IIITG-ADBU at SemEval-2020 Task 8: A Multimodal Approach to Detect Offensive, Sarcastic and Humorous Memes

Arup Baruah[1], Kaushik Amar Das[1], Ferdous Ahmed Barbhuiya[1], and Kuntal Dey[2]*
[1]IIIT Guwahati, India
[2]Accenture Technology Labs, Bangalore
arup.baruah@gmail.com, kaushikamardas@gmail.com,
ferdous@iiitg.ac.in, kuntal.dey@accenture.com

Abstract

In this paper, we present a multimodal architecture to determine the emotion expressed in a meme. This architecture utilizes both textual and visual information present in a meme. To extract image features we experimented with pre-trained VGG-16 and Inception-V3 classifiers and to extract text features we used LSTM and BERT classifiers. Both FastText and GloVe embeddings were experimented with for the LSTM classifier. The best F1 scores our classifier obtained on the official analysis results are 0.3309, 0.4752, and 0.2897 for Task A, B, and C respectively in the Memotion Analysis task (Task 8) organized as part of International Workshop on Semantic Evaluation 2020 (SemEval 2020). In our study, we found that combining both textual and visual information expressed in a meme improves the performance of the classifier as opposed to using standalone classifiers that use only text or visual data.

1 Introduction

The word *meme* was first coined by Richard Dawkins to refer *"an idea, behavior, or style that spreads from person to person within a culture"* (Dawkins, 1976). Internet memes are the result of deliberate modification of an original idea using one's own creativity [1]. It is a type of meme that spreads via the Internet. Memes have become a new way of communication on the Internet. People mostly use it as a way to share jokes. But there are also other classes of memes that are offensive in nature. They spread hatred and racism. This second class of memes is harmful to society. They need to be detected and removed from the Internet. But the scale involved makes manual monitoring difficult.

Automated systems can help detect offensive memes. However, the automatic classification of memes has a lot of challenges. It is not textual data or images in isolation. The information provided by both the image and the text needs to utilized to correctly understand the emotion expressed by the meme. The text itself may be very short having only a few words. But the image provides a context and the expressions exhibited by the images supplement the textual information and it is the combinations of both text and image that enable one to understand the message conveyed by the meme.

Memotion Analysis (Task 8) organized as part of the International Workshop on Semantic Evaluation 2020 (SemEval 2020) required detecting the emotion expressed by a given meme (Sharma et al., 2020). This task consisted of three subtasks: Subtask A - Detect the sentiment of a given meme. It was a three-way classification problem with the labels being *positive*, *negative*, or *neutral*, Subtask B - This was a multi-label classification problem where it was required to determine if a given meme is *humorous*, *motivational*, *offensive* or *sarcastic*. A meme can belong to multiple class, Subtask C: This was a multi-class multi-label classification problem where each class mentioned in subtask B is further sub-divided into four levels. For example, the humorous class is sub-divided as *not funny*, *funny*, *very funny*, and *hilarious*.

*This work was done when the author was affiliated with IBM Research India, New Delhi

[1]https://www.wired.co.uk/article/richard-dawkins-memes

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 885–890
Barcelona, Spain (Online), December 12, 2020.

We participated in all the three subtasks. We used the pre-trained convolutional neural networks VGG-16 (Simonyan and Zisserman, 2015) and Inception-V3 (Szegedy et al., 2016) as our image classifiers. For processing text, we used LSTM (Hochreiter and Schmidhuber, 1997) and the pre-trained BERT classifier (Devlin et al., 2019). The rest of the paper is structured as follows: Section 2 discusses the related works that have used multimodal approaches for detecting emotions expressed by memes, Section 3 describes the dataset used in this shared task, Section 3 discusses the methodology and the architecture of the classifier used by us, Section 5 describes the different experiments we have performed in this task including the questions we are trying to answer through the experiments, and Section 6 discusses the results our classifier obtained on the development set and the test set.

