UoR at SemEval-2020 Task 8: Gaussian Mixture Modelling (GMM) Based Sampling Approach for Multi-Modal Memotion Analysis

Zehao Liu[1], **Emmanuel Osei-Brefo**[1], **Siyuan Chen**[2], **Huizhi Liang**[1]
[1]University of Reading
White Knights, Berkshire, RG6 6AH
United Kingdom
[2]University of New South Wales
High street, Kensington NSW 2033
Australia
zehao.liu@reading.ac.uk, e.osei-brefo@pgr.reading.ac.uk,
siyuan.chen@unsw.edu.au, huizhi.liang@reading.ac.uk

Abstract

Memes are widely used on social media. They usually contain multi-modal information such as images and texts, serving as valuable data sources to analyse opinions and sentiment orientations of online communities. The provided memes data often face an imbalanced data problem, that is, some classes or labelled sentiment categories significantly outnumber other classes. This often results in difficulty in applying machine learning techniques where balanced labelled input data are required. In this paper, a Gaussian Mixture Model sampling method is proposed to tackle the problem of class imbalance for the memes sentiment classification task. To utilise both text and image data, a multi-modal CNN-LSTM model is proposed to jointly learn latent features for positive, negative and neutral category predictions. The experiments show that the re-sampling model can slightly improve the accuracy on the trial data of sub-task A of Task 8. The multi-modal CNN-LSTM model can achieve macro F1 score 0.329 on the test set.

1 Introduction

Contents of social media are typically multi-modal while traditional Natural Language Processing (NLP) and computer vision methods only process text or image data respectively. In recent times, Memes have frequently been used in internet communities and are symbols of modern internet culture (Gal et al., 2016). It typically contains both images and texts to express a certain semantic meaning. Sentiment analysis can classify whether a users' opinion is positive based on both the meaning of the image and the metaphor of the text. So it is important to develop multi-modal data process methods to understand the conveyed semantic meaning of the memes. The given data of SemEval-2020 Task 8 include 6992 human labelled memes image with its text content (Sharma et al., 2020). In sub-task A, the requirement is to classify a given meme into positive, negative or neutral category.

Social media data are typically severely imbalanced. The given data in the sub-task A is also in this case where the instances of the positive class are around 6.59 times more than that in the negative class. Class imbalance refers to skewed class distributions in datasets, where the number of samples of one class is significantly greater than the other classes. Existing studies shows that class imbalance problems can impose negative effects on the performance of a classification problem (Zhou and Liu, 2006). This is because classifier algorithms are often biased towards the majority classes (Ramanan et al., 1998). Multi-class data are however difficult to balance since the relationship between classes are not straight forward.

Dealing with multi-class problems such as the Memotion analysis problem poses some practical challenges that results in a loss of performance in one data class whilst trying to compensate it in the other class (Sáez et al., 2016). Chen et. al. (2015) used Gaussian Mixture Models (GMMs) as embedding for words vectors, which allows words to have multiple meaning. GMMs have been shown to capture a full distribution of images and can also generate realistic image samples although they are not as sharp as the ones from generative adversarial networks (Richardson and Weiss, 2018). To handle the data imbalance problem, we therefore propose a Gaussian Mixture sampling method to balance imbalanced text sequence vectors.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1201–1207
Barcelona, Spain (Online), December 12, 2020.

2 Related Work

The usage of memes in internet communities has been significantly increased in recent years (Xu, 2017). Memes are images with texts that are often recreated from prior art content, including painting, operas, cartoon or movie, which typically contains metaphors that help the users express their feelings and emotions (French, 2018). However, it is found challenging to detect offensive content automatically, since the content is difficult to understand comparing to textual hate speech. Moreover, understanding memes is an important way to know public opinions on social media, thus automatic multi-modal content analysis has attracted great attentions in the research communities. French (2018) tried to analyse the relevance between memes images and texts on social media in terms of semantic meaning. They found that memes were typically used to emphasis the discussions related to users' sentiment. Some other research focused on automatic methods to detect sexiest in multi-modal memes (Fersini et al., 2019).

