SNK @ DANKMEMES: Leveraging Pretrained Embeddings for Multimodal Meme Detection

Stefano Fiorucci
Machine learning engineer
ETI3
stefano.fiorucci@virgilio.it

Abstract

English. In this paper, we describe and present the results of meme detection system, specifically developed and submitted for our participation to the first subtask of DANKMEMES (EVALITA 2020). We built simple classifiers, consisting in feed forward neural networks. They leverage existing pretrained embeddings, both for text and image representation. Our best system (SNK1) achieves good results in meme detection (F1 = 0.8473), ranking 2nd in the competition, at a distance of 0.0028 from the first classified.

Italiano. *In questo articolo, descriviamo e presentiamo i risultati di un sistema di individuazione dei meme, ideato e sviluppato per partecipare al primo subtask di DANKMEMES (EVALITA 2020). Abbiamo realizzato dei semplici classificatori, costituiti da una rete neurale feed-forward: essi sfruttano embedding preesistenti, per la rappresentazione numerica di testo e immagini. Il nostro miglior sistema (SNK1) raggiunge buoni risultati nell'individuazione dei meme (F1 = 0.8473) e si è classificato secondo nella competizione, ad una distanza di 0.0028 dal primo classificato.*

1 System description

1.1 General approach and tools

DANKMEMES (Miliani et al., 2020) is a task for meme recognition and hate speech/event identification in memes and is part of the EVALITA 2020 evaluation campaign (Basile et al., 2020).

For our participation to the first subtask of DANKEMES, we built simple classification models for meme detection.

The main challenge is to effectively combine textual and image inputs. We tried to exploit the ability of pretrained embedding to represent the information present in text and images, paying a limited computational cost.

To quickly build various prototypes of neural networks, we used Uber Ludwig framework (Molino et al., 2019): a toolbox built on top of TensorFlow, which facilitates and speeds up the training and testing of various models.

We trained our models using Google Colaboratory, a hosted Jupyter notebook service, which provides free access to GPUs, with some resource and time limitations.

1.2 Features

1.2.1 DANKMEMES dataset

The dataset provided for the first subtask has the following features:

- **File**: the name of the .jpg image file.
- **Date**: when the image has first been posted on Instagram.
- **Picture manipulation**: entails the degree of visual modification of the images. Non-manipulated or low impact changes are labeled 0. Heavily manipulated, impactful changes are labeled 1.
- **Visual actors**: the political actors (i.e. politicians, parties' logos) portrayed visually, regardless whether edited into the picture or portrayed in the original image.
- **Engagement**: the number of comments and likes of the image.
- **Text**: the textual content of the image.

- **Meme**: binary feature, where 0 represents non meme images and 1 meme images. This is the target label.

The dataset also includes **image embeddings**.

1.2.2 Feature selection and preprocessing

We discarded Date feature, because it seems irrelevant for meme detection.

Picture manipulation and Meme are simple binary features and do not require preprocessing.

We chose to scale Engagement feature, using min-max normalization.

Visual actors feature was preprocessed using Ludwig approach for sets. We report an extract of the official framework documentation[1]:

"Set features are expected to be provided as a string of elements separated by whitespace.

The string values are transformed into a binary valued matrix of size n x l (where n is the size of the dataset and l is the minimum of the size of the biggest set and a max_size parameter) [...]

The way sets are mapped into integers consists in first using a tokenizer to map from strings to sequences of set items. Then a dictionary of all the different set item strings present in the column of the dataset is collected, then they are ranked by frequency and an increasing integer ID is assigned to them from the most frequent to the most rare (with 0 being assigned to PAD used for padding and 1 assigned to UNK item)."

1.2.3 Text representation

For text representation, we chose to use pretrained word embeddings for the Italian language.

Our first model used fastText word representations (Bojanowski et al., 2016): non-contextual word embeddings. fastText word embeddings rely on subword information (bag of character n-grams) and thus provide valid representations for rare, misspelled or out-of-vocabulary words. Particularly, we used word vectors for the Italian language officially distributed in 2018 (Grave et al., 2018). Word embeddings are trained on Common Crawl and Wikipedia, using CBOW with position-weights, in dimension 300, with character n-grams of length 5, a window of size 5 and 10 negatives. We calculated the sentence vectors starting from the word vectors and using get_sentence_vector method of fastText python wrapper: each word vector is divided by its L2 norm and then averaged. Obtained sentence vector has dimension 300.

Our second classifier used BERT word representations (Devlin et al., 2018): context-based word embeddings. BERT model uses word-piece tokenization: therefore it too provides embeddings for unseen words. In particular, we used GilBERTo[2], an Italian pretrained language model based on Facebook RoBERTa architecture and CamemBERT text tokenization approach; it was trained with the subword masking technique for 100k steps managing 71GB of Italian text with more than 11 billion words. As an interface for this language model, we used python library HuggingFace's Transformers (Wolf et al., 2019). To obtain sentence vectors, we took the output from the [CLS] token, which is prepended to the sentence during the preprocessing phase and is typically used for classification tasks; undoubtedly, there are also other methods for extracting sentence embeddings from BERT models that may prove more effective. Obtained sentence vector has dimension 768.

