Hateful meme detection with multimodal deep neural networks

Mihai Gabriel Constantin*†, Dan-Ștefan Pârvu*, Cristian Stanciu*, Denisa Ionașcu*, Bogdan Ionescu*
*AI Multimedia Lab, University Politehnica of Bucharest, Romania
†Email: mihai.constantin84@upb.ro

***Abstract*— The modern advances of social media platforms and content sharing websites led to the popularization of *Internet memes*, and today's Internet landscape contains websites that are predominantly dedicated to meme sharing. While at their inception memes were mostly humorous, this concept evolved and nowadays memes cover a wide variety of subjects, including political and social commentaries. Considering the widespread use of memes and their power of conveying distilled messages, they became an important method for spreading hate speech against individuals or targeted groups. Given the multimodal nature of Internet memes, our proposed approach is also a multimodal one, consisting of two parallel processing branches, one textual and one visual, that are joined in a final classification step, providing prediction results for the samples. We test our approach on the publicly available Memotion 7k dataset and compare our results with the baseline approach developed for the dataset.**

I. Introduction

The idea of *memes* is not new, being coined in 1976 [1] when it is used to describe the way cultural ideas spread among the population, similar to gene mutations. However, as the Internet evolved, so did this concept, and today, according to the Oxford Dictionary, it can mostly be defined as: "An image, video, piece of text, etc., typically humorous in nature, that is copied and spread rapidly by Internet users, often with slight variations". Measurements presented in state-of-the-art literature show that memes can contain different degrees of harmful content, including but not limited to racism and aggression [2] and extreme political and social propaganda [3].

As a particularity, some of the most viral memes tend to contain both images and text, making them hard to process by automated systems. Given the inherently multimodal content, any automatic hate speech detection would have to take into account both text and visual information in order to correctly represent and predict the presence of hateful speech. On the other hand, both these types of information must be jointly processed, as the text or the visual content, by themselves, may not represent hateful content, but when taken together their message changes. Such examples are presented in Figure 1, containing images extracted from the Facebook Hateful Memes Challenge and Dataset [4]. While texts like "Love the way you smell today" cannot be considered hateful, when imposed over the image of a skunk, it can be considered hateful or aggressive speech. While these are only a few examples, they do represent the complexity of this type of problem.

Fig. 1. Sample images extracted from the Facebook Hateful Memes Challenge and Dataset [4].

Interest in the community for automatic methods that would help detect such content is at an all-time high. Several datasets have been released, that take into account many approaches and types of hateful content, including the Facebook Hateful Memes Challenge and Dataset [4] and the Memotion 7k [5] and many of them also include private or public benchmarking competitions. Another interesting community project in this direction is the Code-Against-Hate hackathon [6], that tasks participants to create concepts of automatic tools that "nudge" [7] users towards a desired result instead of outright banning or deleting content.

In recent works, many papers deal with the prediction of hateful content, either from a unimodal perspective, usually by using traditional features [8] or from a multimodal, deep-learning perspective [9]. For a thorough overview of these methods, several literature review papers are available [10], [11].

Our work proposes an automatic end-to-end architecture that can analyze both text and visual information extracted from Internet memes. For text processing we employ an architecture that uses attention and bi-directional Long short-term memory (LSTM) layers, while images are processed in a convolutional architecture. The outputs of these layers are concatenated and processed by several fully connected layers. The rest of the paper is structured as follows. Section II presents the proposed method, analyzing the individual components, and Section III presents the experimental setup and results.

