UPB @ DANKMEMES: Italian Memes Analysis - Employing Visual Models and Graph Convolutional Networks for Meme Identification and Hate Speech Detection

George-Alexandru Vlad*, **George-Eduard Zaharia***,
Dumitru-Clementin Cercel, Mihai Dascalu
University Politehnica of Bucharest, Faculty of Automatic Control and Computers
{george.vlad0108, george.zaharia0806}@stud.acs.upb.ro
{dumitru.cercel, mihai.dascalu}@upb.ro

Abstract

Certain events or political situations determine users from the online environment to express themselves by using different modalities. One of them is represented by Internet memes, which combine text with a representative image to entail a wide range of emotions, from humor to sarcasm and even hate. In this paper, we describe our approach for the DANKMEMES competition from EVALITA 2020 consisting of a multimodal multi-task learning architecture based on two main components. The first one is a Graph Convolutional Network combined with an Italian BERT for text encoding, while the second is varied between different image-based architectures (i.e., ResNet50, ResNet152, and VGG-16) for image representation. Our solution achieves good performance on the first two tasks of the current competition, ranking 3^{rd} for both Task 1 (.8437 macro-F1 score) and Task 2 (.8169 macro-F1 score), while exceeding by high margins the official baselines.

1 Introduction

During the past two decades, the Internet evolved massively and the social web became a hub where people share their opinions, cooperate to solve issues, or simply discuss on various topics. There are many ways in which users can express themselves: plain text, videos, or images. The latter option became widely used due to its convenience; however, images are frequently accompanied by a short text description to better convey information. As the Internet and the online social interactions evolved, certain image templates emerged and gained global popularity, contributing to a *de facto* standardization of joint text-image usage, and thus leading to the creation of memes. Memes can be humorous, satirical, offensive, or hateful, therefore encapsulating a wide range of emotions and beliefs. Properly identifying memes from non-memes, and then analyzing them to detect the users' intentions is becoming a stringent task in online marketing campaigns by targeting the automated identification of opinions pertaining to certain groups of users.

The DANKMEMES competition [22] from EVALITA 2020 [19] challenged participants to approach the previously mentioned issues by creating systems that identify and analyze Internet memes in Italian. The competition consists of three tasks, out of which we tackled two. Task 1 - *Meme Detection* considers the identification of memes from a collection of images, such that a clear distinction can be made between memes and ordinary images. Afterwards, Task 2 - *Hate Speech Identification* targets the classification of images in terms of their purpose, by analyzing content and identifying whether images are hateful or not.

2 Related Work

2.1 Multimodal Fake News Detection

Singhal et al. [16] employed the usage of multimodal techniques for fake news detection. The authors introduced SpotFake, an architecture divided into three sub-parts: one for identifying textual features using Bidirectional Encoder Representations from Transformers (BERT) [10], a second for visual analysis based on VGG-19 [5], while the third combines the previously mentioned elements into a single feature vector.

Similarly, Shah and Priyanshi [23] performed

*These authors contributed equally.

multimodal fake news detection by using two separate channels, visual and textual, both of them aiming to extract relevant features. Moreover, they included a Cultural Algorithm that introduces another dimension by employing situational knowledge, i.e. information about the depicted event as seen by a specific individual. Another approach regarding fake news detection was introduced by Khattar et al. [12] who created MVAE, a multimodal autoencoder including encoders (both visual and textual), decoders, and a detection module for classifying the inputs.

2.2 Multimodal Hate Speech Identification

Kiela et al. [20] created a new dataset specifically designed for identifying hateful speech in memes. At the same time, the authors also introduced a series of baselines for further comparison, including ResNet-152 [7] and VilBERT [13] for the visual channel, and BERT for the textual counterpart.

Furthermore, Sabat et al. [15] tackled the problem of hate speech identification in memes by also employing a multimodal system. However, they used an Optical Character Recognition system for extracting the textual component from the inputs, alongside visual features from a VGG-16 component and the text encoded with BERT.