Recently, many research began to focus on multi-modal methods in deep learning, which can achieve excellent performance on features extraction due to its' nonlinear learning ability. Convolutional Neural Network (CNN) and Long Short Term Memory network (LSTM) models are popularly used to combine image and text features for sentiment classification. In (Xu, 2017), CNN-Multi methods were proposed to concatenate image and text features to classify sentiment. Moreover, attention based mechanism has been proposed to focus on a specific entity in text and image, aiming to minimise the differences between image and text vectors to train a model (Nam and Kim, 2017).

However, only few approaches have been proposed for handling multi-class imbalanced data. The Binarization approach known as one-versus-one (pairwise learning) proposed by (Rifkin and Klautau, 2004) in their work and described by (Fernández et al., 2013) was adapted in conjunction with a hybrid combined approach to overcome the multi-class imbalance problem faced. Fernández et., al (2011) applied Static-SMOTE method involving a re-sampling procedure in T steps, where T represents the number of classes. In each iteration, the re-sampling technique selects the minority size class, and duplicates the number of instances of the class in the original dataset.

3 Data Description

In the originally released dataset, there are 6988 valid referenced meme images out of 6992 entries, which are used as a training set. Also, the released trial set contains 914 (910 valid) different labelled entries, which are used to evaluate performance of models. The test set contains 1878 (1877 valid) unlabelled entries, and produces the results for submission. The original column of "overall_sentiment" has 5 classes, which have been aggregated into 3-class format as sub-task A required. In the training set, there are 4156 meme images belonging to the positive class, 2201 and 631 meme images belonging to the neutral and negative classes respectively making the data highly unbalanced. Next we discuss how text and image data have been prepared before modelling.

3.1 Data Pre-processing

The "corrected_text" column in the file "labels.csv" contains human corrected OCR text from the images, which was used in our model. Five rows of the text that contains irrelevant html tags have been removed before processing. The function of tokenizer in Keras was used to transform text data into text sequence vectors by assigning unique integer to different vocabulary. In the training set, the vocabulary size was 13367. Due to the maximum length of text, which was 192, shorter text sequence was padded with 0 to keep the vector shape of 192 x 1. It allowed all the different shapes of text sequence vector feeding into the model, but it also made the vector sparse. CV2 is an extended library of the OpenCV package in computer vision toolkit (Howse, 2013), was used to read the meme images. Due to memory limit, each image was shrunk to 128 x 128 in resolution from the original images whose average size was 547 x 587. We kept the 3 RGB channels for colour picture, and duplicated missing channels. Therefore, each image was transformed to 3D vector with a shape of 128 x 128 x 3. In addition, because the range of pixel values was between 0 and 255, all the image vector was divided by 255 to standardise data.

3.2 Methodology Applied to Balancing Data-set

For Task A, which involves a multi-class Sentiment Classification, Gaussian Mixture Model (GMM) was used as an oversampling technique to balance the dataset. This was achieved through a generation of synthetic text data for the Neutral and Negative data classes referred as minority class 1 and minority class 2 respectively. The number of text data over-sampled in each of the two minority classes were each in such a way that they were both equivalent to the difference in their respective number of observations and that of the positive class referred as majority class respectively. The procedure involved is outlined below. Foremost, the dataset with the multi-class target variable was divided into 3 distinct classes, namely the majority class, the minority class 1 and the minority class 2, together with their corresponding independent variables.

A GMM algorithm was applied to the minority class 1 to generate extra synthetic text data samples in such a manner that it equalled to the the total number of observation for the Majority class. The GMM algorithm was again applied to the minority class 2 to generate extra minority class 2 text data samples in a manner that the total observations in that class became equal to the majority class. The split training set made up of the majority class, minority class 1 and minority class 2 were all merged with the respective over-sampled data samples of the two minority classes to form a new balanced dataset. A summary of the above procedure has been presented as Algorithms 1 and 2 in the Appendix section.

