2019; Edara et al., 2019; Medhat et al., 2014; . et al., 2020). But with the growing social media, users are expressing their opinions through text and the image. Hence, researchers nowadays are also giving attention to sentiment analysis in multi-modal content (You et al., 2016; Ortis et al., 2020; Man et al., 2019). Spreading of memes is also very fast, but meme analysis is yet to be explored. Recently, SemEval-2020 proposed a task to detect the meme's polarity, which can fall into three predefined classes: positive, negative, or neutral (Sharma et al., 2020). This is the very first attempt towards the meme sentiment analysis.

To analyze the sentiment of memes, the text-only approaches may not be sufficient. For example, consider the meme given in Figure 1, if the only textual content is considered, then the sentence 'FINALLY GETS JOB INTERVIEW' seems to have a neutral sentiment (no explicit positive words are used). However, if we also consider the visual information of meme, as shown in Figure 1, then we can say overall sentiment is positive. Hence, to analyze memes, both text and visual features have their own importance.

In this paper, we work on the SemEval-2020 Task-8 dataset to detect the sentiment of memes. But this dataset is imbalanced. Hence we extend this dataset by adding more training instances for

Proceedings of the 17th International Conference on Natural Language Processing, pages 444–452
Patna, India, December 18 - 21, 2020. ©2020 NLP Association of India (NLPAI)

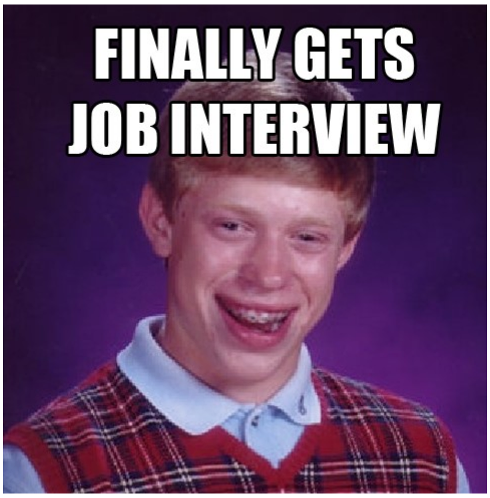

Figure 1: Meme example

balancing purposes and then propose a multi-modal framework based on deep neural networks to classify the sentiment of the meme into one of the predefined classes, namely positive, negative, and neutral. We use a multi-modal framework with attention applied to both image and text to find out important regions and important words. Thereafter, to combine the image and textual modality, we use a fully connected layer that tries to find the relation between textual and visual features and finally produces a combined feature vector. We evaluate the proposed approach using accuracy and Macro F1 score on the test set of the SemEval-2020 dataset. We get the macro F1 of 34.23%, and accuracy of 50.02%, respectively, which is higher than the SemEval baseline, i.e., Macro F1 of 0.21%.

The rest of the paper is organized as follows. Section 2 describes the related work. Section 3 describes the methodology for classification. Section 4 describes the data collection, annotation, and experimental setup. Section 5 describes the results and detailed error analysis. Section 6 concludes the paper and describes future research plans.

2 Related work

This section describes the works on sentiment analysis for text as well as for multi-modal content. (Murty and Allu) proposed an approach for finding the sentiment analysis on text reviews by using Long Short Term Memory. (Agarwal et al., 2011) proposed the framework to classify the sentiment of tweets into positive, negative and neutral class using prior polarity scoring, which is based on the prior polarity of words. (Li et al., 2019) proposed a sentiment-feature-enhanced deep neural network (SDNN) to detect the sentiment of text by deep neural network integrated with sentiment linguistic knowledge via attention mechanism. (Mozetič et al., 2016) proposed a framework for textual sentiment analysis using lexicon based and machine learning based approach. Sentiment is predicted from the set of sentiment-bearing words identified in the text using lexicons. (Ghiassi and Lee, 2018) proposed a set of domain transferable Twitter lexicons, obtained from tweets for the task of sentiment analysis. (Kumar and Jaiswal, 2017) proposed a model to detect the sentiment of images using Convolutional neural network. They used Flicker images dataset to train their model and Twitter images dataset for testing. (Akhtar et al., 2020) proposed a stacked ensemble model for predicting the degree of intensity for sentiment and emotion. They used multi-layer perceptron network to combine outputs of feature based models and deep learning models. (Poria et al., 2018) explored different deep-learning based architectures for multi-modal sentiment classification. They used deep convolutional neural network (CNN) to extract features from the visual and text modalities. (Jiang et al., 2020) proposed a fusion-extraction network model for multi-modal sentiment analysis. Their proposed model learned two types of representations, visual-specific textual representations and textual-specific visual representations using interactive information fusion mechanism.