3 Method

Our approach for both tasks consists of a multi-task learning technique [1] and our architecture consists of two main neural network components, one for the text input, while the other for the image input. Thus, we combined the outputs of these two components and used the learned features for determining the required class, either for Task 1 or Task 2.

3.1 Corpus

The dataset for the meme detection task is split into two parts, train and test. The training dataset contains 1,600 image entries, together with a CSV file containing other useful metadata, such as: the engagement (i.e., number of comments and likes), date, and manipulation (i.e., binary coding denoting the low/high level of image modifications), alongside a transcript of the text present in the image. We kept 85% of the entries for training, while 15% are used for validation; the same class distribution is kept in both partition. The test dataset for the first task contains 400 entries with a corresponding CSV file of a similar structure. The second task offers a dataset containing 800 entries which was partitioned in a similar manner.

3.2 Image Component

Several image-based neural networks were considered for the first component of our final architecture. First, we used VGG-16 which consists of five stacks of Convolutional Neural Networks [4] accompanied by max-pooling layers. Pretrained weights on the ImageNet dataset [3] were afterwards fine-tuned. Second, we also experimented with ResNet in two variants, ResNet50 and ResNet152. ResNet introduced the concept of skip connections as a solution to the vanishing gradient problem; as such, the networks could be further scaled in terms of depth, enabling more abstract high-level features to be extracted from the input images. Similar VGG-16 architecture, pretrained weights on ImageNet were fine-tuned for ResNet152, whereas pretrained weights on VGGFace2 [9] were used for ResNet50.

3.3 Text Component

A Graph Convolutional Network (GCN) [18] for representing long-term dependencies between tokens was selected, alongside a pretrained version of BERT for Italian (ItalianBERT)[1] to model the contextual information at sample level. The underlying implementation of the textual feature extractor follows the architectural design of Vocabulary Graph Convolutional Network with BERT (VGCN-BERT) [21].

The proposed architecture (VGCN-ItalianBERT) uses a tight coupling between the graph convolutional layers and the ItalianBERT embeddings, enabling the model to better adjust the GCN extracted features through ItalianBERT's attention mechanism. The input to the VGCN layer is represented by a vector $X_{d,v}$, where *d* is the dimension of the ItalianBERT embedding and *v* is the number of tokens in the dataset vocabulary. A symmetric adjacency matrix $A_{v,v}$ is built to preserve the prior global relationship between tokens, where v is the vocabulary dimension. The edge weight between two nodes *i, j*, denoted as $A_{i,j}$, is initialized with the normalized point-wise mutual information (NPMI) value [2] between the two vocabulary tokens *i, j*. The mechanism of

[1] https://github.com/dbmdz/berts#italian-bert

the VGCN layer is formally summarized by the following equations:

$$H_{v,h} = Dropout(\widetilde{A}_{v,v} W_{v,h}) \quad (1)$$

$$H_{d,h} = ReLU(X_{d,v} H_{v,h}) \quad (2)$$

$$H_{d,g} = H_{d,h} W_{h,g} \quad (3)$$

where terms $W_{v,h}$ and $W_{h,g}$ represent the weights of the two GCN internal layers, with *v* the vocabulary dimension, *h* and *g* the output feature dimensions. In Equation 1, we add the global context by multiplying the normalized adjacency matrix $\widetilde{A}$ with the weight matrix of the first GCN layer. We use the normalized adjacency matrix $\widetilde{A} = D^{-1/2} A D^{-1/2}$ to ensure numerical stability. A convolution between the input vector $X_{d,v}$ and the result from the previous operation (Equation 2) is performed to combine the global information with the ItalianBERT embeddings. Lastly, Equation 3 projects the features to the dimensions required to fill in the reserved VGCN-ItalianBERT embedding slots.