4 System Description

4.1 LSTM Model for Text

Recurrent neural networks (RNN) can process a series of sequence information. The output of each element in the sequence is related to previous elements, so the RNN model can remember information in the previous step (Dai et al., 2016). LSTM is a kind of RNN, which combines long-term and short-term memory. Since the text vector can be regarded as a sequence pattern, the model can learn the relationships between different words in context. In the proposed text-only model, we mapped each 192-dimension text vector into 100-dimension embedding vectors as input. The original sparse text sequence vector was compressed into denser representation, which is more effective in learning. Secondly, the 100 x 192 embedding output connected to a 128-node LSTM layer to extract features in the sequence and transformed them into 128 dimensions. Next, the model flattened each vector as a scalar, and connected with a 32-node fully connected layer to compress the data into 32 bits with Rectify Linear Unit (ReLu) activation. The dense layer again extracted the hidden features. Finally, another dense layer mapped the data into a 3-bit value to form a classification learning problem, which used softmax function to non-linearly activate output. Dropout layers was added to regularise the model, which can resolve over-fitting problem. Also, the model used categorical cross entropy as loss function and “adam” optimiser for stochastic gradient descent optimization.

4.2 CNN Model for Image

CNNs have been proven to be useful for object recognition in images, detecting digits, faces and objects with varying orientations because their spatial structures are preserved and learned by internal feature representation (LeCun et al., 2010). This motivated us to employ a CNN model to learn the internal features of the image data for Memotion analysis. In our system, the CNN model consisted of two convolution layers along with two pooling layers and one fully connected layer for an effective learning, considering the size of this Memotion dataset. The input was the given images scaled to 128 × 128 pixels with three colour channels. 32 filters moved across the whole image with a fixed receptive field 3 × 3 to learn features. These features were expressed in feature maps as an output by collecting the results of neuron activation with ReLu. A max pooling layer was followed to down-sample the feature maps to 64 × 64 with an intention to compress and generalise feature representations. The structure was repeated by connecting two convolutional layers with a 64 and 128 filters and two max pooling layers vertically to make the neural network deeper. At the end of network, the 16 × 16 square feature maps were flattened out into a flat fully connected layer with 128 hidden neurons. Finally, a 3-node dense layer with softmax activation function was used to output probabilities of the predicted class. This CNN

network architecture for image data was combined with the LSTM model for text data, so the model for Memotion could be learned together in an end-to-end manner.

4.3 Fusing Text and Image data

Recently there has been renewed interest from researchers in the use of multi-modal fusion model to combine textual features and image features, as it is beneficial for features learning to integrate different types of data as joint representation (Wang et al., 2016). Features vectors can be merged together as a joint representation. Xu (2017) and Nam & Kim (2017) concatenated image and text vectors as multimodal features. We concatenated the last dense layer before the output layer of the LSTM and CNN model to build a fusion model, which combines the representation for a unified training.

To combine the two models, their output shape should be identical. We therefore adapted the previous LSTM and CNN model to output 128-dimension vectors from their dense layers. They were then directly concatenated into a 256-dimension joint representation. The output was used as input for the fully-connected layers to further learn the hidden features. The model output and compiling method were the same as the previous model, which formed the 3-class classification learning task.

5 Experiments and Results

5.1 Experimental Setup

Google Colab was used to implement the experiments in a Python 3 environment. Google Colab provides Tesla P100-PCIE-16GB GPU and 25GB RAM. Keras and Tensorflow 2.0 have been introduced to build the neural network models. The GPU could accelerate neural network training process significantly, but it creates some difficulty in terms of replicability, because the normal random seed cannot control randomized variables in GPU computations. The model parameters were tested and set empirically, based on the loss in the validation results.

5.2 Results

Accuracy and macro F1 score were chosen to evaluate our model. The accuracy shows the percentage of correct predictions out of all predictions. The F1 score is the harmonic average of the precision and recall. In multi-class classification problem, Macro F1 score is calculated by averaging the F1 scores of each class, which has been adopted by the organisers of the competition. It thus requires the model to have a balanced performance on each class to achieve good macro F1 score. Table 1 shows the results of sub-task A from all the 4 test models with or without data sampling method, which are all above the baseline F1 score. Due to the fact that GPU computation has unknown random seed that cannot be controlled, each model was trained 5 times and recorded the range and the mean of metrics. Also, the validation split function in Keras was used to conduct 20% hold-out validation. The validation accuracy only showed the value when the model fully converged around 20 epochs. The accuracy and F1 score on the trial set are the crucial metrics used to select models. Regarding text-only models, the LSTM models show overall better metrics than the CNN-1D models.