Above mentioned works are either for text or multi-modal content. Meme classification has not been explored much in detail. So we proposed a framework for meme classification by utilizing text written on it and image features.

3 Methodology

This section represents our proposed methodology in detail. We develop a multi-modal neural network that learns from the two modalities, *viz.* textual and visual. For text modality, our model takes as input the embedding representation of each word present in the OCR extracted text. Further, we use Convolutional Neural Network (CNN) to learn textual features, and then we apply attention to the output of CNN to extract the most relevant features for classification. We use the pretrained model VGGNet to extract the visual features for image modality, and then we apply attention to the extracted features to detect important visual features for classification. Finally, both the features

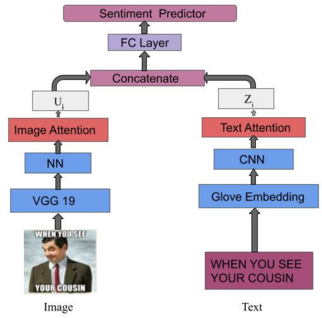

Figure 2: Proposed architecture

are fused with the help of a fully connected layer. The overall architecture of our proposed model is shown in Figure 2.

3.1 Textual features

In this section, we discuss the textual features, how they are given as input to our system, and how to apply attention to the features extracted.

3.1.1 Embedding Layer

The embedding layer takes the input as a sequence of words present in the sentence. For each word *w* present in the sentence, a lookup matrix is created to obtain its embedding representation. Lookup matrix can be initialized using pretrained word embedding vectors (Bojanowski et al.; Pennington et al., 2014). In our work, the pre-trained vector representations provided by Glove (Pennington et al., 2014) are used. It captures syntactic and semantic relations among the words. The embedding of each word *w* is then given as an input to the CNN to learn the text representation. Equation 1 shows the sequence of words present in sentence where w_i is i_{th} word present in the sentence and L is length of sentence.

$$W_i = w_i^1, w_i^2, ...w_i^j, ...w_i^L \tag{1}$$

3.1.2 Convolutional Neural Network (CNN)

The convolutional neural network automatically learns the features with the help of convolutional filters. Convolutional filters capture the semantic and syntactic features of a given sentence. CNN has been used in a wide variety of tasks (Rios and Kavuluru, 2015), (Kim, 2014). The CNN consists of convolutional layers. Convolutional layers are followed by non-linear layers that contain the Relu activation function, followed by the pooling layers. For our task, we use 3 convolutional layers. The three convolutional layers contain 128 filters of sizes 2, 3, and 4 each. Word embedding vectors of a sentence are given as input to CNN to learn the n-gram features. Equation 2 shows the CNN output for a sentence after convolving different size filters on the word embedding matrix of the sentence.

$$H_i = h_i^1, h_i^2, ...h_i^j, ...h_i^L \tag{2}$$

Where H$_i$ represents the final feature vector for a sentence.

3.1.3 Attention for text

In NLP related tasks, some words in the sentence are more important for the task compared to the other words in the same sentence. To capture this phenomena, attention model for the text has been proven beneficial for many NLP related tasks i.e., text summarization, machine translation (Luong et al., 2015; Bahdanau et al., 2014), textual sentiment analysis(Corpora, 2000; Chen et al., 2016), etc. Attention models calculate the attention score α_i^j which lies in the range of 0 and 1. Attention score is assigned to feature representation of each w_i^j i.e., h_i^j based on its importance, which is calculated as follows

$$\alpha_i^j = \frac{exp(p_i^j)}{\sum_{j=1}^{L} exp(p_i^j)} \tag{3}$$

Where,

$$p_i^j = \theta(Mh_i^j + b) \tag{4}$$

θ refers to nonlinear activation function ($tanh$). The weight matrix M and bias b are the network parameters and h_i^j is the feature representation of word w_i^j (CNN output). α is calculated for all the words in the sentence. The attended text feature vector can be calculated as a weighted sum of all the words present in a sentence, as shown in Equation 5.

$$Z_i^k = \sum_{1<=j<=L} \alpha_i^j h_i^j \tag{5}$$

Attention process for text is illustrated Figure 3.

