BennettNLP at SemEval-2020 Task 8: Multimodal sentiment classification Using Hybrid Hierarchical Classifier

Ambuje Gupta, Harsh Kataria, Souvik Mishra, Tapas Badal, Vipul Mishra*
Department of computer science engineering
Bennett University, Greater Noida, India
(ambujegupta99, hkataria99, mishra.souvik911)@gmail.com
(tapas.badal, vipul.mishra)@bennett.edu.in

Abstract

Memotion analysis is a very crucial and important subject in today's world that is dominated by social media. This paper presents the results and analysis of the SemEval-2020 Task-8: Memotion analysis by team Kraken that qualified as winners for the task. This involved performing multimodal sentiment analysis on memes commonly posted over social media. The task comprised of 3 subtasks, Task A was to find the overall sentiment of a meme and classify it into positive, negative or neutral, Task B was to classify it into the different types which were namely humour, sarcasm, offensive or motivation where a meme could have more than one category, Task C was to further quantify the classifications achieved in task B. An imbalanced data of 6992 rows was utilized for this which contained images (memes), text (extracted OCR) and their annotations in 17 classes provided by the task organisers. In this paper, the authors proposed a hybrid neural Naïve-Bayes Support Vector Machine and logistic regression to solve a multilevel 17 class classification problem. It achieved the best result in Task B i.e 0.70 F1 score. The authors were ranked third in Task B.

1 Introduction

With the increased availability of internet and connected smart multimedia devices, there has been a rapid increase in the spreading of memes over social media platforms like Twitter, Facebook, Instagram, Reddit, etc. Memes are images containing text both of which are related to each other with a context. Memes usually pertain to an idea, belief, thought, theory, etc. which are usually derived from popular or unpopular culture with factors like place, people, events, and actions affecting it. These memes are an integral source of information about a community (Gal et al., 2016). This is why we require a thorough analysis of these memes. Not only that but also due to hate speech which has been the bane of social media since the beginning. These are the platforms where people from different parts of the world come together to interact and share ideas, such platforms encourage the concept of free speech however with this comes the issue of usage of offensive or anti-social content. Such content, which is not considered criminal, originates from the intent to harass in the name of criticism, leads to a radicalized person or community. Detecting such hate speech and offensive content to curb the spread of negativity and emotional distress using the internet is not only a priority task but a social responsibility. Such memes that cause trauma and torment to a directed person or a community needs to be flagged, reported, and taken proper action upon to avoid any unwanted disturbance to the people. This process of filtering online content has been traditionally dominated by manual checking of such content which is not possible considering the current internet activity. However, manual filtering is not only a time-consuming task but a possible mental health hazard too which indicates the need to automate this process. Even researchers in (Peirson et al., 2018), (Oliveira et al., 2016) were generating memes automatically.

Memes comprise multiple modalities which are textual and visual. These are context-related modalities and machines need a hybrid approach to leverage both these modalities. This paper discusses the data provided by the task organisers (Sharma et al., 2020) and the proposed approach towards the problem at

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1085–1093
Barcelona, Spain (Online), December 12, 2020.

hand. It goes through the data pre-processing and augmentation performed on the imbalanced set of data and the application of different machine learning and neural network training on the given data-set which includes recurrent neural network (RNNs) (Mikolov et al., 2010), long short term memory (LSTMs) (Greff et al., 2016) , Logistic regression (Indra et al., 2016) and neural Naive Bayes Support Vector Machine (NBSVM) (Wang and Manning, 2012).

2 Related Work

Not much solid work has been done to classify memes yet. In the classification of a meme, major work has been done on the image segment (Kolawole, 2015). In this work (Kolawole, 2015) the author has mainly focused on images of memes and has tried to take out the features of images such as lines, edges, and contrast or interest points such as blobs or corners. In meme classification there exists a lot of ambiguity because a meme consists of images and text both. By experimentation, it was found that text is the most important part of a meme as images follow a template, and what is written in that template matters more, rather than the template. People often confuse meme classification with sentiment analysis. Sentiment analysis is a small part of meme classification. A lot of work has been done in the sentiment analysis segment. The first task of meme classification is sentiment analysis i.e predicting whether a meme is positive, negative or neutral. Sentiment analysis is a Natural Language Processing(NLP) task and has many levels. Starting from the very beginning sentiment analysis started with a document level classification (Turney, 2002) moving further to sentence level (Kim and Hovy, 2004) and from that it came to phrase level (Wilson et al., 2005). If we see there is very little or no work done on sentiment analysis on memes, a lot of work has been done on sentiment analysis on twitter data. A very broad approach of sentiment analysis of twitter data was presented in (Pak and Paroubek, 2010). The author here collected the twitter data based on emoticons for example for positive ":) ,:-) " for negative " :(:-(" were used. The author's best approach was multinomial Naive Bayes. In (Go et al., 2009) author collected data through twitter and tried various classifiers.