2 Related Work

Extracting the text from the meme is an important step in the multimodal processing of meme. Borisyuk et al. (2018) describes Facebook's OCR system for extracting text from images in realtime. Bruni et al. (2014) talks about integrating both text and image based distributional information to create multimodal distributional semantic vectors.

Kumar et al. (2020) used a multimodal approach to determine the sentiment of a meme. Google Lens was used to separate the text from the meme. The sentiment expressed by the text was determined using the combination of a convolutional neural network and the sentiment scores of words from the VADER sentiment lexicon. When determining the sentiment scores of the words, the context in which the word appears was taken into consideration. The image was processed using an SVM classifier trained with Bag of Visual Words features. A Boolean decision system was then used to combine the text and image scores to make the final classification. Hu and Flaxman (2018) also used a multimodal approach to determine the emotion expressed in the Tumblr posts. Inception and LSTM were used to process the image and text respectively. Sabat et al. (2019) used a multimodal approach to detect hate memes. BERT was used to process the text and VGG-16 was used to process the image. Both the information was concatenated and the final classification was done using an MLP classifier.

3 Data Set

The train and test data set provided as part of this task consisted of 6,992 and 1,878 memes respectively. The memes in the train data set were annotated for the following categories: sentiment, humour, motivational, offensive, and sarcasm. Table 1 to 5 shows the statistics of the labels used for each category. The labels used for the *sentiment* category were very_positive (L1), positive (L2), neutral (L3), negative (L4), and very_negative (L5). The labels used for the *humour* category were hilarious (L1), not_funny (L2), very_funny (L3), and funny (L4). The labels used for the *motivational* category were not_motivational (L1), and motivational (L2). The labels used for the *offensive* category were not_offensive (L1), very_offensive (L2), slight (L3), and hateful_offensive (L4). The labels used for the *sarcasm* category were general (L1), not_sarcastic (L2), twisted_meaning (L3), and very_twisted (L4). As can be seen from the tables, the data set was, in general, imbalanced.

4 Methodology

In this multimodal approach of detecting the emotion expressed by a meme, we combined both the visual and textual information to perform the classification. The pre-trained image classifiers VGG-16 (Simonyan and Zisserman, 2015) and Inception-v3 (Szegedy et al., 2016) were used to process the image. The textual data was processed using LSTM (Hochreiter and Schmidhuber, 1997) and the pre-trained BERT (Devlin et al., 2019) classifier.

VGG-16 is a convolutional neural network (CNN) based architecture for image classification. It was ranked second in the ImageNet image classification task in 2014 (ILSVRC 2014). It has 13 convolutional layers, 5 Max pooling layers, and 3 Dense layers. Out of these 21 layers, only 16 are weight layers. Inception-V3 is the third version of Google's Inception network. It too is a CNN based architecture. It was the 1st runner-up in the ImageNet image classification task in 2015. It consists of 42 layers.

L1	3,507 (50%)
L2	1,544 (22%)
L3	1,547 (22%)
L4	394 (6%)
Total	6,992

Table 1: Sarcasm

L1	2,713 (39%)
L2	1,466 (21%)
L3	2,592 (37%)
L4	221 (3%)
Total	6,992

Table 2: Offensive

L1	1,033 (15%)
L2	3,127 (45%)
L3	2,201 (31%)
L4	480 (7%)
L5	151 (2%)
Total	6,992

Table 3: Sentiment

L1	651 (9%)
L2	1,651 (24%)
L3	2,238 (32%)
L4	2,452 (35%)
Total	6,992

Table 4: Humour

L1	4,525 (65%)
L2	2,467 (35%)
Total	6,992

Table 5: Motivational

LSTM is a type of recurrent neural network (RNN). It handles the vanishing and the exploding gradient problem through the use of an input gate, output gate, and forget gate. It is thus able to handle long range dependencies. BERT is a bi-directional model based on the transformer architecture. The transformer architecture is an architecture based solely on attention mechanism (Vaswani et al., 2017).