3.2 Visual Features

The image with size 224*224 is used as the input to the pre-trained model VGG-19 to extract features of the image. We use the output of *conv5*4* layer of VGG-19 as the region features which consist of 196 regions, and each region is represented in 512 dimensions. Thus region features are having dimensions of (196*512). The output of VGG-19 is further passed to a dense layer that has 250 hidden

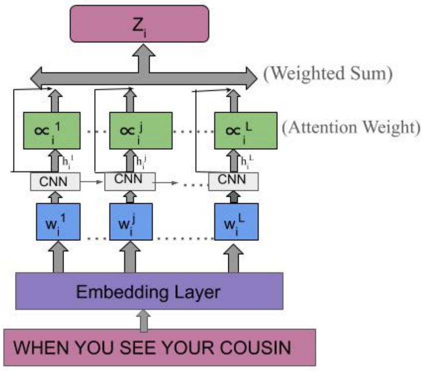

Figure 3: Attention for text

neurons. The output of this dense layer is passed to the attention layer to find out the important regions for classification.

3.2.1 Attention for Image

Image attention has been proven to be beneficial for many vision-related tasks (Zhou et al., 2019). We apply the attention over the image regions (output of dense layer) to find out the most important regions. Equation 6 shows the sequence of region maps for i$_{th}$ image.

$$R_i = r_i^1, r_i^2, ...r_i^j, ...r_i^C \tag{6}$$

where, C is the number of regions and each region is now represented in *D* (250) dimension.

Attention score β_i^j is calculated for each region, signifying the region importance. It lies in the range between 0 and 1. If a region is more important for classification, then value of β_i^j will be more. Attention score β_i^j is calculated as shown in Equation 7

$$\beta_i^j = \frac{exp(p_i^j)}{\sum_{j=1}^{D} exp(p_i^j)} \tag{7}$$

Where,

$$p_i^j = \phi(Mr_i^j + b) \tag{8}$$

The weight matrix M and bias b are the parameters to be learned. ϕ is a nonlinear activation function and we use $tanh$ function. Finally, image features are calculated as weighted sum over all regions as shown in Equation 9.

$$U_i^k = \sum_{1<=j<=L} \beta_i^j r_i^j \tag{9}$$

The architecture of the attention for image is illustrated in Figure 4.

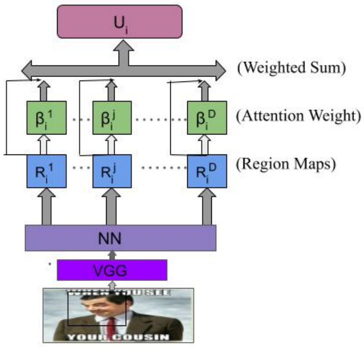

Figure 4: Attention for image

Type	Positive	Negative	Neutral	Total
Train	4155	629	2218	7002
Test	1109	173	593	1875

Table 1: SemEval dataset

3.2.2 Fusion of Text and Image features

Finally, the attended image features vector and text features vector are passed to a fully connected layer containing hidden neurons. This layer tries to find out the relation between image and text features and finally combines both.

3.3 Output Layer

The output of dense layer, i.e., the combined feature vector of image and text is finally passed to the output layer, which contains softmax as an activation function. The output layer maps the combined feature vector to a probability score. This probability score helps to classify the tweet into one of its predefined categories.

4 Dataset and Experiment

In this section, we discuss about the dataset used for the experiment, data collection, data annotation, and experimental details.

4.1 Dataset

We use the SemEval-2020 task 8 dataset[1] for sentiment analysis of memes. This dataset contains 8877 memes annotated for 3 classes, *viz.,* positive, negative, and neutral. The dataset is divided into 2 parts, training, and test. The distribution of the dataset is shown in Table 1.

As shown in Table 1, the data set is highly imbalanced. There are very less number of instances in

[1] https://competitions.codalab.org/competitions/20629

Dataset	Positive	Negative	Neutral
9277	4109	2375	2811

Table 2: Class-wise distribution

Type	Positive	Negative	Neutral
Train	2557	1907	1875
Development	443	295	343
Test	1109	173	593

Table 3: Data statistics

the negative class. So we crawl some data to make it balanced.