3 Data

Data is the most important factor in any analytics and processing task. The data was provided to us by the competition organisers (Sharma et al., 2020) in two parts, the first set was a more of ill-prepared data which contained missing rows, missing annotations, missing images, swapped column values and all of this was revealed while working on it as it was creating a lot of unexpected issues, whereas the second set was an updated and cleaner but a similarly flawed version of the original data.

3.1 Dataset

The dataset included a folder of 6992 images and a comma-separated values(csv) file of annotated data corresponding to those images containing 6992 rows and 8 columns. Column headers included an index, image-name, text-ocr, text-corrected, humour, sarcasm, offensive, motivation, overall-sentiment. The annotations with their count were divided as shown in Table 1:

Table 1: Data Description with count

Humour	Sarcasm	Offensive	Motivation	Overall sentiment
Not funny(1651)	Not sarcastic(1544)	Not offensive(2713)	Motivational(2467)	Positive(4160)
Funny(2452)	General(3507)	Slight (2592)	Not motivational(4525)	Negative(631)
Very Funny(2238)	Twisted meaning(1547)	Very offensive(1466)		Neutral(2201)
Hilarious(651)	Very twisted(394)	Hateful offensive(221)		

3.2 Pre-processing

The columns in dataset: humour, sarcasm, offensive, motivation, overall-sentiment; were the major classes and were processed and prepared using basic one-hot encoding to be fed to the learning model directly. The column, image-name was used as it is to fetch the images, these images were further processed before putting them in the model and this process is explained in the later section. The text-corrected column was used to extract the text found in the corresponding memes and if this column had issues like missing values, null values, nonsensical values, wrong values, or unwanted symbols then the text-ocr column was used. There were instances where values from both the columns were missing, in such situations a custom Google's OCR was implemented to find out the text from these columns which involved obtaining an API from Google. Most of the pre-processing (Vijayarani et al., 2015) was done on the ocr-column.

This pre-processing helped to normalize the text by removing integer characters and replacing with textual references, removing contractions, removing non-ASCII characters, removing URLs, removing punctuation, and removing stopwords. Also, this converted the given text into lowercase and to its very basic form by stemming and lemmatization, and this was done by tokenizing the text. NLTK and regex were of much importance in performing these tasks. Finally, this resulted into something that could be used for putting in the training model which the machine could recognise and work on while eliminating the redundant stuff.

3.3 Data augmentation

Later it was realised that the data was not satisfactory and sufficient, so to obtain a good score data augmentation techniques were applied. The image data augmentation made not much sense as the memes were found to be following a sort of trend in the forms of templates. The text augmentation however included a series of techniques. With some analysis of the data it was found that the data was acutely imbalanced (Kotsiantis et al., 2005). Every class had highly disproportionate labels, to the extent that some were in a 1:9 ratio. we introduced redundant data by data replication. This up-sampling of data surprisingly increased the score by quite a lot, but all up-sampling did was bring in redundant data, which did not effect the learning of the machine.

4 Proposed Work

To solve this multi-class classification various approaches were tried, top performer presented here:

4.1 Transfer Learning Approach

The initial approach was transfer learning. Transfer learning is a machine learning method where a pre-trained model developed for a task is reused as a building block for a custom model on a similar task. For images of the memes, the VGG16/VGG19 (He et al., 2018) model was used. Here the bottom layers, except the final output layer, were used to make a custom model. These images were fed into the model and the predicted features were stored in a numpy array and saved. For the text extracted from the memes, BERT (Houlsby et al., 2019) model was used where a bunch of text was fed to the model and text features were stored in a NumPy array and saved. Both these NumPy arrays that were received from images and text were concatenated together to form a single numpy array that contained the features extracted from both the major factors i.e. image and text. After concatenating these numpy arrays, a feature matrix was formed which was then used in training the final model. Naïve-Bayes (Muhammad and Yan, 2015) and decision tree trained on this feature matrix to develop a final classifier. The maximum accuracy achieved using the decision tree model was 40 percent. After applying the transfer learning technique to all the 5 classes it was concluded that no satisfactory results were produced.

