UAEM-ITAM at SemEval-2022 Task 5: Vision-Language Approach to Recognize Misogynous Content in Memes

Edgar Roman-Rangel
Instituto Tecnológico Autónomo de México
Cd. de México
México
edgar.roman@itam.mx

Jorge Fuentes-Pacheco
CONACyT-CInC-UAEM
Cuernavaca, Morelos
México
jorge.fuentes@uaem.mx

Jorge Hermosillo Valadez
CInC - Universidad Autónoma del Estado de Morelos (UAEM)
Cuernavaca, Morelos
México
jhermosillo@uaem.mx

Abstract

In the context of the Multimedia Automatic Misogyny Identification (MAMI) competition 2022, we developed a framework for extracting lexical-semantic features from text and combine them with semantic descriptions of images, together with image content representation. We enriched the text modality description by incorporating word representations for each object present within the images. Images and text are then described at two levels of detail, globally and locally, using standard dimensionality reduction techniques for images in order to obtain 4 embeddings for each meme. These embeddings are finally concatenated and passed to a classifier. Our results overcome the baseline by 4%, falling behind the best performance by 12% for Sub-task B.

1 Introduction

The Multimedia Automatic Misogyny Identification (MAMI) competition (Fersini et al., 2022) consists in the identification of misogynous memes, taking advantage of both text and images available as source of information. The task was organized around two main sub-tasks. Sub-task A: a basic task about misogynous meme identification, where a meme should be categorized either as misogynous or not misogynous; Sub-task B: an advanced task, where the type of misogyny should be recognized among potential overlapping categories such as stereotype, shaming, objectification, and violence.

In this paper, we present a proposed solution for Sub-task B only, which consists of a framework for extracting lexical-semantic features from text and combine them with semantic descriptions of images, together with image content representation. We propose to use a pre-trained BERT model (Devlin et al., 2019) as lexical feature enhancer, and to use a vision-language model (Zhang et al., 2021) as visual descriptor.

The rest of this paper is organized as follows. We introduce our multimodal framework in Section 2. In Section 3 we present and discuss our results. We conclude the paper in Section 4.

2 Methods

Fig. 1 shows a diagram of the method that we propose to describe memes using both visual and text inputs. It is known that local descriptors can provide rich representations, as they can extract fine details from local areas within documents (Lowe, 2004). Therefore, we compute two types of description for each modality -visual or text-. Namely, global and local descriptors, and then concatenate the resulting four descriptors into a single vector that we use for classification.

More specifically, we relied on transfer learning coupled with fine tuning procedures, in which one pre-trained model was further adjusted for each of

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 605 - 609
July 14-15, 2022 ©2022 Association for Computational Linguistics

the four input modalities, and for the five classes problem presented by the MAMI competition.

2.1 Image descriptors

We describe images at two levels of detail. First globally using a pre-trained CNN, and also locally by detecting and describing individual objects within the images.

2.1.1 Global descriptors

In order to generate a global visual description of the image, we use the InceptionV3 (Szegedy et al., 2016) network pretrained on ImageNet (Russakovsky et al., 2015) to perform a fine tuning on the MAMI dataset. The classification head is replaced with a global average pooling, a dropout layer, and 4 dense layers with weights randomly initialized. The first three dense layers have 1024, 512, and 64 units. The last dense layer has five units with sigmoid activations.

To train the network, first the convolutional basis is frozen and the parameters of the classification head are optimized for ten epochs using a learning rate of $1e^{-3}$. Next, all parameters are unfrozen and retrained with a smaller learning rate ($1e^{-5}$). The training is stopped by using the regularization strategy of Early Stopping to avoid overfitting. Once the training process is completed, a 64-vector of floating-point values is generated for each image. This descriptor is obtained in inference mode from the output of the penultimate dense layer.

2.1.2 Local descriptors

We detect and describe each of the objects contained in the images by means of the pre-trained VinVL model (ResNeXt-152 C4 architecture) (Zhang et al., 2021). VinVL was pre-trained on four public datasets specialized in object localization, so it considers up to 1848 object categories and 524 attribute categories (nouns and adjectives respectively).

This stage generates a feature vector of length 2048 for each object within the image. More precisely, this step produces a matrix of varying length according to the number of objects detected in an image, where each component is a fixed-length vector of size 2048. We post-process this matrix using Principal Component Analysis (PCA) on both of its axes, and recovering the eighth principal components for each axis. This is, we identify features corresponding to the eighth most relevant objects in a given image, as well as those corresponding to the eighth most relevant variables describing each object representation, both of them in an orthogonal space of PCA that is independent across images.

This PCA processing results in a 64-D representation of the visual features for all object detected within an image.