4.2 Data Collection and Annotation

We collect the memes from Reddit. After data collection, we extract the text written on memes using a python library known as python-tesseract. Python-tesseract is an optical character recognition (OCR) tool for python. After extracting text with Python-tesseract, we manually verify the output to correct the wrong instances. Then we conduct manual annotations for memes. Three annotators with post-graduate level knowledge in English are employed for annotations. Annotators are asked to write the overall polarity of the tweet for 3 classes, *viz.,* neutral, negative, and positive. Initially, to build an understanding of the class labels, we provide some tweets to the annotators with gold labels.

We added the newly annotated instances for negative class to the training part of the SemEval dataset. After merging, we divide it into two parts, train and validation. The test set is the same as provided in the original SemEval dataset. We downsample the positive class data for the balancing purpose. Class wise distribution of combined dataset is shown in Table 2. Train-dev-test distribution is shown in Table 3. [2]

4.3 Data Pre-processing

We perform the following steps to pre-process the text written memes.

- Convert all the characters of text into lowercase.
- Tokenize the sentence into sequence of words.
- Sentences with length less than maxlen are padded with zeros and greater than length maxlen are truncated.

4.4 Experimental Setup

We implement our model using python based Keras library [3]. We train our system for the 50 epochs and we save the checkpoints after every epoch to find the best performing model. We set the maximum sentence length to 80. We use batch size of 16 and ReLu activation function at the hidden layers of the network. We use optimizer Adam (Kingma and Ba, 2014) to optimize the weights of the network with a learning rate of 0.001. We use the softmax activation function at the last layer and categorical cross-entropy as the loss function. To prevent overfitting (Hawkins, 2004), dropout (Srivastava et al., 2014) of rate 0.5 is used at hidden layers. To find optimal values of hyper-parameters, we use the grid search.

4.5 Baseline models

We define the following baseline models.

- Baseline 1 (Textual model): Baseline 1 uses only textual information (text written on meme) for classification. We use the textual component without attention from the architecture shown in Figure 2.
- Baseline 2 (Visual model): Baseline 2 uses only visual information for classification. We use the image component without attention from the architecture shown in Figure 2.
- Baseline 3 (Textual model with attention): Baseline 3 uses only textual information and applies attention to the output of CNN to extract the most important words for classification as shown in Figure 2.
- Baseline 4 (Visual model with attention): Baseline 4 uses only visual information by extracting region features from VGG and apply attention over the regions to find out relevant regions for classification. Architecture is shown in Figure 2.
- Baseline 5 (Visual and textual without attention): Baseline 5 uses both textual as well as visual information for classification. We apply the architecture, as shown in Figure 2 by removing the attention layer from both image and text where image and textual features

[2] The annotated dataset is available from the authors upon request.

[3] https://keras.io/

Model	Macro F1 Score	Accuracy
Textual Model	31.42 %	43.25%
Visual Model	32.07%	42.16%
Textual Model With Attention	33.17%	44.34%
Visual Model With Attention	33.01%	42.98%
Visual And Textual Without Attention	33.22%	47.72%
SemEval Baseline	21.76	-
Proposed	34.23%	50.02%

Table 4: Evaluation results of different modalities

are concatenated and then passed to the fully connected layer.

- Baseline 6: Baseline 6 is provided by SemEval-2020 Task 8 which utilizes textual and image features.

- Final model: Our proposed model uses textual and visual information for classification by applying attention to text as well as image. Figure 2 describes our final architecture.

5 Evaluation Results

In this section, we discuss the detailed experimental results. We use accuracy and macro F1 score to evaluate the performance of our system. Table 4 shows the performance of our proposed model and comparison to the baseline models. The textual model (Baseline 1) yields the macro F1 and accuracy of 31.42% and 43.25%, respectively. Visual model (baseline 2) yields the macro F1 and accuracy of 32.07% and 42.16%, respectively. The model using only textual features with attention component (baseline 3) yields the macro F1 and accuracy of 33.17% and 44.34%, respectively. The visual model with attention (i.e., Baseline 4) yields macro F1 and accuracy of 33.01% and 42.98% ,respectively. Concatenation of textual and visual features (Baseline 5) without applying attention to image and textual features yields the macro F1 and accuracy of 33.22% and 47.72%, respectively. Reported macro F1 of SemEval baseline (Baseline 6) is 21.76%. Our proposed model obtains the macro F1 and accuracy of 34.23% and 50.02%, respectively. Our proposed system outperforms the other baselines, which indicates that multi-modal information actually helps to improve the effectiveness of the system. All the reported results are statistically significant as we have performed pairwise Welch's t-test (Welch, 1947) at 5% significant level.