After multiple experiments, it was concluded that the image features were less contributing, meaning their presence in the feature array would not add much to the final output instead in some cases it would depreciate the final output quality. This was then analysed on a much lower level by picking up the actual pictures of memes and evaluating the annotations. It was found out that these pictures were following a trend by the means of templates. These templates could have a variety of text on them which would result in a totally different meaning. This was then making the data contradictory and decreasing the quality of the models.

4.2 Multi-step approach

In this approach first, the dataset was divided into multiple classes and has multiple layers as shown in Figure 1. For example, we first classified whether a meme is offensive or not offensive (binary classification). If the meme is offensive then we further classify to slight, very offensive, hateful-offensive(multi class classification).
We were getting good results in binary classification . We thought of converting multi-classification to binary classification. Let us assume we have 3 classes positive, negative, neutral. We took 2 classes at a time and trained it. So for one multi-class classification (which is converted to binary class classification) we made 3 models (Example - 1. Positive and negative 2. Positive and neutral 3. Negative and neutral). So by this method, we converted multi-class classification to binary-class classification. Algorithm 1 represents the pseudo-code at the time of testing.

Algorithm 1 Multi class classification to binary class classification

```
Function multi_class_classification_TO_binary_class_classification() :
input $x$;                                   ▷ Text input of a meme
K = predict($x$);                            ▷ K is the model for positive and negative
L = predict($x$);                            ▷ L is the model for positive and neutral
M = predict($x$);                            ▷ M is the model for negative and neutral
/* K, L, M will give their outputs accordingly. */
if $K==L$ then
    ▷ Basically taking the intersection of K and L
    print("Positive");
else if $K==M$ then
    print("Negative");
else if $M==L$ then
    print("Neutral");
else
    print("None");
```

In the first layer, the classes are 1. Overall Sentiment (positive, negative, and Neutral), Motivational, Not offensive, Offensive, Not sarcastic, Sarcastic, Not humorous, Humorous. In the second layer, humour is divided into 3 classes i.e Hilarious, Funny, and Very funny then sarcasm is divided into 3 classes i.e General, Very twisted, and Twisted meaning finally offensive is divided into 3 classes i.e Slight, Very offensive, and Hateful offensive.The second layer of classification was converted to binary classification as explained above. Now it was binary classification for offensive, sarcastic, humour. All the above models were tried but the best results were given by neural Naive Bayes Support Vector Machine (NBSVM) (Wang and Manning, 2012). In data pre-processing step 5-gram data was given to NBSVM. By applying this technique the results were good but this approach was ambiguous. If we look at algorithm 1 there might be a condition when none is given as an outcome so we come up with our final hybrid approach in the next section.

4.3 Hybrid NBSVM and logistic classifier based Approach

This hybrid approach consisted of two well proved algorithms first is Logistic Regression (Indra et al., 2016) and second is neural approach NBSVM (Wang and Manning, 2012) .

4.3.1 Naive Bayes Support Vector Machine(NBSVM)

We formulated our NBSVM linear classifier as:

$$y^{(q)} = sign(w^T x^{(q)} + b) \tag{1}$$

where, the prediction of q test cases is $y^{(q)}$ 'w' is weight matrix and $x^{(q)}$ is input vector. Let us assume $\alpha^i \in \mathbb{R}^{|U|}$ be the feature count vector for i training cases with label to be $y^i \in \{0, 1, 2\}$, where U is the set of features and f^i_j the number of occurrence of feature (U_k) in training cases i. A, B, C