From Table 1, we also can see that the sampling methods improved the accuracy but decreased the F1 score, which suggested that it increased the number of corrected predictions overall, but decreased the average performance of all the classes, leading to lower macro F1 scores in both trial set and test set. In other words, although this sampling method can improve overall accuracy, it leads to a bigger difference among the performance of each class. Meanwhile, image-based model demonstrates better performance than all the text-based models. Regarding the fusion model CNN-LSTM-TI, it achieved the highest F1 score and accuracy on the trial set, 72.9% F1 score in average, suggesting an effective learning from joint features in sub-task A. However, in the test set, all the models produced similar macro F1 scores from 0.33-0.34. Among them, the CNN-I models produced the best result, 0.337.

6 Conclusions

Memes analysis can help people better understand opinions on social media. However, memes data in the real world are typically unbalanced. The given memes dataset also demonstrates a significant

Classifier	Data	Macro F1 Score (%)		Accuracy (%)	
		Trial set	Test set	Trial set	Valid set
Baseline	-	-	0.217	-	-
LSTM-T	Text only	0.310-0.321 (avg: 0.316)	0.332	0.56-0.57	0.45-0.46
LSTM-T	Text only with GMM	0.251-0.253 (avg: 0.252)	0.327	0.61-0.62	0.53-0.54
Conv1D-T	Text only	0.290-0.298 (avg: 0.296)	0.333	0.55-0.56	0.47-0.49
Conv1D-T	Text only with GMM	0.252-0.271 (avg: 0.264)	0.325	0.55-0.56	0.48-0.59
CNN-I	image only	0.697-0.727 (avg: 0.710)	0.337	0.77-0.79	0.46-0.51
CNN-LSTM-TI	image + text	0.710-0.754 (avg: 0.729)	0.329	0.78-0.81	0.47-0.52

Table 1: Task 8A results

skewed distribution among classes, which could impact model training. We proposed and experimented a Gaussian Mixture sampling method on the text data to handle the imbalance problem. The results show that the method slightly improved the overall multi-class classification prediction accuracy. We also experimented a multi-modal CNN-LSTM model (i.e., CNN-LSTM-TI in the experiments) to accommodate both text and image information. It achieved better performance than the text only or image only single-modal models in the experiments.

There are some limitations in our models. Due to the memory limits, we only used images in the size of 128 x 128. In future work, larger size of images and other pre-trained text embedding could be used to further improve the performance. We only tested 3-channel RGB images while 1-channel gray-scale images can also be explored. In addition, further research could be focused on applying the proposed Gaussian Mixture sampling method on image and fusion model.

References

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer-Verlag New York Inc.

Xinchi Chen, Xipeng Qiu, Jingxiang Jiang, and Xuanjing Huang. 2015. Gaussian mixture embeddings for multiple word prototypes. *arXiv preprint arXiv:1511.06246*.

Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. 2016. Deep Coevolutionary Network: Embedding User and Item Features for Recommendation. sep.

Alberto Fernández, Victoria López, Mikel Galar, María José [del Jesus], and Francisco Herrera. 2013. Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. *Knowledge-Based Systems*, 42:97 – 110.

F. Fernández-Navarro, C. Hervás-Martínez, and P. Antonio Gutiérrez. 2011. A dynamic over-sampling procedure based on sensitivity for multi-class problems. *Pattern Recognition*, 44(8):1821–1833. cited By 83.

Elisabetta Fersini, Francesca Gasparini, and Silvia Corchs. 2019. Detecting Sexist MEME on the Web: A Study on Textual and Visual Cues. In *2019 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos, ACIIW 2019*, pages 226–231. Institute of Electrical and Electronics Engineers Inc., sep.

Jean H. French. 2018. Image-based memes as sentiment predictors. In *International Conference on Information Society, i-Society 2017*, volume 2018-Janua, pages 80–85. Institute of Electrical and Electronics Engineers Inc., may.

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.

Joseph Howse. 2013. *OpenCV computer vision with python*. Packt Publishing Ltd.

Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. 2010. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, pages 253–256. IEEE.

Hyeonseob Nam and Jeonghee Kim. 2017. Dual Attention Networks for Multimodal Reasoning and Matching Jung-Woo Ha Naver Labs. pages 299–307.

S. Ramanan, T.G. Clarkson, and J.G. Taylor. 1998. Adaptive algorithm for training pram neural networks on unbalanced data sets. *Electronics Letters*, 34(13):1335–1336. cited By 4.