Class	Negative	Neutral	Positive
Negative	17	43	113
Neutral	63	139	391
Positive	116	211	782

Table 5: Confusion matrix

5.1 Error Analysis

In this section, we present a detailed error analysis. Table 6 shows the example cases to establish the need for image as well textual model for sentiment classification of memes.

Column name is same as image name.

- Columns *a* shows the case where the textual model (Baseline 1) performs misclassification, but the visual model (Baseline 2) correctly predict the class.

- Column *b* shows the case where the visual model (Baseline 2) performs misclassification, but the visual model with attention model (Baseline 4) predict it correctly.

- Column *c* describes the case where the visual model (Baseline 2) is wrong, but the textual model (Baseline 1) performs correct classification.

- Column *d* shows the case where the textual model (Baseline 1) performs misclassification, but the textual model with attention (Baseline 3) performs correct classification.

- Column *e* shows the case when all the above-mentioned models perform misclassification, but the model which combine image and text through dense layer (Baseline 5) performs correct classification.

- Column *f* shows the case where all the baseline models fail, but our proposed model performs correct classification.

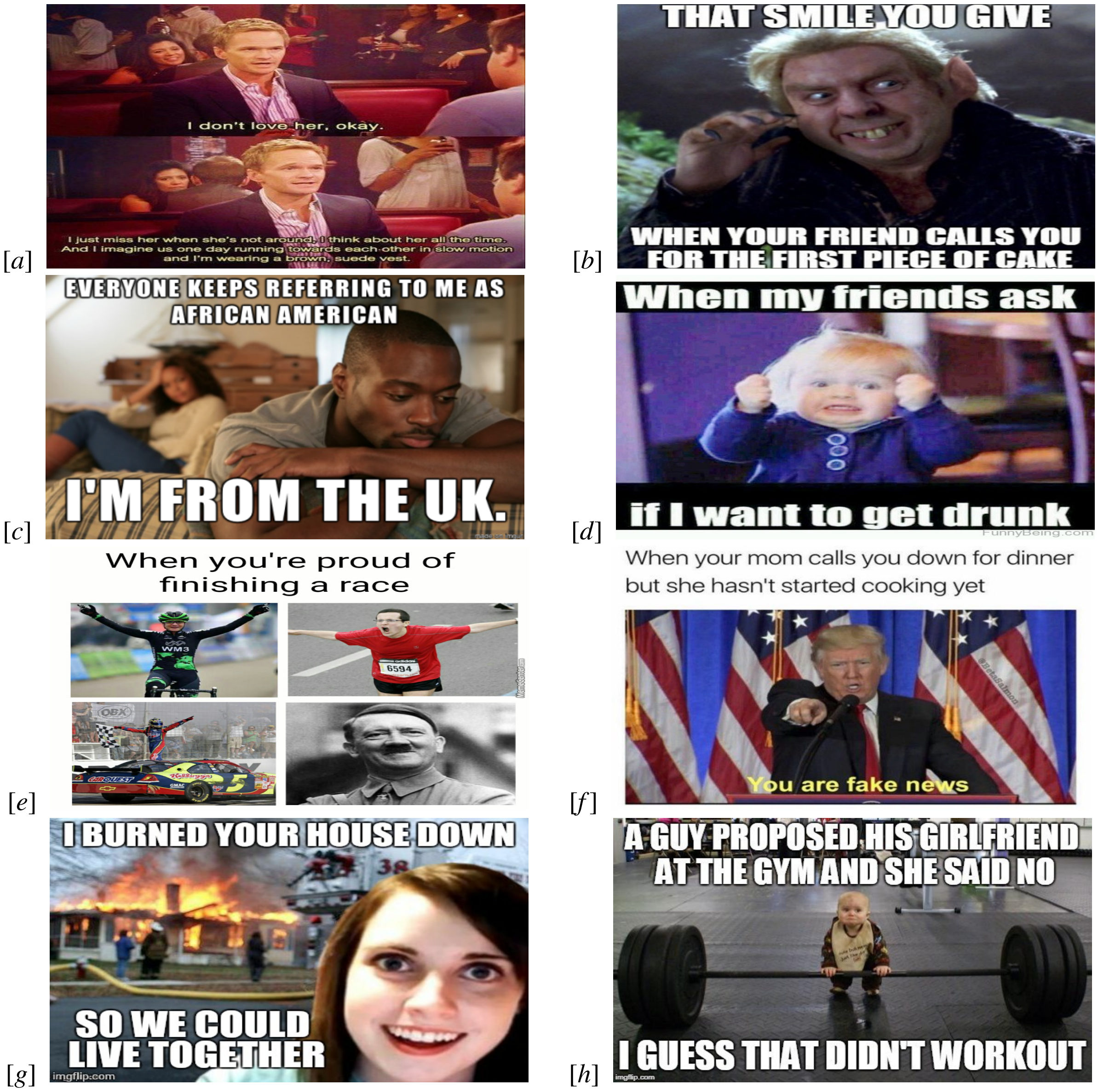
[a]
I don't love her, okay.
I just miss her when she's not around, I think about her all the time. And I imagine us one day running towards each-other in slow motion and I'm wearing a brown, suede vest.
[b]
THAT SMILE YOU GIVE
WHEN YOUR FRIEND CALLS YOU FOR THE FIRST PIECE OF CAKE
[c]
EVERYONE KEEPS REFERRING TO ME AS AFRICAN AMERICAN
I'M FROM THE UK.
[d]
When my friends ask
if I want to get drunk
FunnyBeing.com
[e]
When you're proud of finishing a race
[f]
When your mom calls you down for dinner but she hasn't started cooking yet
You are fake news
[g]
I BURNED YOUR HOUSE DOWN
SO WE COULD LIVE TOGETHER
imgflip.com
[h]
A GUY PROPOSED HIS GIRLFRIEND AT THE GYM AND SHE SAID NO
I GUESS THAT DIDN'T WORKOUT
imgflip.com

