KAFK at SemEval-2020 Task 8: Extracting Features From Pre-trained Neural Networks To Classify Internet Memes

Kaushik Amar Das [◇], **Arup Baruah** [◇], **Ferdous Ahmed Barbhuiya** [◇], **Kuntal Dey** [♡†]
[◇]IIIT Guwahati, India
[♡]Accenture Technology Labs, Bangalore
kaushikdas4@acm.org, arup.baruah@gmail.com,
ferdous@iiitg.ac.in, kuntal.dey@accenture.com

Abstract

This paper presents two approaches for the internet meme classification challenge of SemEval-2020 Task 8 by Team KAFK (cosec). The first approach uses both text and image features, while the second approach uses only the images. Error analysis of the two approaches shows that using only the images is more robust to the noise in the text on the memes. We utilize pre-trained DistilBERT and EfficientNet to extract features from the text and image of the memes respectively. Our classification systems obtained macro f1 score of 0.3286 for Task A and 0.5005 for Task B.

1 Introduction

Dawkins (1976) defines a "Meme" as the simplest cultural unit that can spread from one mind to another. These cultural units include but are not limited to, rumours, catch-phrases, melodies, current affairs, etc. Internet Memes is a phenomenon of content or concepts that spread rapidly among internet users (Bauckhage, 2011). Internet memes typically include an image superimposed with text or text placed in a white space over an image (Beskow et al., 2020).

Although Internet memes are often meant for humour or social commentary, it is also used as a means to propagate offensive, racist, sexist, and hateful content (Williams, 2016; Drakett et al., 2018). The growing popularity of memes makes it impossible for traditional human-flagging methods to scale. Hence, there is a growing need for automated methods.

There are many research works that have studied automated classification of text and images in social media such as (Schmidt and Wiegand, 2017; Felbo et al., 2017; Soleymani et al., 2017). However, there are few studies on the utility of these techniques on internet memes, even more so in regards to hate speech. Some of these include identification of the semantic correlation between the text and images of image-based memes (French, 2017). Meme-Hunter, a meme-detection system that classifies images on the internet as a meme or not (Beskow et al., 2020). A multi-modal system built using BERT (Devlin et al., 2019) and VGG-16 (Simonyan and Zisserman, 2014) for discriminating between hateful and non-hateful memes (Sabat et al., 2019).

The limitations of pre-existing techniques and the challenges in the domain of internet meme classification have not been widely explored. The Memotion Analysis workshop in SemEval-2020 has been organized with the goal of promoting research for the same. As participants in this workshop, we build two Deep Neural Network (DNN) based classification systems for the multi-modal classification of internet memes. Our models try to make use of both the text and the image feature of the memes. We also attempt at using only the image features and contrast it with the multi-modal approach of using text and images together. We have released our code online at https://github.com/cozek/memotion2020-code.

2 Task Description

The Memotion Analysis (Sharma et al., 2020) challenge is comprised of the three tasks listed below. The metric of evaluation for Task A is Macro F1. For Task B and C, the evaluation metric is the average of the

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

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1148–1154
Barcelona, Spain (Online), December 12, 2020.

Macro F1 for each of the sub-tasks. We participated in Task A and Task B.

- Task A: **Sentiment Classification**. Classify a meme as either positive, negative or neutral.
- Task B: **Humor Classification**. Identify the type of humour expressed by a meme. The categories of humour include sarcastic, humorous, offensive and motivational. A meme might belong to one or more of these categories or none at all, in event of which, we are to mark it as so.
- Task C: **Scales of Semantic Classes**. Quantify, on a scale of 0, 1, 2 and 3, the extent to which a particular effect is being expressed. The effects being humour, sarcasm, offensive and motivational. One exception here is that motivation is not scaled 0 to 3, instead it is limited to just 0 and 1.

3 Data

The dataset is compiled by Sharma et al. (2020). It contained 6992 memes labelled for each of the tasks mentioned in Section 2. Typically, memes are composed by overlaying text on images. Hence, the dataset aptly consists of images and their corresponding (OCR extracted) overlay-ed texts. The text is in the English language. The amount of samples for Task A and B are given in Table 2 and 3 respectively. On analysing the images, we find that there is a large variance in their size, i.e, their height and width. This can also be observed in the length of the text as well (see Table 1). This high variance in properties of both the text and the images suggests that there exists no standard format for memes. We suspect that this is further magnified when these memes hop through the internet.