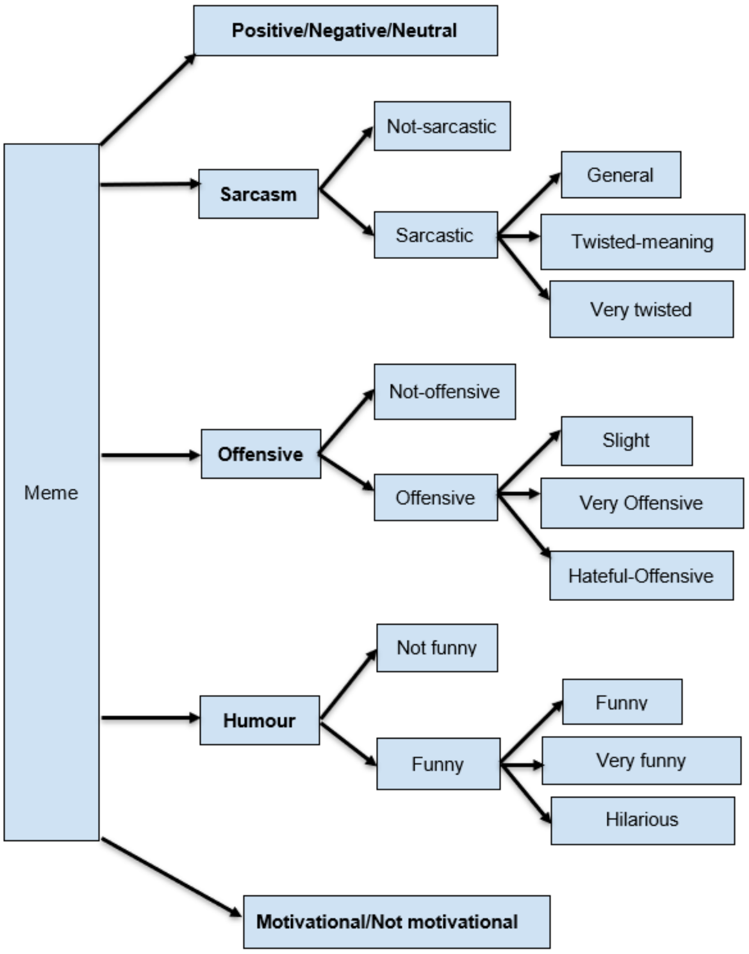

Figure 1: Tree of classes into subclass

defined as count vector which are as follows: $A = \varkappa + \sum_{i:y^{(i)}=0} f^i$ and $B = \varkappa + \sum_{i:y^{(i)}=1} f^i$ and $\left(C = \varkappa + \sum_{i:y^{(i)}=2} \alpha^i\right)$ for parameter smoothing ($\varkappa$). The log count ratio for A and B is defined as:

$$l = \log\left(\frac{A/||A||_1}{B/||B||_1}\right) \tag{2}$$

Similarly, we compute log count ratio for (B, C) and (C, A).

For **Support Vector Machine** $x^q = f^q$, and (W, B) are obtained by minimizing:

$$W^TW + C\sum_i max\Big(0, 1 - y^{(i)}\,(W^T\hat{\alpha}^{(i)} + B)\Big)^2 \tag{3}$$

Support Vector Machine with Naive Bayes (NBSVM) For NBSVM, we compute:

$$\tilde{f}^{(q)} = \hat{l} \circ f^{(q)} \tag{4}$$

This finds an interpolation between NB and SVM and therefore, $x^{(q)} = \tilde{f}^{(q)}$. The proposed algorithm used following model:

$$\mathbf{w}' = (1-\alpha)\,\hat{\mathbf{w}} + \alpha\mathbf{w} \tag{5}$$

where $\hat{\mathbf{w}} = \frac{\|\mathbf{w}\|_1}{|U|}$ is the mean magnitude of w and $\alpha \in [1, 0]$, is the interpolation parameter.

4.3.2 Logistic Regression

Logistic regression (Indra et al., 2016) falls under supervised classification algorithm. It is best for binary classification. We have used logistic regression where there was binary classification involved. It is a discriminate model which works by discriminating possible values of class y based on input x. Equation for this is :-

$$P(c|x) = \sum_{i=1}^{N} w_i.f^i \tag{6}$$

If we calculate $P(y/x)$ directly using equation 6, the result will lie between ∞ to $-\infty$. To obtain an output between 0 and 1 we have to use function:-

$$P(c|x) = \frac{1}{z} \exp \sum_{i=1} w_i.f^i \tag{7}$$

The hybrid approach had 8 different models working together to output the final result as shown in figure 2. All 8 models used either NBSVM (given in section 4.3.1) or logistic regression (given in section 4.3.1). The detail of these models are as follows:

First, is the **Overall sentiment** class model with the labels positive, negative, neutral which was trained using NBSVM. Second is the **motivational** or not motivational class model which was trained using logistic regression. The third is the **Humour** class model which was trained using logistic regression. It outputs whether the meme is funny or not funny. Fourth is the **Only funny** class model with the labels hilarious, funny, very funny which was trained using NBSVM. This model is only called if the humour class model outputs funny and the further results are calculated else the final result remains as not funny. Fifth is the **Sarcasm** class model which was trained using logistic regression. It outputs whether the meme is Sarcastic or not Sarcastic. Sixth is the **Only Sarcastic** class model with the labels twisted meaning, very twisted and general which was trained using NBSVM. This model is only called if the sarcasm class model outputs sarcastic and the further results are calculated else the final result remains as not sarcastic. Seventh is the **offensive** class model which was trained using logistic regression. It outputs whether the meme is offensive or not offensive. Finally eighth is the **Only offensive** class model with the labels hateful offensive, very offensive and slight which was trained using NBSVM. This model is only called if the offensive class model outputs offensive and the further results are calculated else the final result remains as not offensive. All these models were independently and individually trained. For training, the eight sub data sets were created. In general, we found during the experiment that, logistic regression performed better in binary classification such as motivational, humour, sarcasm, offensive. On the other hand in the case of multi-class classification neural approach of NBSVM performed better.

4.4 Inference

Once we successfully trained our models we then tested them with the testing dataset. A flow of the full testing process is defined in Figure 2. It starts with the input of the memes. Then the meme text is extracted from the image. Then pre-processing is performed on the extracted text, as defined in section 3.2. This pre-processed data is then fed into the trained hybrid model which consists of two layers of classification as shown in figure 2. In the first layer, it has five classifiers and three classifiers in the second layer. The proposed hybrid model classifies data using the first layer classifiers, then based on its result it uses the second level classifiers for further classification as shown in figure 2. For example, to classify a sarcastic meme, once the data is fed to the system our sarcasm model present in the first layer classifies the memes into Sarcastic or not Sarcastic, if the answer is sarcastic then the second layer classifier will classify it into twisted meaning, very twisted and general. The system performs similarly for offensive and humour as well, as shown in figure 2. After collecting all the labels, they are sent to the final result.

5 Results

The experiments were performed on DGX-1 V100 supercomputer. It is built using the NVIDIA Pascal powered Tesla V100 accelerators with 1000 TFLOPS speed, 40,960 cuda cores, 5120 ten-