2.2 Text representations

Analogous to the image processing stage, we also describe the meme's text transcriptions at two levels of granularity. First, generating an embedding for the whole sentence, and then incorporating individual word representations.

2.2.1 Contextual embeddings

We generated a global sentence embedding for each meme transcription. This was performed using a pre-trained BERT model (Devlin et al., 2019). Namely, the small uncased BERT "L-4_H-512_A-8" for English language (Turc et al., 2019). This model was used up to the layer that produces its so-called pooled output, which provides a sentence embedding vector of 512 elements. We connected such output to a classification multi-layer perceptron (MLP) for fine tuning the model.

The classification MLP, added to BERT for fine tuning, consists of two fully-connected hidden layers of 512 and 64 perceptrons, and a final 5 units layer that performs multi-class multi-label classification for the five possible labels defined for this challenge. Both hidden layers contain 'swish' activation functions, while the output layer implements 'sigmoid' non-linearities to ensure that it outputs values bounded between 0 and 1. We chose 'swish' as activation function to obtain a smooth transition between the positive and negative sides of the response space of the non-linear projection (Ramachandran et al., 2017). Dropout layers with rate equal to 0.1 were added in between fully-connected layers.

We performed fine tuning of this model on the training set of the MAMI challenge. First, warming up only on the added MLP during 8 epochs. Then, on the full model during 8 more epochs. Both training stages made use of the Adam optimizer (Kingma and Ba, 2014), the first one with initial learning rate of $3e^{-3}$ and the latter with $3e^{-5}$.

2.2.2 Enhanced vocabulary representations

We enriched the text modality description by incorporating word representations for each object present within the images. To this end, we relied on

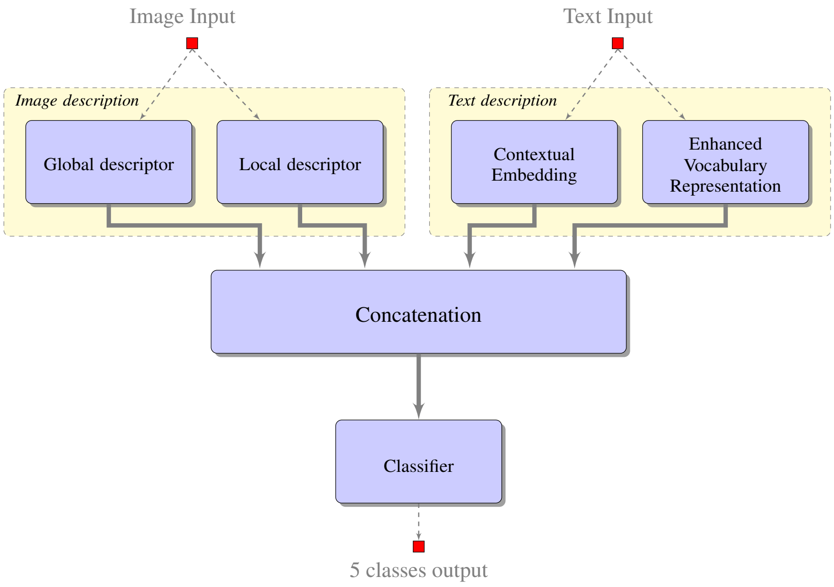


Figure 1: Architecture proposed

the pre-trained VinVL model (Zhang et al., 2021), which is used to segment objects, and provides a list of nouns and adjectives for each segmented object. Some examples of words generated in this stage are: woman, man, red, blue, thin, tall, etc.

This process produces as many lists as there are objects detected within the image. We concatenated all individual words discovered for the same image into a single vector. Then, we used this vector of nouns and adjectives to train a classification network with the same architecture and process as the one explained in sec. 2.2.1, i.e., the architecture and training process are repeated on a different set of parameters.

2.3 Classifier

After the individual training of each of the models described through sections 2.2 and 2.1, we used them in inference mode to process their corresponding inputs, and obtained their respective outputs up to their next-to-last layers. This step produces a 64-D vector for each of the four models.

By concatenating these four representations into a single vector, we produced a multi-modal feature vector of length 256. This resulting vector is used as input for a final MLP classification model, which consists of nine fully-connected layers as shown in Fig. 2.

This final model also uses 'swish' activation functions for all hidden layers, and the 'sigmoid' activation function for its output layer. As shown in Fig. 2, this model is organized in four blocks, each of which is composed by: a regularization process plus two consecutive fully-connected layers. Regularizers are either dropout or batch normalization. Dropout regularizers use a dropout rate of 0.3. Similarly to the previous individual models, this one also uses an output layer of five perceptrons corresponding to each of the five possible classes in the classification task.