4.1 The Architecture of our Classifier

Figure 1 shows the architecture of the classifier used by us. This architecture is inspired by a work performed for automatic image captioning [2]. As shown in the figure, the text first needs to be extracted from the meme. In this SemEval task, the text was already extracted from the meme by the organizers of the task. The extracted text was provided in the data set along with the memes. The text and the image are then processed independently by our classifier. We used the pre-trained convolutional neural network based classifiers, VGG-16, and Inception-v3, to process the image. For both these pre-trained classifiers, the output of the second-last layer is used as the representation of the image. For VGG-16, the size of the vector produced by the second-last layer is 4096 and in the case of Inception-v3 the size of this vector is 2048. These vectors were then fed to a Dense layer having 256 units.

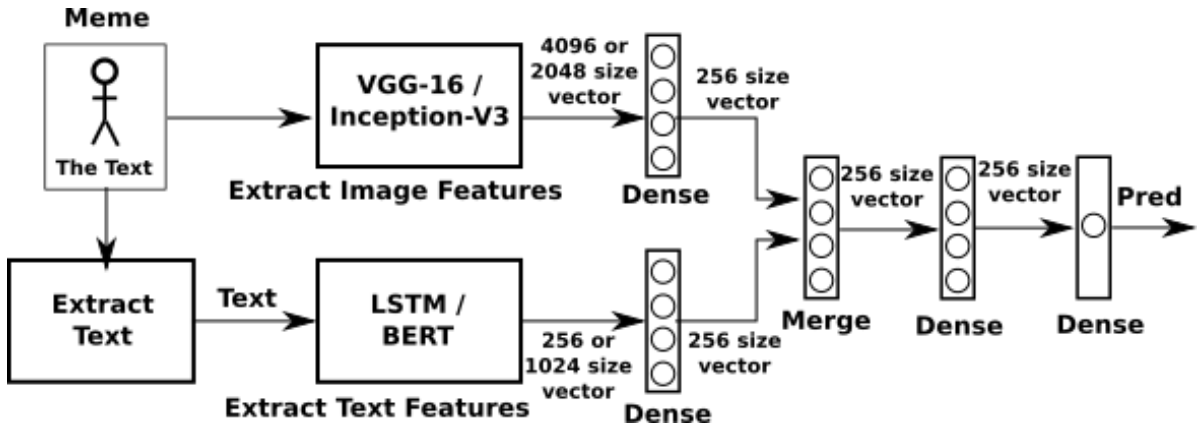

Figure 1: Architecture of our classifier

The text was processed using LSTM and the pre-trained BERT classifier. For LSTM, the words in the text were represented using pre-trained GloVe [3] and fastText [4] embeddings. An LSTM of size 256 units was used in our experiment. The hidden states of the intermediate time steps were not used. Only the hidden state of the final step was used as the representation of the text. This vector was merged with the

[2]https://towardsdatascience.com/image-captioning-with-keras-teaching-computers-to-describe-pictures-c88a46a311b8
[3]https://nlp.stanford.edu/projects/glove/
[4]https://fasttext.cc/docs/en/english-vectors.html

image vector directly. Thus, the intermediate Dense layer that is shown in the diagram is not used in the case of LSTM. The intermediate Dense layer is used only in the case of BERT. For BERT, we used the uncased large version of it [5]. We used the 1024 dimensional vector produced by the Extract layer of BERT as the representation of the text. This vector was then fed to a Dense layer having 256 units. The maximum sequence lengths of 30 and 50 were used for LSTM and BERT respectively.

The 256-dimensional vectors produced for the image and text were combined together in a Merge layer. The output of the merge layer is then fed to a Dense layer having 256 units. The output of this Dense layer is then fed to a classification layer. The classification layer consisted of three units for subtask A, a single unit for subtask B and four units for subtask C. During training, we fine-tuned only the Dense and LSTM layers. The pre-trained layers of VGG-16, Inception-v3, and BERT were not retrained. The *adam* optimizer with the default learning rate of 0.001 was used to train the classifier. The loss function *categorical crossentropy* was used for subtask A and C and *binary crossentropy* was used for subtask B. The *relu* activation function was used for all the Dense layers except the final classification layer. The final classification layer used the *softmax* activation function for subtask A and C, and the *sigmoid* activation function for subtask B.