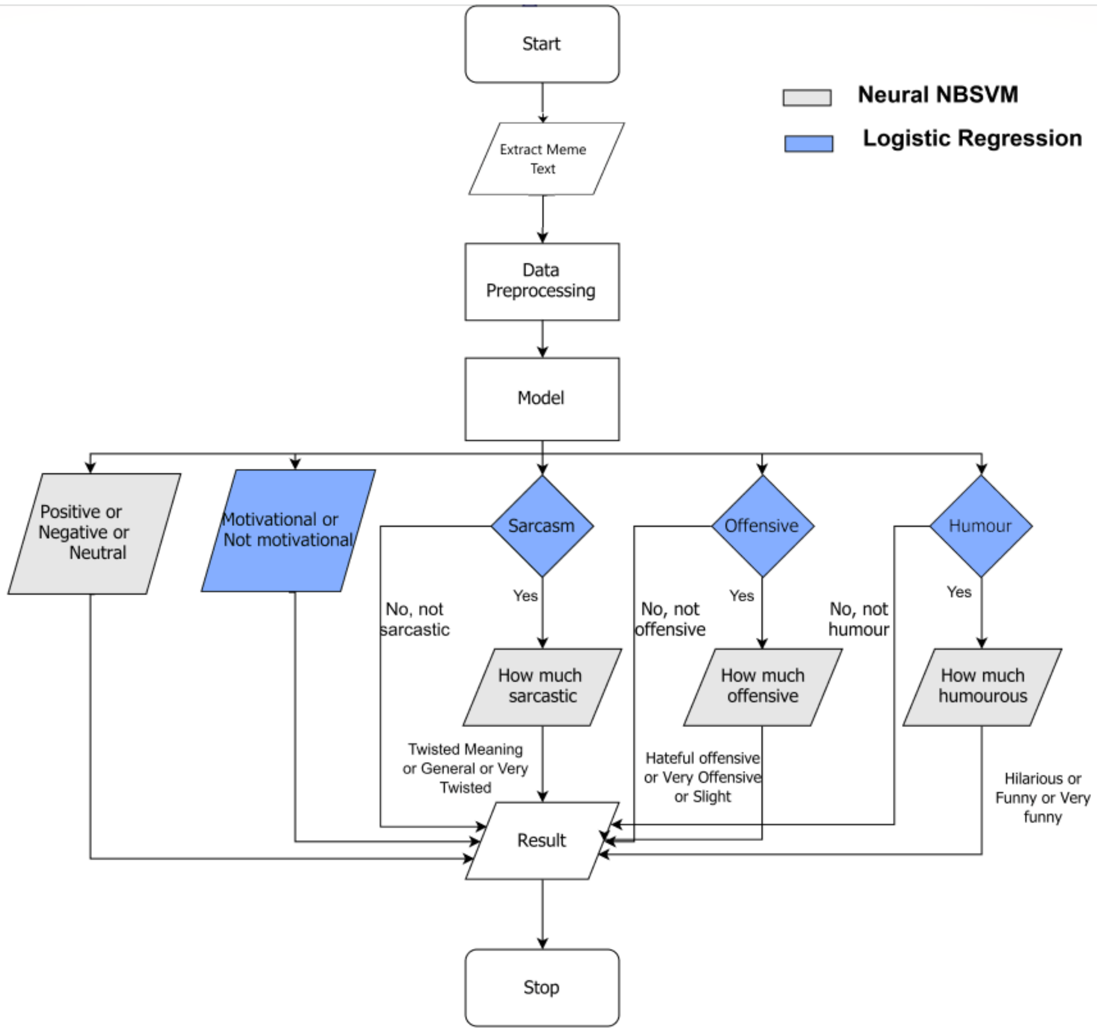

Figure 2: Flow chart for inference

sor cores. Python codes for cleaning the data, training the models and all the relevant scripts for testing and other tasks are available in a github repository. It can be accessed using this https://github.com/rockangator/memotion-analysis. The proposed approach has been implemented in python using Keras, NLTK, Ktrain, Tensorflow, scikit learn, re, os libraries. The results are shown in Table 2, 3, 4, 5, 6, 7, 8 based on hybrid NBSVM and logistic classifier approach. As shown in Table 2 our model has given precision - 0.95, recall- 1.00 and f1 score of 0.97 by classifying how **negative** the memes are. Moreover, proposed methodology is able to classify **not motivational** memes with precision of 0.80, recall - 0.65 and f1 score of 0.71, as shown in Table 3. Our model was able to classify memes as **offensive** with 0.74 precision, 0.62 recall and 0.68 f1 score, **hateful offensive** with precision - 0.97, recall - 1.00 and f1 score 0.99, **sarcastic** with 0.95 precision, 0.90 - recall and 0.90 f1 score. **Very Twisted** memes were classified with 0.97 precision, 1.00 recall and 0.98 f1 score. The model classified a meme as **funny** with 0.98 precision, 0.87 recall and 0.92 f1 score. After funny was classified the model was able to classify **humorous** meme with 0.75 precision, 0.64 recall and 0.69 f1 score as shown in Table 4, 5, 6, 7, 8, 9 respectively.

Finally as compared to (Go et al., 2009) our proposed hybrid model achieved 6% higher accuracy. In (Go et al., 2009) the authors achieved best accuracy of 81 percent in contrast, our proposed hybrid model for sentiment analysis of memes achieved 87 percent using Neural Naive Bayes Support Vector Machine(NBSVM).

Table 2: Classification Report for overall sentiment (Task A)

Positive Negative Neutral				
	Precision	Recall	f1-score	Support
positive	0.90	0.66	0.76	420
negative	0.95	1.00	0.97	443
neutral	0.77	0.93	0.84	435
accuracy			0.87	1298
macro avg	0.87	0.86	0.86	1298
weighted avg	0.87	0.87	0.86	1298

Table 3: Classification Report for motivation (Task B)

Motivational - Not Motivational				
	Precision	Recall	f1-score	Support
not-motivational	0.80	0.65	0.71	474
motivational	0.65	0.79	0.71	383
accuracy			0.71	857
macro avg	0.72	0.72	0.71	857
weighted avg	0.73	0.71	0.71	857

Table 4: Classification Report for Offensive-not offensive (Task B)