2.4 Training

The final classification model was trained using binary cross entropy as loss function, and the Adam optimizer (Kingma and Ba, 2014) with default parameters as implemented in tensorflow: learning rate $\eta = 0.001$, decay for the smoothing of first and second order moments $\beta 1 = 0.9$ and $\beta 2 = 0.999$, and minimum tolerance $\epsilon = 1e{-}7$. We trained this model during 50 epochs with batches of size 64.

We decided to stop training at 50 epoch, as we observed after several attempts that the loss function has consistently converged by then, for both training and validations sets, i.e., no overfitting was observed.

3 Results and discussion

Table 1 shows the accuracy obtained by our model during training and validation, as well as the F1 score obtained on the test set as reported by the server of the MAMI challenge. These scores are presented for Task B (multi-class - multi-label classification) and for three models: the baseline as reported by the organizers of the challenge; our

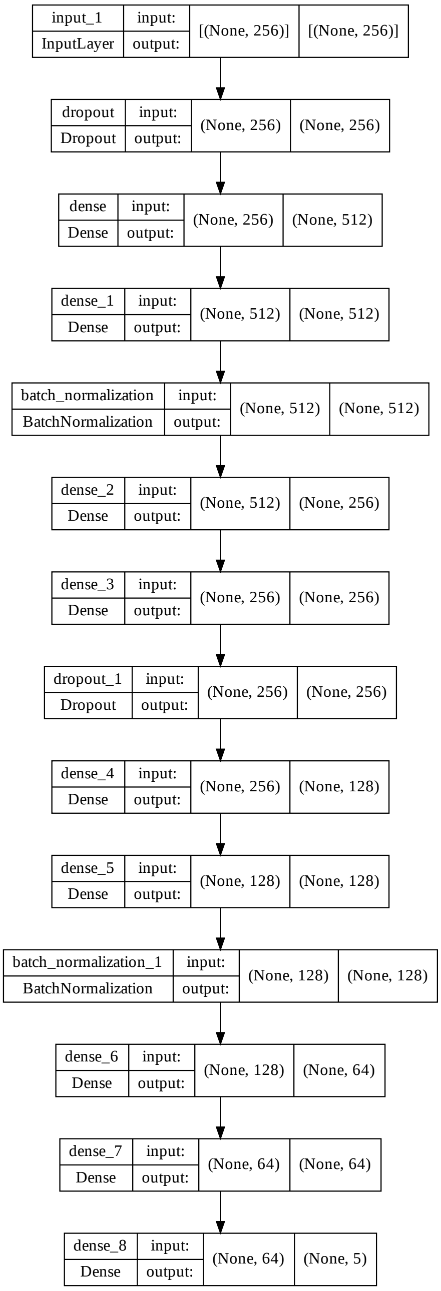

input_1 | InputLayer | input: [(None, 256)] | output: [(None, 256)]
dropout | Dropout | input: (None, 256) | output: (None, 256)
dense | Dense | input: (None, 256) | output: (None, 512)
dense_1 | Dense | input: (None, 512) | output: (None, 512)
batch_normalization | BatchNormalization | input: (None, 512) | output: (None, 512)
dense_2 | Dense | input: (None, 512) | output: (None, 256)
dense_3 | Dense | input: (None, 256) | output: (None, 256)
dropout_1 | Dropout | input: (None, 256) | output: (None, 256)
dense_4 | Dense | input: (None, 256) | output: (None, 128)
dense_5 | Dense | input: (None, 128) | output: (None, 128)
batch_normalization_1 | BatchNormalization | input: (None, 128) | output: (None, 128)
dense_6 | Dense | input: (None, 128) | output: (None, 64)
dense_7 | Dense | input: (None, 64) | output: (None, 64)
dense_8 | Dense | input: (None, 64) | output: (None, 5)

Figure 2: Classification MLP that processes the merged multi-modal feature descriptor.

proposed model; and the best model submitted to the leader board of the competition.

Model	**Training**	**Validation**	**Test (F1)**
Baseline	–	–	0.621
Ours	0.937	0.895	0.646
Best	–	–	0.731