Feature	**Mean**	**Max**	**Min**	**Std**
Image Width (px)	587.06	4961	100	256.83
Image Height (px)	546.50	5553	123	250.04
Text Length (char)	82.78	1026	2	50.38
Text Length (words)	14.66	189	1	8.97

Table 1: Dataset Feature Statistics

Label	*Positive*	*Negative*	*Neutral*
Count	4160	631	2201

Table 2: Task A Label Counts

Label		*Humor*	*Sarcasm*	*Offensive*	*Motivation*
Count	1	5341	5448	4279	2467
	0	1651	1544	2713	4525

Table 3: Task B Label Counts. The samples marked 1 are positive for their corresponding label

4 Classification System Overview

We built two DNN based classification systems for the classification tasks. The first model utilizes both text and images together while the second models use only the images. Both of these models follow the typical neural network fine-tuning procedure. This procedure includes using a pre-trained model to extract features which are then fed into a fully connected layer that makes the classification. The classification systems are described in details in Section 5.

For extracting features from the text and images we used pre-trained DistilRoBERTa (Sanh et al., 2019) and EfficientNets (Tan and Le, 2019) respectively. The main reason for selecting these models was their fast inference and training time and small size while having high benchmark scores in their

individual domains. Also, ImageNet pre-trained networks, such as EfficientNets, can be fine-tuned to attain performance improvements in other computer vision tasks (Kornblith et al., 2018). These base models are briefly described below.

4.1 DistilRoBERTa

At present, fine-tuning BERT models (Devlin et al., 2019) for various Natural Language Processing (NLP) tasks is a popular technique. RoBERTa (Liu et al., 2019b) is a robustly optimized version of the same. RoBERTa introduces a few modifications to the original. These consists of training the model for longer with more data in larger batches, removing the next sentence prediction pre-training task, using longer sequences for training and using a dynamic masking pattern. Studies show that RoBERTa models outperform or evenly match the performance of BERT models.

In this paper, we used a distilled version of RoBERTa, called DistilRoBERTa. Distil* is a class of compacted BERT-based (Devlin et al., 2019) transformers. These distilled models train faster while retaining up to 97% of the knowledge of their original models. This is done by using Knowledge Distillation (Hinton et al., 2015), which is a compression technique that trains a (compact) student model to reproduce the behaviour of a (larger) teacher model. The student model has an architecture identical to the teacher but the number of layers is reduced by a factor of 2. In addition, token-type embeddings are removed and training is done using a recipe similar to RoBERTa. In our experiments, we used the implementation by Wolf et al. (2019).

4.2 EfficientNets

EfficientNet (Tan and Le, 2019) is a family of Convolution Neural Networks(CNNs) that have achieved state-of-the-art performance on the ImageNet challenge (Russakovsky et al., 2014). This family of CNNs are generally around 8x smaller and 6x faster on inference compared to best existing ones such as SENet (Hu et al., 2017), GPipe (Huang et al., 2018). EfficientNet uses a compound scaling method to create the different models in the family that trade size for accuracy. Compound scaling uniformly scales a network's width, depth and resolution. In our experiments, we tested each of the seven publicly available pre-trained models in the family from b0 to b7. Here b0 is the baseline model which has been scaled up using compound scaling to obtain the models from b1 to b7.

5 Experimental setup

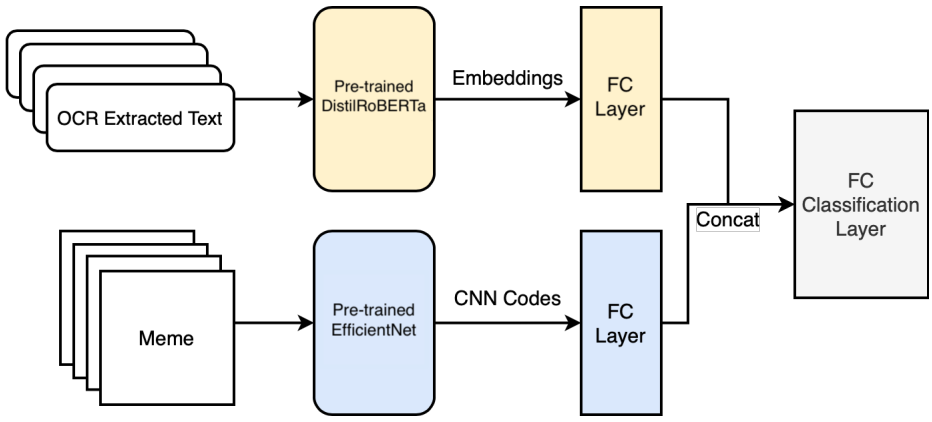

Figure 1: Classifier Model. System 1 uses both of the pre-trained models, whereas System 2 only makes use of the images. Fair Use