Offensive - Not Offensive				
	Precision	Recall	f1-score	Support
not-offensive	0.67	0.78	0.72	428
offensive	0.74	0.62	0.68	429
accuracy			0.70	857
macro avg	0.71	0.70	0.70	857
weighted avg	0.71	0.70	0.70	857

Table 5: Classification Report for Slight - Very Offensive - Hateful Offensive (Task C)

Slight - Very Offensive - Hateful Offensive				
	Precision	Recall	f1-score	Support
slight	0.84	0.76	0.79	268
very-offensive	0.75	0.81	0.78	219
hateful-offensive	0.97	1.00	0.99	262
accuracy			0.86	749
macro avg	0.85	0.86	0.85	749
weighted avg	0.86	0.86	0.86	749

Table 6: Classification Report for sarcastic - not sarcastic (Task B)

Not sarcastic - Sarcastic				
	Precision	Recall	f1-score	Support
not-sarcastic	0.90	0.91	0.91	554
sarcastic	0.95	0.90	0.90	546
accuracy			0.90	1100
macro avg	0.90	0.90	0.90	1100
weighted avg	0.90	0.90	0.90	1100

Table 7: Classification Report for General - Twisted Meaning - Very Twisted (Task C)

General - Twisted Meaning - Very Twisted				
	Precision	Recall	f1-score	Support
general	0.92	0.78	0.84	363
twisted-meaning	0.81	0.92	0.86	321
very-twisted	0.97	1.00	0.98	331
accuracy			0.89	1015
macro avg	0.90	0.90	0.89	1015
weighted avg	0.90	0.89	0.89	1015

Table 8: Classification Report for Funny - Not Funny (Task B)

Funny - Not Funny				
	Precision	Recall	f1-score	Support
not-funny	0.88	0.98	0.93	497
funny	0.98	0.87	0.92	533
accuracy			0.93	1030
macro avg	0.93	0.93	0.93	1030
weighted avg	0.93	0.93	0.93	1030

Table 9: Classification Report for Hilarious-Funny-Very Funny (Task C)

Hilarious - Funny - Very Funny				
	Precision	Recall	f1-score	Support
hilarious	0.75	0.64	0.69	389
funny	0.35	0.47	0.40	244
very-funny	0.33	0.29	0.31	232
accuracy			0.50	865
macro avg	0.48	0.47	0.47	865
weighted avg	0.52	0.50	0.51	865

6 Conclusion

We encourage systematic research and development in the field of memotion analysis. Using combined machine learning and training methods i.e quantitative and qualitative, we were able to create a model and empirically validate our results on a dataset of 6992 data points. The results are very encouraging and quite a remarkable feat (secure third position in task B). Memotion analysis can be one of the most utilised tools in this era of social media (and in most cases, better than) among all the other highly regarded sentiment analysis tools that are available to us till date.

References

Noam Gal, Limor Shifman, and Zohar Kampf. 2016. “it gets better”: Internet memes and the construction of collective identity. *New media & society*, 18(8):1698–1714.

Alec Go, Lei Huang, and Richa Bhayani. 2009. Twitter sentiment analysis. final projects from cs224n for spring 2008/2009.

Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. 2016. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232.

Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. 2018. Bag of tricks for image classification with convolutional neural networks.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp.

ST Indra, Liza Wikarsa, and Rinaldo Turang. 2016. Using logistic regression method to classify tweets into the selected topics. In *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 385–390. IEEE.

S Kim and E Hovy. 2004. Determining the sentiment of opinions [in:] coling’04 proceedings of the 20th international conference on computational linguistics. *Geneva 2004*.

Olamide Temitayo Kolawole. 2015. *Classification of internet memes*. Ph.D. thesis.

Sotiris Kotsiantis, D. Kanellopoulos, and P. Pintelas. 2005. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30:25–36, 11.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*.

Iqbal Muhammad and Zhu Yan. 2015. Supervised machine learning approaches: A survey. *ICTACT Journal on Soft Computing*, 5(3).

Hugo Gonçalo Oliveira, Diogo Costa, and Alexandre Miguel Pinto. 2016. One does not simply produce funny memes!–explorations on the automatic generation of internet humor. In *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016). Paris, France*.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326.

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

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.

Peter D Turney. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics.

S Vijayarani, Ms J Ilamathi, and Ms Nithya. 2015. Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, 5(1):7–16.

Sida Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL ’12, page 90–94, USA. Association for Computational Linguistics.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 347–354.