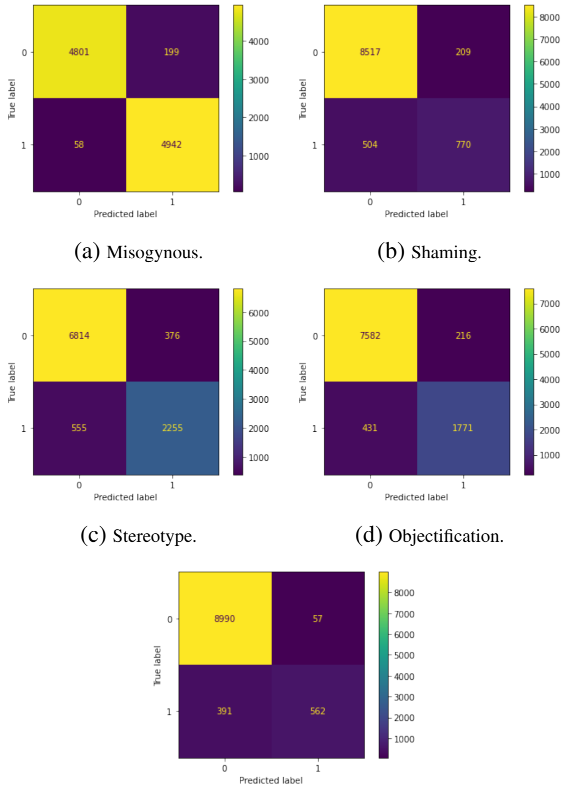
Table 1: Performance on Sub-task B from the baseline model, our proposed model, and the top model reported on the leader board. Columns Training and Validation report accuracy, while column Test reports the F1 score.

Fig. 3 shows the confusion matrices produced by our model on the instances of the MAMI challenge, computed on the joined training and validation sets. We show five confusion matrices because the categories are not mutually exclusive. Each of the matrices contains true negatives [0,0], false positives [0,1], false negatives [1,0], and true positives [1,1]. In all cases, the accuracy is greater than 0.90, with the “misogynous” class having the highest score (0.97) and the stereotyped class the lowest (0.90). Using the F1 measure, the “shaming” and “violence” (the most unbalanced) classes have the worst performance with 0.68 and 0.71 respectively. This situation is due to the small number of true positive instances in these two classes, which might bias the model towards the prediction of the negative label. Meanwhile, the “misogynous”, “stereotype” and “objectification” (the most balanced) classes have a similar performance with an F1 score above 0.82. Moreover, the classes with the lowest performance, have a proportionally much lower number of training examples. This fact limits the fine tuning of the model parameters in the overall training process.

Fig. 4 visualizes the ROC curve, which represents the rate of true positives versus the rate of false positives. As in the confusion matrices, it can be observed that the best performance is obtained in the class “misogynous”, while the worst occurs with the classes “shaming” and “violence”.

4 Conclusions

In this paper, we proposed a framework for extracting lexical-semantic features from text and combine them with semantic descriptions of images in the context of the Multimedia Automatic Misogyny Identification (MAMI) competition 2022. Our results overcame the baseline by 4%, but fell behind the best performance by 12% for Sub-task B. Our model’s performance could be explained by the un-



(a) Misogynous.

(b) Shaming.

(c) Stereotype.

(d) Objectification.

(e) Violence.

Figure 3: Confusion matrices for the 5 classes in the challenge.

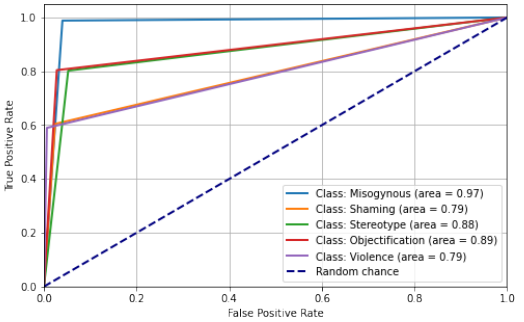


Figure 4: ROC curve for the five classes in the MAMI challenge: "misogynous", "shaming", "stereotype", "objectification", and "violence". "Shaming" and "violence" curves are overlapped.

balanced classes and low number of examples, and, therefore, is limited in this sense, achieving a performance below 0.9 and around 0.7, for accuracy and F1 measure, respectively. Still, for balanced classes we obtained a performance above 0.9 and 0.8, in terms of accuracy and F1 score respectively. As future work we propose three alternatives: 1) To fine tune the classification threshold of the sigmoid output layer of the model, independently for each class, i.e., not all classes need to have 0.5 as classification threshold; 2) Optimize the loss function directly on the F1 score; and, 3) Weight the contribution of each class in the overall loss function during backpropagation.

Aknowledgements

This research has been made possible thanks to generous support from the Consejo Nacional de Ciencia y Tecnología (CONACyT). And the support of the Asociación Mexicana de Cultura A.C.

References

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 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota.

Elisabetta Fersini, Francesca Gasparini, Giulia Rizzi, Aurora Saibene, Berta Chulvi, Paolo Rosso, Alyssa Lees, and Jeffrey Sorensen. 2022. Semeval-2022 task 5: Multimedia automatic misogyny identification. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. In *Proc. 3rd Int. Conf. Learning Representations*.

David G. Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2(60):91–110.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv:1908.08962v2*.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models. In *CVPR 2021*.