Subtask B and C were multi-label classification problems. For these subtasks, we used the *binary relevance* approach for classification. We trained a separate classifier for each class (humorous, motivational, offensive, and sarcasm). For subtask B, the classifiers were binary classifiers and for subtask C the classifiers were multi-class classifiers.

5 Experiments

We performed the following nine types of experiments for this task: (1) Only VGG-16, (2) Only Inception-v3, (3) Only LSTM with fastText embeddings, (4) Only BERT, (5) VGG-16 for processing image and LSTM with GloVe embeddings for processing text, (6) VGG-16 for processing image and LSTM with fastText embeddings for processing text, (7) Inception-v3 for processing image and LSTM with GloVe embeddings for processing text, (8) Inception-v3 for processing image and LSTM with fastText embeddings for processing text, and (9) Inception-v3 for processing image and BERT for processing text. The first four experiments used standalone classifiers (meaning classifiers that used only textual or visual data). The next five experiments used multimodal classifiers (classifiers that combined both textual and visual data). The reason for conducting these experiments was to find an answer to the following questions:

1. Does combining both textual and visual data improve performance compared to using only textual or visual data?
2. Among standalone classifiers, is image based classifier better than text based classifier?
3. Among multimodal classifiers, which combination provides the best performance?

6 Results

Table 6 shows the results obtained by our classifiers on the development set. The development set was created by doing a stratified split on the train data set. 20% of the train data set was used as the development set. The table shows the F1 score obtained by the classifiers for the subtasks Task A (A), Task B Humour (B-H), Task B Motivational (B-M), Task B Offensive (B-O), Task B Sarcasm (B-S), Task C Humour (C-H), Task C Motivational (C-M), Task C Offensive (C-O), and Task C Sarcasm (C-S). The *Motivational* class had only two unique labels (*motivational*, *not-motivational*) as opposed to the other three classes (*Humour*, *Offensive*, and *Sarcasm*) which had four unique labels. Thus, *Task B Motivational* and *Task C Motivational* reduced to be the same task. For this reason, only one column is shown in the table for these two subtasks.

As can be seen from the table, except for *Task B Offensive* subtask, all the other subtasks have benefited from combining both the image and text data. In the case of *Task B Offensive* subtask, the best F1 score

[5] https://github.com/google-research/bert

was obtained when using only the fastText based LSTM classifier. Compared to the best performing standalone classifier, the best performing multimodal classifiers obtained gain of 1.44%, 4.41%, 3.87%, 4.91%, 2.18%, 2.18%, 0.89%, and 1.62% in the F1 score for subtask A, B-H, B-M, B-S, C-H, C-O, and C-S respectively. We can thus say that detecting emotion in memes benefits from combining both textual and visual data as compared to using standalone classifiers.

Among the standalone classifiers, the classifier that used only visual data performed better for subtasks A, B-M, C-O, and C-S. Whereas, the classifier that used only text data performed better in subtasks B-H, B-O, and C-H. Thus, there was no clear winner among the standalone classifiers.

Among the multimodal classifiers, VGG16+LSTM(FastText) performed the best for subtasks A, B-O, and C-H; InceptionV3+LSTM(FastText) performed the best for subtasks B-H, and B-M; InceptionV3+LSTM(GloVe) performed the best for subtasks B-S and C-S; and VGG16+LSTM(GloVe) performed the best for subtask C-O.