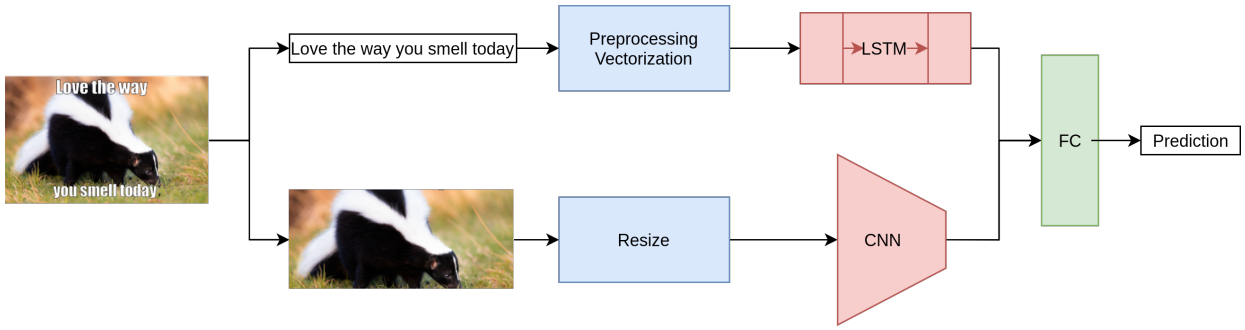

Fig. 2. General diagram of the proposed architecture, presenting the text and visual processing branches that contain the preprocessing steps and the deep processing layers, and the final prediction branch, containing the fully connected layers.

Section IV concludes the paper and identifies some future research directions.

II. Proposed Method

Our proposed method processes text and visual information on two separate processing branches of an end-to-end deep neural network architecture, uniting the results of the processing branches and producing the prediction results with a final fully connected layer architecture. The outline of this approach is presented in Figure 2.

A. Text processing

There have been multiple approaches to perform sentiment analysis in Natural Language Processing. Some of the most successful attempts were accomplished by translating words into numerical representations, which are stored in an embedding matrix and then passed through a recurrent neural network [12], [13].

The text processing branch uses an initial preprocessing step that adapts the input text for the LSTM network. Several actions are taken by the text preprocessor that include: (i) text conversion, (ii) removing unnecessary information, (iii) tokenization, (iv) lemmatization, and (v) vectorization.

In the text conversion phase, all words from the text are transformed into lower case words, in order to lower the degree of complexity associated with text processing. The unnecessary information that is removed from the text is composed of punctuation, URLs, stop words, numbers and emojis. The Natural Language Toolkit (NLTK) [14] Python library is used as a dictionary for common stop words, along with a custom list of stop words created by us, containing typical Internet slang that the NLTK dictionary does not contain. The tokenization phase is a preprocessing step that transforms the given text into tokens, while Lemmatization consists of reducing words to a root form, therefore giving all the related words the same form. Both these steps are performed with the help of the spaCy [15] Python library. Finally, the vectorization phase is carried out with the help of a word2vec approach [16].

The model architecture we implemented is a bi-directional LSTM neural network, with a vanilla attention layer as described in [17]. The LSTM-based text processing architecture takes the input created by the preprocessing steps, and, after creating an embedding lookup table, passes the text through an attention layer, followed by two bi-directional LSTM [18] layers. While the attention layer has the size of the created lookup table, the LSTM layers have the same number of units as the initial size of the input vector. The output of this architecture is a uni-dimensional vector of size 64, that can be concatenated and further processed with the outputs of the visual processing stage. This architecture is presented in Figure 3.

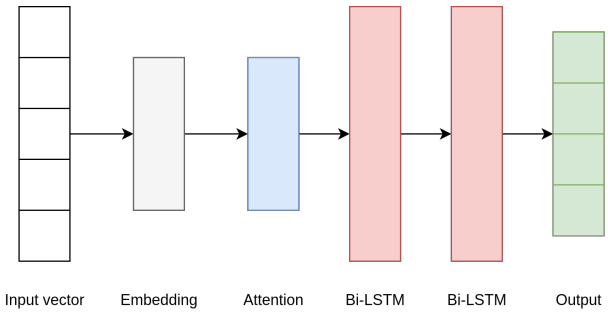

Fig. 3. Text processing architecture, presenting the Embedding lookup table, Attention and Bi-LSTM layers.