Visual text features describing the actors of a meme are added as the pair sentence to ItalianBERT's input. We cap the second sentence containing the visual text features to *K* tokens, overflowing tokens being dropped. Considering *L* the maximum number of input tokens, the remainder of $L - K$ tokens are being split between the text tokens associated with a meme and *G* VGCN reserved slots. Those slots are kept empty to be internally filled with VGCN embeddings during training. Alongside ordinary inputs required by ItalianBERT (i.e. *input ids*, *input masks* and *segment ids*), we build a *gcn ids* vector similarly to *input ids*, by mapping each unique input token to the corresponding index in the task vocabulary V_{task}; V_{task} represents the set of tokens available in the task text corpus and in the ItalianBERT's vocabulary. The second additional input is represented by a binary mask vector having the value of 1 for the VGCN reserved tokens, and 0 otherwise. During training, all ItalianBERT layers with the exception of the last 4 encoder blocks were frozen.

3.4 Multimodal Architecture

The final solution consists of a multimodal architecture with two main components, each specialized on processing one informational channel, namely text or image-based. The dates are segmented and encoded by using complementary sine and cosine functions to preserve the cyclic characteristics of days (in a month) and months. Equation 4 describes the time cyclical encoding procedure, where *n* represents the day value subtracted by 1 and divided by the number of days in the corresponding month. The same operations are applied for the months encoding over the month index, but the denominator is 12 in this case. Additional metadata (i.e., manipulation and engagement) was also encoded and used in the final prediction. Values representing the year and engagement were normalized to ensure the model's stability during training.

$$\begin{aligned} \theta &= 2 * \pi * n \\ time_{sin} = sin(\theta); time_{cos} &= cos(\theta) \end{aligned} \quad (4)$$

The two feature vectors from the image and text components were fused together by concatenation into a single vector and passed through two fully connected layers, followed by a dropout layer of 0.5. The output of the dropout layer is then concatenated together with the other extracted features like time, engagement, manipulation, and fed to the output layer. Softmax activation function is used over the last fully connected layer to compute the distribution probability over the task classes. L2 regularization kernel is used on the two hidden layers before fusion to account for large activations and to keep our output layer sensible to the metadata encoded features.

In addition, an *ensemble*-based architecture using our ResNet50 + VGCN-ItalianBERT model was also considered. First, the training dataset was split into 5 sets, while preserving the class distribution of each fold. The aforementioned model was trained 5 times using 4/5 sets for training, and the remainder set for validation. A weighted voting procedure is performed at prediction time, in which the weights are represented by the average confidence score of the voters in the class receiving the highest probability after softmax. Thus, we advocate for higher confidence scores over the number of voters in choosing the predicted class.

3.5 Experimental Setup

Preprocessing steps were performed to feed the datasets to our architecture. The texts were tokenized using the ItalianBERT tokenizer, and

then the *input ids*, *input masks*, *segment ids*, *gcn ids* and *gcn masks* were computed. Images were resized to a uniform dimension (i.e., 448 x 448) and were serialized alongside the text components in a *tfrecords* file specific for Tensorflow [6]. An Adam Weight decay optimizer [8] with a learning rate of *1e-5* and a weight decay rate of 0.01 were used in all conducted experiments. Furthermore, the warm up proportion was set to 0.1.

The maximum input length was limited to $L = 100$ tokens and the *Visual* text features to $K = 20$ tokens as the textual channel of memes is represented by short sentences. Following the experimental setup described in [21], we reserve $G = 16$ slots to be filled with the resulted VGCN-ItalianBERT embeddings. Moreover, only NPMI values larger than 0.3 are kept in the adjacency matrix A, corresponding to a higher semantic correlation between words; all the other values below this threshold are set to 0.

We empirically found 1e-5 to be a good learning rate value, which is on par with the results of [21]. Lastly, we choose to train all the models for 9 epochs with a batch size of 8 examples.

3.6 Results

Table 1 contains the results obtained by our models for the first two tasks of the DANKMEMES competition. The components that were frozen during the training process are varied for the three main conducted experiments (i.e. combining ItalianBERT with VGCN and ResNet50, ResNet152 and VGG-16, respectively) to identify proper adjustments for the weights of the pretrained models. The best results among the four evaluated sets (i.e. validation, test for Task 1 and validation, test for Task 2) are obtained by either freezing only the VGCN-ItalianBERT component or by freezing both textual and image components. The necessity of freezing the text branch of the architecture underlines the fact that the pretrained weights for the ItalianBERT model already properly capture specific traits of Italian and prove to be a viable option, even when analyzing short texts such as memes. Furthermore, the last convolutional block of the image component needs to be unfrozen because training an architecture on potential meme images is a more specific task when compared to analyzing Italian text.