For training our models, we used 90% of the labelled data for training and remaining 10% as the dev set. The splits were made preserving the percentage of samples for each class. Both of the classification systems are built using the PyTorch (Paszke et al., 2017) framework. Since Task B was multi-label, we trained individual classifiers for each of the classes, i.e, 1 classifier for each of Humor, Sarcasm, Offensive and Motivational. The details of the two classification systems are given in Section 5.1 and Section 5.2.

5.1 System 1: DistilRoBERTa + EfficientNet

This system uses both EfficientNet[1] and DistilRoBERTa[2] together for classifying the memes. Firstly, we extract the embeddings of the texts from pre-trained DistilRoBERTa. These embeddings are then passed through a fully connected layer L^1 with c-neurons, where c is the number of classes. Secondly, we extract CNN codes from the final layer of pre-trained EfficientNet using the images, which are fed into another fully connected layer L^2 having c neurons. Finally, we concatenate the outputs (column-wise) of L^1 and L^2 and feed it to a final fully connected layer L^F which makes the classification. We use a memory-efficient version of $swish$ (Ramachandran et al., 2017) as the activation function for all of the aforementioned fully connected layers. Dropout (Hinton et al., 2012) of 0.4 , 0.1, and 0.1 is used on L^1, L^2 and L^F respectively.

5.2 System 2: EfficientNet

System 2 is largely similar to System 1, but only uses EfficientNet. We feed the images to the pre-trained EfficientNet Model to extract the CNN codes. These codes are then fed into a fully connected layer L^F having c neurons which makes the classification. We employ a dropout of 0.2 and the same memory-efficient $swish$ activation function on L^F. We tested the various scaled versions of EfficientNet for this model, b0 through b7. The best performing versions were chosen for inference.

5.3 System Training and Inference

On both of the systems described above [5.1, 5.2], we fine-tune the entire model, all the layers including the pre-trained model, using a small learning rate coupled with a cross-entropy loss function. The learning rate and batch size for each model are given in Table 4. We use $argmax$ on the final layer output to make the predictions. The models are optimized with Ranger optimizer which is a combination of the RAdam (Liu et al., 2019a) optimizer wrapped with LookAhead (Zhang et al., 2019). LookAhead's (k, α) are set to (5, 0.5). The systems are trained for 20 epochs on the training split with early-stopping. Early-stopping monitors the macro f1 score of the system on the dev split and halts the training if the score fails to improve for 4 epochs. We save the weights of the network with the highest macro-f1 score on the dev set and reload it for making inference on the test set.

	Model		*Batch Size*	*Learning Rate*
Task A	DistilRoBERTa + EfficientNet_b4		72	2e-4
	EfficientNet_b4		300	1e-4
Task B	Humor	EfficientNet_b1	128	1e-4
	Sarcasm	EfficientNet_b0	300	1e-4
	Offensive	EfficientNet_b1	128	1e-4
	Motivational	EfficientNet_b4	128	1e-4

Table 4: Hyperparameters for the fine-tuned models

6 Results

We report the performance of the best models from our experiments in Table 5. In Task A, the experiments show that using both the images and text together performs better than using images alone on the training set. The contrary is true for the dev set albeit by a tiny margin. Both of these models underfit the dataset as is evident from the poor f1 score. Since adding text features did not seem to improve the system and only added additional processing time, we decided to use the image-only EfficientNet models for the final test set for both Task A and Task B. In Task B, even though only the images are used for classification, the performance is better than that of Task A.

[1] https://github.com/lukemelas/EfficientNet-PyTorch
[2] https://huggingface.co/transformers/

The official rank and macro f1 score of our models on the test set are given in Table 6. The test set contained 1878 unlabelled memes. The EfficientNet models used for test set inference are same as the ones mentioned in Table 5.