Eitan Richardson and Yair Weiss. 2018. On gans and gmms. *CoRR*, abs/1805.12462.

R. Rifkin and A. Klautau. 2004. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141. cited By 1175.

Chhavi Sharma, Deepesh Bhageria, William Paka, Scott, Srinivas P Y K L, Amitava Das, Tanmoy Chakraborty, 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.

José A. Sáez, Bartosz Krawczyk, and Michał Woźniak. 2016. Analyzing the oversampling of different classes and types of examples in multi-class imbalanced datasets. *Pattern Recognition*, 57:164 – 178.

Liwei Wang, Yin Li, and Svetlana Lazebnik. 2016. Learning deep structure-preserving image-text embeddings. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem(Figure 1):5005–5013.

Nan Xu. 2017. Analyzing multimodal public sentiment based on hierarchical semantic attentional network. In *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 152–154. IEEE, jul.

Z.-H. Zhou and X.-Y. Liu. 2006. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77. cited By 661.

7 Appendix

Algorithm 1 Gaussian Mixture Model Algorithm

1: **Input:** N training samples such that $x_1, x_2 ... x_N \in \mathbf{X}$

2: Initialize the means $\boldsymbol{\mu}$, co-variances $\boldsymbol{\Sigma}_i$ and mixture weights $\boldsymbol{\pi}_i$ and evaluate the initial log-likelihood.

3: **E STEP:** Determine the conditional probability for each mixture component i to be responsible for observation x_N using current parameter values:

$$\phi(Z_{ni}) = \frac{\pi_i \mathcal{N} \mathbf{x}_N | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i}{\Sigma_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_N | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{1}$$

4: **M STEP:** Re-estimate the parameters using the current conditional probabilities

$$\boldsymbol{\mu}_i^{new} = \frac{1}{N_i} \sum_{n=1}^{N} \phi(Z_{ni}) \mathbf{x}_N, \Sigma_i^{new} = \frac{1}{N_i} \Sigma_{n=1}^{N} \phi(Z_{ni}) (\mathbf{x}_N - \boldsymbol{\mu}_i^{new})(\mathbf{x}_N - \boldsymbol{\mu}_i^{new})^{\mathrm{T}},$$

$$\pi_i^{new} = \frac{N_i}{N}, \text{where} N_i = \sum_{n=1}^{N} \phi(Z_{ni})$$

5: The old parameters are replaced by the new ones and the log-likelihood is calculated as below:

$$\ln p(\mathbf{X} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^{N} \ln \left\{ \sum_{i=1}^{K} \pi_i \mathcal{N}(\mathbf{x}_N | \boldsymbol{\mu}_i, \Sigma_i) \right\} (Bishop, 2006) \tag{2}$$

6: Check that the stopping criteria such as maximal number of iterations is reached or the relative change of the last two log-likelihoods . If the stopping criterion is not met return to step 3 to repeat.

7: **Output:**

Algorithm 2 GMM re-sampling Technique

1: **input**: For a given dataset sample $\mathbf{X}$ with multi class $y_i \in \{0, 1, 2\}$
2: Split the training dataset $\mathbf{X}_{tr}$ into majority class $\mathbf{X}^{maj}$, the minority class 1 $\mathbf{X}^{min1}$ and minority class 2 $\mathbf{X}^{min2}$
3: **for** The Minority Class 1, $\mathbf{X}^{min1}$: **do**
4: Apply the GMM Algorithm operation in Algorithm 1 in such a way that $N_1^{min} \leftarrow N^{maj}$ samples to produce new samples; $\mathbf{x}^{ns1}$
5: **for** The Minority Class 2, $\mathbf{X}^{min2}$: **do**
6: Apply the GMM Algorithm operation in **Algorithm 1** in such a way that $N_2^{min} \leftarrow N^{maj}$ samples to produce new samples; $\mathbf{x}^{ns2}$
7: Merge the training datasets $\mathbf{X}^{maj}$, $\mathbf{X}^{min1}$ and $\mathbf{X}^{min2}$ with the over-sampled data points; $\mathbf{x}^{ns1}$ and $\mathbf{x}^{ns2}$ together to form a new balanced dataset $\mathbf{X}_{BD}^{os}$
8: End
9: **Output:** Generate Balanced re-sampled Dataset