B. Visual processing

Image processing has been dominated by convolutional approaches, since the success registered by the AlexNet network [19]. Employing a convolutional approach in our case would imply using mainly the convolutional layers, in order to create an output that can be concatenated with the output created by the text processing branch and processed by the final set of fully connected layers.

The image preprocessing step for our proposed method consists of fewer steps, only resizing the image to the appropriate size that corresponds to the input of our convolutional image processing network. The backbone of the convolutional networks is represented by the ResNet architecture [20], a popular image processing network that previously achieved top results in image classification tasks.

The basic building block of this type of network is a traditional residual block, presented in Figure 4. The network

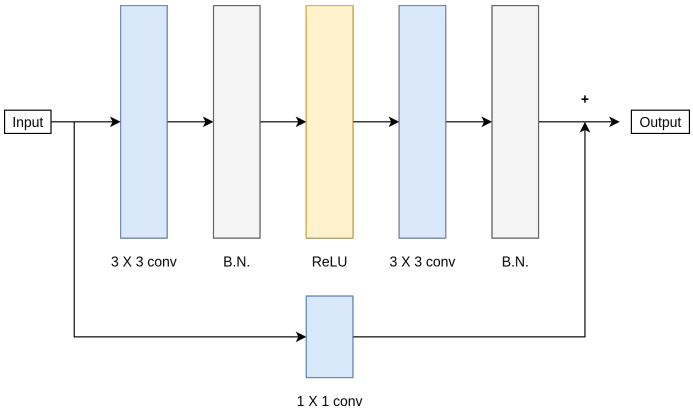

Fig. 4. The residual block of the visual processing architecture, presenting the different sizes of the convolutional layers, the batch normalization and ReLU layers.

TABLE I

RESULTS ON THE MEMOTION 7K DATASET [5]. WE PRESENT THE VARIATIONS OF THE ARCHITECTURE, AND COMPARE OUR RESULTS WITH THE BASELINE DNN PROPOSED BY THE AUTHORS OF THE DATASET. BEST MACRO F1 (MF1) METRIC RESULTS ARE PRESENTED IN BOLD.

Conv	Width	Height	MF1
ResNet-34	3	512	0.2225
ResNet-34	5	512	**0.2279**
ResNet-34	7	512	0.2116
ResNet-34	9	512	0.2037
ResNet-34	5	256	**0.2381**
ResNet-34	5	512	0.2279
ResNet-34	5	1024	0.2073
ResNet-18	**5**	**256**	**0.2498**
ResNet-34	5	256	0.2381
ResNet-50	5	256	0.2310
ResNet-101	5	256	0.1871
Baseline DNN [5]			0.2301

is created by adding residual blocks to the network and linking them in a sequential manner. Finally, in order to create a one-dimensional output for this architecture, the output of the final residual block is processed by an AvgPool layer. Based on the evolution of the traditional ResNet approaches, several types of ResNet backbone networks are tested, mainly differentiated by the number of residual layers employed, namely ResNet-18, ResNet-34, ResNet-50 and ResNet-101.

C. Prediction processing

As presented in Figure 2, the outputs of the text and visual processing architectures are concatenated inside the end-to-end architecture and are processed by a series of fully connected layers that have the role of learning the patterns in the input space in order to create accurate predictions. Several variations of the prediction processing branch are tested, by varying the network width and height. For height variations, we changed the number of neurons in each of the fully connected layers, testing values of $H \in \{256, 512, 1024\}$, while for the network width we changed the number of layers in the network, testing values of $W \in \{3, 5, 7, 9\}$.

D. Training protocol

The training process is carried out in an end-to-end manner, starting a new training run for each of the variations presented in the previous sections. We employ an Adam optimizer [21], with an initial learning rate of 0.01, and binary crossentropy as the loss function. For each network variant, training is done for 50 epochs, with a batch size of 16 samples.