Table 7 shows the official results on the test set. The five runs that were submitted for analysis are InceptionV3+LSTM(GloVe) (Run1), VGG16+LSTM(GloVe) (Run2), VGG16+LSTM(FastText) (Run3), InceptionV3+LSTM(FastText) (Run4), and InceptionV3+BERT (Run5). As can be seen from table 7, InceptionV3+BERT which did not perform well on the development set, produced the best F1 scores for Task A and Task B. VGG16+LSTM(FastText) produced the best result for Task C. The predictions from InceptionV3+LSTM(GloVe) was our final submission for Task A and this submission was ranked 30. The predictions from InceptionV3+LSTM(FastText) were our final submission for Task B and C, and these submissions obtained the rank of 26 and 23 for the two tasks respectively.

System	**A**	**B-H**	**B-M, C-M**	**B-O**	**B-S**	**C-H**	**C-O**	**C-S**
VGG-16	0.2847	0.4420	0.4678	0.4124	0.4358	0.2198	0.2406	0.1910
Inception-v3	0.2487	0.4331	0.3930	0.3798	0.4379	0.1782	0.1847	0.1670
LSTM(FastText)	0.2670	0.4424	0.4479	**0.5030**	0.4379	0.2295	0.2112	0.1670
BERT	0.2532	0.4331	0.3930	0.4980	0.4379	0.2042	0.2138	0.1716
LSTM(GloVe) + VGG-16	0.2675	0.4428	0.4206	0.4785	0.4808	0.2337	**0.2495**	0.1938
LSTM(FastText) + VGG-16	**0.2991**	0.4420	0.4696	0.5027	0.4529	**0.2513**	0.2447	0.1854
LSTM(GloVe) + Inception-v3	0.2645	0.4451	0.4574	0.4498	**0.4870**	0.1985	0.2254	**0.2072**
LSTM(FastText) + Inception-v3	0.2485	**0.4865**	**0.5065**	0.4235	0.4653	0.2252	0.2215	0.1785
BERT + Inception-v3	0.2487	0.4416	0.4162	0.3958	0.4379	0.2119	0.1444	0.1670

Table 6: Macro F1 scores on development set

Task	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**	**Baseline**	**Best**	**Rank**
Task A	*0.3078*	0.2690	0.2759	0.2471	**0.3309**	0.2176	0.3546	30 (for Run 1)
Task B	0.4646	0.4619	0.4666	*0.4650*	**0.4752**	0.5002	0.5183	26 (for Run 4)
Task C	0.2616	0.2763	**0.2897**	*0.2850*	0.2839	0.3008	0.3224	23 (for Run 4)

Table 7: Macro F1 scores on test set (Official Scores)

7 Conclusion

Memes are becoming a very popular means of communication in social media. Some of these memes are offensive in nature and they are used to spread hatred. Thus, it is very important to develop systems to automatically determine the emotion expressed by memes. Developing such systems requires combining both textual and visual information present in the meme. In this paper, we presented a multimodal architecture that combines both textual and visual data to determine the emotion expressed by the meme. This classifier obtained F1 scores of 0.3309, 0.4752, and 0.2897 for task A, B, and C respectively. In our study, we found that combining both textual and visual data improves the performance of the classifiers than using standalone classifiers.

References

Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. 2018. Rosetta: Large scale system for text detection and recognition in images. In Yike Guo and Faisal Farooq, editors, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 71–79. ACM.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Intell. Res.*, 49:1–47.

Richard Dawkins. 1976. *The Selfish Gene*. Oxford University Press, Oxford, UK.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Anthony Hu and Seth R. Flaxman. 2018. Multimodal sentiment analysis to explore the structure of emotions. In Yike Guo and Faisal Farooq, editors, *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 350–358. ACM.

Akshi Kumar, Kathiravan Srinivasan, Wen-Huang Cheng, and Albert Y. Zomaya. 2020. Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data. *Inf. Process. Manag.*, 57(1).

Benet Oriol Sabat, Cristian Canton-Ferrer, and Xavier Giró-i-Nieto. 2019. Hate speech in pixels: Detection of offensive memes towards automatic moderation. *CoRR*, abs/1910.02334.

Chhavi Sharma, Deepesh Bhageria, William Scott Paka, 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, Dec. Association for Computational Linguistics.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan. N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.