The best results are obtained using variations of the ResNet50 + VGCN-ItalianBERT model, with an .9041 macro-F1 score for the custom validation dataset used for Task 1, and .8745 and .8169 macro-F1 scores on the validation and test datasets for Task 2. However, the best result for the Task 1 test set is yielded by the ResNet152 + VGCN-ItalianBERT architecture, with an .8700 macro-F1 score.

ItalianBERT, ResNet50, and ResNet50 + ItalianBERT are used as baseline models to explore the improvements made by adding VGCN to the textual architecture while maintaining the same experimental setup. As expected, the model using only the textual channel (i.e. ItalianBERT baseline model) is performing considerably worse than the joint architecture ResNet50 + ItalianBERT, thus arguing for the importance of considering images in disambiguating the textual input. The ResNet50 + VGCN-ItalianBERT model performs consistently better than its baseline counterpart (i.e., ResNet50 + ItalianBERT), by obtaining improvements of 2.92% and 3.35% macro-F1 score on the validation sets for Task 1 and Task 2, respectively.

3.7 Error Analysis

Although the models performed arguably well on both task, the identified misclassifications represent a good starting point for further analysis and improvement. Figure 1 depicts a series of misclassified entries from both tasks.

The short texts encountered in memes require in several situations prior information on the sociopolitical context, therefore making the detection of memes an exceedingly difficult task. In general, a few well known and highly popular image templates are reused, by changing or partially adjusting the text to expressively convey an idea or a view on a certain subject. However, the used templates in the current competition are extensively customized and tailored specifically to the political context of Italy. In addition, the subjectivity of the annotators also plays a decisive role, considering that the concept of the hateful speech tag for the second task is not well defined for all situations and can be interpreted differently.

4 Conclusion and Future Work

This paper introduces our multimodal architecture for the first two tasks of the DANKMEMES competition from EVALITA 2020. Several

Table 1: Macro-F1 scores on the validation and test datasets, for both Task 1 and Task 2. Submitted models are shown in italics.

Neural Architecture	Frozen Component		Task 1		Task 2	
	Image	Text	Dev	Test	Dev	Test
ItalianBERT	-	-	0.7618	0.7546	0.8083	0.7996
ResNet50	-	-	0.8203	0.7899	0.5661	0.5598
ResNet50 + ItalianBERT	-	✓	0.8749	0.8499	0.8331	0.7949
ResNet50 + VGCN-ItalianBERT	-	-	0.8666	0.8348	0.8413	0.8150
ResNet50 + VGCN-ItalianBERT	-	✓	**0.9041**	0.8235	0.8666	**0.8169**
ResNet50 + VGCN-ItalianBERT	✓	-	0.8874	0.8375	0.8493	0.7584
ResNet50 + VGCN-ItalianBERT	✓	✓	0.8833	0.8499	**0.8745**	0.7992
ResNet152 + VGCN-ItalianBERT	-	-	0.8458	0.8424	0.8331	0.7998
ResNet152 + VGCN-ItalianBERT	-	✓	0.8791	**0.8700**	0.8666	0.7994
ResNet152 + VGCN-ItalianBERT	✓	-	0.8246	0.8474	0.8310	0.8093
ResNet152 + VGCN-ItalianBERT	✓	✓	0.8915	0.8273	0.8489	0.7490
VGG-16 + VGCN-ItalianBERT	-	-	0.8124	0.7923	0.6906	0.5478
VGG-16 + VGCN-ItalianBERT	-	✓	0.8083	0.7620	0.5566	0.5469
VGG-16 + VGCN-ItalianBERT	✓	-	0.7485	0.7447	0.6414	0.5263
VGG-16 + VGCN-ItalianBERT	✓	✓	0.7621	0.7248	0.6003	0.5388
Ensemble Architecture	-	-	0.8916	0.8437	0.7874	0.7692
Competition Baselines	-	-	-	0.5198	-	0.5621

a) Task 1 - not a meme, classified as meme

b) Task 1 - meme, classified as not meme

c) Task 2 - not hateful speech, classified as hateful speech

d) Task 2 - hateful speech, classified as not hateful speech

Figure 1: Examples of misclassified samples for both tasks.