III. EXPERIMENTAL RESULTS

Experimentation is carried out on the Memotion 7k dataset [5], a dataset that contains labels for different categories of memes, classifying offence, humor, sarcasm, and motivation. The dataset contains a total of 6992 samples, that we split by means of a random stratified k-fold approach, generating 80% training data (5593 memes) and 20% testing data (1399 memes). For our experiments we used the four classes of offensive content, as described by the dataset authors in Task 3: “not offensive”, “slightly offensive”, “very offensive” and “highly offensive”, with non-hateful content representing 36% of the data.

For testing and comparing our results we used the Macro F1 metric (MF1), and used the baseline score presented by the creators of the dataset as a measure of performance ($MF1 = 0.2301$). The baseline results are achieved by the creators of the dataset with a deep architecture that processes the text information via 1-D convolutional and LSTM layers, and visual information via a VGG-16 [22] architecture. Finally, it is worth mentioning that the organizers provide text information for each of the memes in the dataset, therefore no optical character recognition (OCR) module is needed in our experiments. In a real world environment however, such a module would be necessary and the final result of the prediction network would also depend on the recognition accuracy of the OCR module.

A. Network variations

As we presented in Section II, several network variants are tested, using different ResNet backbone architectures (ResNet-18, Resnet-34, ResNet-50, and ResNet-101), different network width for the prediction branch (3, 5, 7, and 9 layers), and different network height for the prediction branch (256, 512, and 1024 neurons per layer). The results are presented in Table I. For each parameter the best variants have the MF1 results presented in bold, while the best overall performing architecture is presented entirely in bold.

It is interesting to note that the majority of best performing variants tend to have a lower dimensionality. As presented in the results table, the best performing architecture uses a ResNet-18 backbone, 5 fully connected layers and has 256 neurons per layer. Furthermore, there is an observable trend for low results when the network is at its largest. Using 9 fully connected layers brings the results down to 0.2037, using 1024 neurons per layer 0.2073 and ResNet-101 architecture 0.1871.

While these observations may seem somehow counter-intuitive, as one would expect larger networks to perform better, we theorize that this behaviour may occur due to the

low number of training samples in the dataset (5593 samples). Using a larger dataset may change this trend.

B. Final results

As shown in Table I, our best performing system achieves a MF1 score of 0.2498, representing an increase of 8.24% over the provided DNN baseline. While this Macro F1 score may not seem too high, the fact that it surpasses the proposed baseline indicates that it represents an accurate depiction of the current possibilities under this particular setting. Furthermore, while tasks with an objective ground truth value are starting to show near-perfect machine learning methods of prediction, tasks that have a certain degree of subjectivity in them are more complex. The prediction of hateful content is, by its nature, a subjective property of multimedia content, as annotators would not always agree on what is and is not hateful. This subject is touched upon in some of the literature review papers that deal with hateful content detection [11].

Several other well known tasks that seek to predict and analyze subjective properties of multimedia data have also shown results that are far from near-perfect performance. One example in this case would be represented by the prediction of media interestingness [23]. As presented during the 2017 MediaEval Predicting Media Interestingness task [24], systems submitted by participants to this task rarely score above 0.3 with regards to the official metric (mean average precision). However it is interesting to note that in many such tasks results tend to grow each year, indicating that participants start to better understand and tune their systems to the concept that is being analyzed.

IV. Conclusions

In this work we presented our approach for the prediction of hateful and offensive memes. We developed an automatic end-to-end architecture that analyzes both text and visual information extracted from Internet memes. The text processing branch uses attention and bi-directional LSTM layers, while the visual processing branch uses a ResNet-like convolutional architecture. The outputs of these deep architectures are processed in the final stage of the network by a fully connected architecture. Several setups and variants of this end-to-end architecture are proposed. We perform experiments on the Memotion 7k dataset, using a similar dataset setup, data split, and metric as the ones proposed by the creators of the dataset. Final results show that our proposed system performs better than the baseline DNN architecture suggested in the Memotion dataset, improving those results by 8.24%. Future work on the proposed method may include varying the parameters of the text processing branch, testing other backbone architectures for the visual processing branch and testing our methods on larger datasets.