	Model		*Macro F1*	
			Train Set	*Dev Set*
Task A	DistilRoBERTa + EfficientNet_b4		0.6288	0.3536
	EfficientNet_b4		0.3429	0.3557
Task B	Humor	EfficientNet_b1	0.5004	0.5266
	Sarcasm	EfficientNet_b0	0.4915	0.5075
	Offensive	EfficientNet_b1	0.5087	0.5114
	Motivational	EfficientNet_b4	0.5010	0.5196

Table 5: Experiment Results

	Model	*Test Set Macro F1*	*Rank*	*Organizer Baseline*	*Best Score*
Task A	EfficientNet	0.3286	22	0.2176	0.3546
Task B	EfficientNet	0.5005	9	0.5002	0.5183

Table 6: Official Results.

7 Error Analysis

Although, System 1 was better at classifying the memes that had longer and more coherent sentences, it failed to classify memes that had short and incoherent sentences. Due to higher number of positive samples in the dataset, this system developed a tendency to classify the memes as positive in Task A. System 1 performed poorly on memes that had multiple panels or when two or more memes were stitched together. We also found that many of the samples contained texts that were unrelated to the context of the meme such as web addresses which might have thrown off the text-based classifier. Such unrelated text were often just for the promotion of the sites or users that host or create these memes. For example, "9GAG Fat Cat Memes Funny - 9gag.com", etc. And being random in nature, it is difficult to remove such unnecessary parts/noise from the texts. But these samples were somewhat more correctly classified by System 2 which uses only the images. Also, since System 2 does not use text, it failed to classify the samples where most of the context was in the text.

8 Conclusion

During the course of our experiments, we find that there are multiple challenges in the classification of internet memes. Memes don't adhere to any standards and they only continue to evolve. We tested out two approaches for classifying memes, using both text and image together and using only the images. Error analysis showed that the image-only approach is somewhat robust to noise in the text but fails to capture the context conveyed by the text. We suspect this task will get more difficult when we take into account memes that use video and sound. In future work, it will be interesting to explore pre-trained SimCLR models (Chen et al., 2020). These models are trained to learn generic representations from images in a self-supervised manner, which can be used to improve classification tasks.

References

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

David M Beskow, Sumeet Kumar, and Kathleen M Carley. 2020. The evolution of political memes: Detecting and characterizing internet memes with multi-modal deep learning. *Information Processing & Management*, 57(2):102170.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709.

Dawkins. 1976. *The Selfish Gene*. Oxford Univ. Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Jessica Drakett, Bridgette Rickett, Katy Day, and Kate Milnes. 2018. Old jokes, new media – online sexism and constructions of gender in internet memes. *Feminism & Psychology*, 28(1):109–127.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jean H French. 2017. Image-based memes as sentiment predictors. In *2017 International Conference on Information Society (i-Society)*, pages 80–85. IEEE.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. 2017. Squeeze-and-excitation networks.

Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Mia Xu Chen, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V. Le, Yonghui Wu, and Zhifeng Chen. 2018. Gpipe: Efficient training of giant neural networks using pipeline parallelism.

Simon Kornblith, Jonathon Shlens, and Quoc V. Le. 2018. Do better imagenet models transfer better?

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2019a. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. In *NIPS-W*.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. 2017. Swish: a self-gated activation function. *arXiv preprint arXiv:1710.05941*, 7.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2014. Imagenet large scale visual recognition challenge.

Benet Oriol Sabat, Cristian Canton Ferrer, and Xavier Giro i Nieto. 2019. Hate speech in pixels: Detection of offensive memes towards automatic moderation.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10.

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, Sep. Association for Computational Linguistics.

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

Mohammad Soleymani, David Garcia, Brendan Jou, Björn Schuller, Shih-Fu Chang, and Maja Pantic. 2017. A survey of multimodal sentiment analysis. *Image and Vision Computing*, 65:3–14.

Mingxing Tan and Quoc V. Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks.

J Williams. 2016. The good and the bad of political memes during election seasons. *The Chronicle, Independent News Organization at Duke's University*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Michael R. Zhang, James Lucas, Geoffrey Hinton, and Jimmy Ba. 2019. Lookahead optimizer: k steps forward, 1 step back.