joint text - Vocabulary Graph Convolutional Network alongside an Italian BERT model - and image-based architectures - ResNet50, ResNet152, VGG-16 - were experimented. The consideration of meme meta-information, such as cyclic temporal characteristics and post engagement, boosted even further our F1-scores when compared to the competition baseline.

In terms of future work, we intend to experiment with other visual architectures, including VGG-19 [5] and EfficientNet [17], and also with multilingual neural networks, such as mBERT [14] and XLM-RoBERTa [11], that will empower transfer learning across meme datasets in different languages.

References

[1] Rich Caruana. “Multitask learning”. In: *Machine learning* 28.1 (1997), pp. 41–75.

[2] Gerlof Bouma. “Normalized (pointwise) mutual information in collocation extraction”. In: *Proceedings of GSCL* (2009), pp. 31–40.

[3] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and*

Pattern Recognition. Ieee. 2009, pp. 248–255.

[4] Yoon Kim. "Convolutional neural networks for sentence classification". In: *arXiv preprint arXiv:1408.5882* (2014).

[5] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[6] Martin Abadi et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems". In: *arXiv preprint arXiv:1603.04467* (2016).

[7] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[8] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[9] Qiong Cao et al. "Vggface2: A dataset for recognising faces across pose and age". In: *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE. 2018, pp. 67–74.

[10] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[11] Alexis Conneau et al. "Unsupervised cross-lingual representation learning at scale". In: *arXiv preprint arXiv:1911.02116* (2019).

[12] Dhruv Khattar et al. "Mvae: Multimodal variational autoencoder for fake news detection". In: *The World Wide Web Conference*. 2019, pp. 2915–2921.

[13] Jiasen Lu et al. "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks". In: *Advances in Neural Information Processing Systems*. 2019, pp. 13–23.

[14] Telmo Pires, Eva Schlinger, and Dan Garrette. "How multilingual is Multilingual BERT?" In: *arXiv preprint arXiv:1906.01502* (2019).

[15] Benet Oriol Sabat, Cristian Canton Ferrer, and Xavier Giro-i-Nieto. "Hate Speech in Pixels: Detection of Offensive Memes towards Automatic Moderation". In: *arXiv preprint arXiv:1910.02334* (2019).

[16] Shivangi Singhal et al. "SpotFake: A Multi-modal Framework for Fake News Detection". In: *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*. IEEE. 2019, pp. 39–47.

[17] Mingxing Tan and Quoc V Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *arXiv preprint arXiv:1905.11946* (2019).

[18] Liang Yao, Chengsheng Mao, and Yuan Luo. "Graph convolutional networks for text classification". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 7370–7377.

[19] Valerio Basile et al. "EVALITA 2020: Overview of the 7th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian". In: *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*. Ed. by Valerio Basile et al. Online: CEUR.org, 2020.

[20] Douwe Kiela et al. "The Hateful Memes Challenge: Detecting Hate Speech in Multimodal Memes". In: *arXiv preprint arXiv:2005.04790* (2020).

[21] Zhibin Lu, Pan Du, and Jian-Yun Nie. "VGCN-BERT: Augmenting BERT with Graph Embedding for Text Classification". In: *European Conference on Information Retrieval*. Springer. 2020, pp. 369–382.

[22] Martina Miliani et al. "DANKMEMES @ EVALITA2020: The memeing of life: memes, multimodality and politics". In: *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*. Ed. by Valerio Basile et al. Online: CEUR.org, 2020.

[23] Priyanshi Shah and Ziad Kobti. "Multimodal fake news detection using a Cultural Algorithm with situational and normative knowledge". In: *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE. 2020, pp. 1–7.