Figure 5: Qualitative analysis

Model	**a**	**b**	**c**	**d**	**e**	**f**
Actual	Neutral	Positive	Positive	Neutral	Neutral	Positive
Baseline 1	Negative	Negative	Positive	Positive	Positive	Negative
Baseline 2	Neutral	Neutral	Neutral	Positive	Positive	Negative
Baseline 3	Negative	Negative	positive	Neutral	Positive	Negative
Baseline 4	Neutral	Positive	Neutral	Positive	Negative	Neutral
Baseline 5	Negative	Neutral	Neutral	Positive	Neutral	Neutral
Proposed	Neutral	Positive	Positive	Neutral	Neutral	Positive

Table 6: Predictions of different models

Model	**g**	**h**
Actual	Negative	Neutral
Proposed Methodology	Neutral	Negative

Table 7: Qualitative error analysis of proposed model

These cases establish the effectiveness of our proposed approach. Further, we analyzed the output of our proposed model, both quantitatively and qualitatively. Confusion matrix is shown in the Table 5. It shows that the majority of negative class memes and neutral class got confused with positive class, and the majority of positive class memes got confused with neutral class. In Table 7, we show the cases where our proposed model performs misclassification. The column name is same as the image name. For image g, the proposed model misclassifies it to the neutral class, but the actual label is negative. A possible reason could be the presence of a happy face in the image. For image h, the predicted sentiment is negative, but the actual label is neutral. A possible reason could be the presence of a sad face in the image.

6 Conclusion

In this paper, we have proposed a multi-modal framework for meme sentiment classification by utilizing textual and visual information of memes. We use the SemEval-2020 task data and also annotated our own dataset to make this dataset balanced. We found that only textual information or only visual information is not sufficient to analyze a meme's sentiment. Our proposed framework utilizes textual and visual features and finally fuses both the information through a fully connected layer. Our proposed framework achieved the macro F1 and accuracy of 34.23% and 50.02%, respectively. Our proposed framework increases the accuracy by 6.77% and 7.86% compared to only textual and visual features, respectively. In the future, we are planning to explore other fusion methods to incorporate textual and visual features. We would also explore contextual embeddings for the text part of meme classifications.

References

Mamta ., Asif Ekbal, Pushpak Bhattacharyya, Shikha Srivastava, Alka Kumar, and Tista Saha. 2020. Multi-domain tweet corpora for sentiment analysis: Resource creation and evaluation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5046–5054, Marseille, France. European Language Resources Association.

Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca J Passonneau. 2011. Sentiment analysis of twitter data. In *Proceedings of the workshop on language in social media (LSM 2011)*, pages 30–38.

Md Shad Akhtar, Asif Ekbal, and Erik Cambria. 2020. How intense are you? predicting intensities of emotions and sentiments using stacked ensemble. *IEEE Computational Intelligence Magazine*, 15(1):64–75.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Akshat Bakliwal, Jennifer Foster, Jennifer van der Puil, Ron O'Brien, Lamia Tounsi, and Mark Hughes. 2013. Sentiment analysis of political tweets: Towards an accurate classifier. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:P10008.

Hongshen Chen, Yue Zhang, and Qun Liu. 2016. Neural network for heterogeneous annotations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 731–741.

Very Large Corpora. 2000. Empirical methods in natural language processing.

Deepak Chowdary Edara, Lakshmi Prasanna Vanukuri, Venkatramaphanikumar Sistla, and Venkata Krishna Kishore Kolli. 2019. Sentiment analysis and text categorization of cancer medical records with lstm. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–17.

Manoochehr Ghiassi and S Lee. 2018. A domain transferable lexicon set for twitter sentiment analysis using a supervised machine learning approach. *Expert Systems with Applications*, 106:197–216.

Douglas M Hawkins. 2004. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12.

Tao Jiang, Jiahai Wang, Zhiyue Liu, and Yingbiao Ling. 2020. Fusion-extraction network for multimodal sentiment analysis. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 785–797. Springer.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Akshi Kumar and Arunima Jaiswal. 2017. Image sentiment analysis using convolutional neural network. In *International Conference on Intelligent Systems Design and Applications*, pages 464–473. Springer.

Wenkuan Li, Peiyu Liu, Qiuyue Zhang, and Wenfeng Liu. 2019. An improved approach for text sentiment classification based on a deep neural network via a sentiment attention mechanism. *Future Internet*, 11(4):96.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

A Man, Yuanyuan Pu, Dan Xu, Wenhua Qian, Zhengpeng Zhao, and Qiuxia Yang. 2019. Multi-feature fusion for multimodal attentive sentiment analysis. In *MMAsia*, pages 43–1.

Walaa Medhat, Ahmed Hassan, and Hoda Korashy. 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, 5(4):1093–1113.

Igor Mozetič, Miha Grčar, and Jasmina Smailović. 2016. Multilingual twitter sentiment classification: The role of human annotators. *PloS one*, 11(5):e0155036.

Gorti Satyanarayana Murty and Shanmukha Rao Allu. Text based sentiment analysis using lstm.

Alessandro Ortis, Giovanni Maria Farinella, Giovanni Torrisi, and Sebastiano Battiato. 2020. Exploiting objective text description of images for visual sentiment analysis. *Multimedia Tools and Applications*, pages 1–24.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 115–124. Association for Computational Linguistics.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

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

Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Erik Cambria, Alexander Gelbukh, and Amir Hussain. 2018. Multimodal sentiment analysis: Addressing key issues and setting up the baselines. *IEEE Intelligent Systems*, 33(6):17–25.

Anthony Rios and Ramakanth Kavuluru. 2015. Convolutional neural networks for biomedical text classification: application in indexing biomedical articles. In *Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 258–267.

Chhavi Sharma, Deepesh Bhageria, William Paka, Scott, Srinivas P Y K L, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Björn Gambäck. 2020. SemEval-2020 Task 8: Memotion Analysis-The Visuo-Lingual Metaphor! In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020)*, Barcelona, Spain. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Bernard L Welch. 1947. The generalization ofstudent's' problem when several different population variances are involved. *Biometrika*, 34(1/2):28–35.

Guixian Xu, Yueting Meng, Xiaoyu Qiu, Ziheng Yu, and Xu Wu. 2019. Sentiment analysis of comment texts based on bilstm. *Ieee Access*, 7:51522–51532.

Quanzeng You, Liangliang Cao, Hailin Jin, and Jiebo Luo. 2016. Robust visual-textual sentiment analysis: When attention meets tree-structured recursive neural networks. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 1008–1017.

Jinfei Zhou, Yaping Zhu, and Hong Pan. 2019. Image caption based on visual attention mechanism. In *Proceedings of the 2019 International Conference on Image, Video and Signal Processing*, pages 28–32.