Acknowledgment

This work was funded under project AI4Media “A European Excellence Centre for Media, Society and Democracy”, grant #951911, H2020 ICT-48-2020.

References

[1] R. Dawkins. The selfish gene. Oxford university press, 2016.
[2] IJ Yoon. ”Why is it not just a joke? Analysis of Internet memes associated with racism and hidden ideology of colorblindness.” Journal of Cultural Research in Art Education 33, 2016.
[3] D. Olsen. ”How memes are being weaponized for political propaganda.” Salon, February 24, 2018.
[4] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. arXiv preprint arXiv:2005.04790. 2020.
[5] C. Sharma, W. Paka, D.B. Scott, A. Das, S. Poria, T. Chakraborty, and B. Gambäck. Task report: Memotion analysis 1.0@ semeval 2020: The visuo-lingual metaphor. In Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2020), Barcelona, Spain, Sep. Association for Computational Linguistics. 2020.
[6] Code Against Hate 2. https://www.codeagainsthate.eu/. Accessed on 17-July-2021.
[7] C. Schneider, M. Weinmann, and J. Vom Brocke. Digital nudging: guiding online user choices through interface design. Communications of the ACM, 61(7), 67-73. 2018.
[8] M. Mozafari, R. Farahbakhsh and N. Crespi. A BERT-based transfer learning approach for hate speech detection in online social media. In International Conference on Complex Networks and Their Applications (pp. 928-940). 2019.
[9] L. Li, Y.C. Chen, Y. Cheng, Z. Gan, L. Yu and J. Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. arXiv preprint arXiv:2005.00200. 2020.
[10] P. Fortuna and S. Nunes. A survey on automatic detection of hate speech in text. ACM Computing Surveys (CSUR), 51(4), 1-30. 2018.
[11] T.H. Afridi, A. Alam, M.N. Khan, J. Khan and Y.K. Lee. A Multimodal Memes Classification: A Survey and Open Research Issues. arXiv preprint arXiv:2009.08395. 2020.
[12] O. Irsoy, and C. Cardie. Bidirectional recursive neural networks for token-level labeling with structure. arXiv preprint arXiv:1312.0493. 2013.
[13] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 2: Short papers) (pp. 49-54). June, 2014.
[14] S. Bird, E. Klein and E. Loper: Natural language processing with Python: analyzing text with the natural language toolkit. O’Reilly Media, Inc. 2009.
[15] M. Honnibal, I. Montani, S. Van Landeghem and A. Boyd: spaCy: Industrial-strength Natural Language Processing in Python. Zenodo. 2020.
[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. 2013.
[17] J. Liu and Y. Zhang. Attention modeling for targeted sentiment. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers (pp. 572-577). April, 2017.
[18] M. Schuster, and K.K. Paliwal. Bidirectional recurrent neural networks. IEEE transactions on Signal Processing, 45(11), 2673-2681. 1997.
[19] A. Krizhevsky, I. Sutskever, and G.E. Hinton. ”Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems 25: 1097-1105. 2012.
[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778). 2016
[21] D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014.
[22] K. Simonyan and A. Zisserman: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. 2014.
[23] M.G. Constantin, L.D. Ştefan, B. Ionescu, N.Q. Duong, C.-H. Demarty and M. Sjöberg. Visual Interestingness Prediction: A Benchmark Framework and Literature Review. International Journal of Computer Vision, 1-25. 2021.
[24] C.-H. Demarty, M. Sjöberg, B. Ionescu, T.T. Do, M. Gygli and N. Duong. Mediaeval 2017 predicting media interestingness task. In MediaEval workshop. September, 2017.