1.2.4 Image representation

For image representation, we used the embeddings provided in DANKMEMES dataset. The vector representations are computed employing ResNet (He et al., 2016), a state-of-the-art model for image recognition based on Deep Residual Learning. Every image vector has dimension 2048.

1.3 System architecture

Figure 1 shows a block diagram of system architecture, which is very simple. Picture manipulation, Visual actors, Engagement, Image vector and Sentence vector (obtained from word embedding) were combined by concatenation. The resulting multimodal feature vector was fed as input into a feed-forward neural network with two hidden layers of 256 and 16 neurons respectively, with a ReLU activation function. The last single neuron predicts whether the image is a meme or not.

2 Experiments and results

2.1 Experimental settings

To train our neural networks, we chose cross-entropy loss as the objective function. As defined in the subtask, the metrics of interest are precision, recall and F1 score. In the following, all metrics

[1] https://ludwig-ai.github.io/ludwig-docs/user_guide/#set-features-preprocessing

[2] https://github.com/idb-ita/GilBERTo

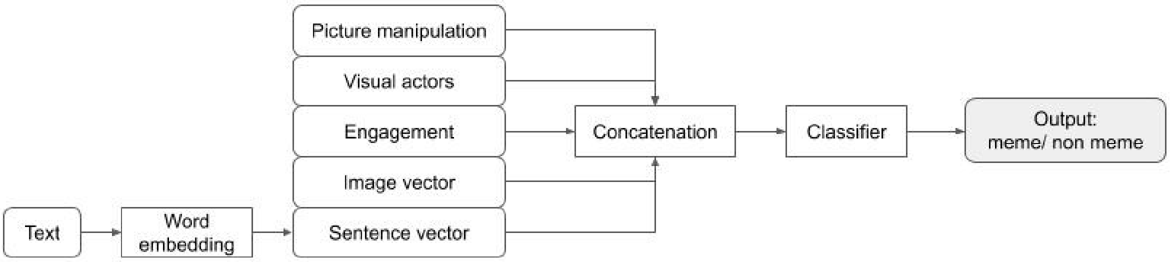

Figure 1: System architecture

reported were calculated using the officially provided evaluation script[3].

We used Adam optimizer with the following parameters: $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\varepsilon = 10^{-8}$. We set an early stop of 5 epochs, in order to avoid overfitting.

Hyperparameter optimization was manually conducted and we tried various combinations of learning rate and batch size: our final models have learning rate of 10^{-5} and batch size of 10.

During our experiments, we studied the impact of a multimodal analysis, compared to using language or vision only.

We trained various models, including different combinations of basic features (Picture manipulation, Visual actors and Engagement), text representation (fastText or GilBERTo) and image representation (ResNet).

2.2 Results

Model	Pr	Re	F1
random baseline	0.525	0.5147	0.5198
Basic Features	**0.8732**	0.6078	0.7168
BF+fastText	0.8253	0.6716	0.7405
BF+GilBERTo	0.7685	0.7647	0.7666
BF+ResNet	0.8341	0.8382	0.8362
BF+fastText+ ResNet (**SNK1**)	0.8515	0.8431	**0.8473**
BF+GilBERTo+ ResNet (**SNK2**)	0.8317	**0.848**	0.8398

Table 1: Experimented models

We observe that basic features are quite informative: the model based only on them far outperforms the random baseline.

Models based on basic features and visual representations perform meme detection well. It should be noted that unimodal vision models perform significantly better than textual models. As Sabat et al. (2019) pointed out, an obvious reason is that the dimensionality of the image representation (2048) is much larger than the linguistic one (fastText: 300; GilBERTo: 768), so it has the capacity to encode more information. It would be interesting to conduct further experiments to investigate less obvious motivations and understand if the image representation actually conveys features of the visual scene, which are specific and distinctive of a meme.

As shown by Beskow et al. (2019), multimodal classifiers are considerably better than textual models and provide some improvement over unimodal vision models, which nevertheless provide solid performance in meme detection.

Team + Run	Pr	Re	F1
A2	0.8522	**0.848**	**0.8501**
SNK1	0.8515	0.8431	0.8473
B2	0.8543	0.8333	0.8437
A1	0.839	0.8431	0.8411
SNK2	0.8317	**0.848**	0.8398
B1	**0.861**	0.7892	0.8235
...			
baseline	0.525	0.5147	0.5198

Table 2: DANKMEMES subtask 1 results table

With reference to the competition, model SNK1 (Basic features + fastText + ResNet) ranked 2nd, at a short distance from the first classified. Model SNK2 (Basic features + GilBERTo + ResNet) ranked 5th.

[3] `https://github.com/gianlucalebani/dankmemes2020`

3 Discussion and conclusion

In this paper, we have presented simple multimodal systems for meme detection, based on a neural network classifier; they leverage existing pretrained embeddings to represent both text and image. Our systems achieve good performance, providing improvements over unimodal classifiers. In the first subtask of DANKMEMES (EVALITA 2020), our models ranked 2nd and 5th.

Based on our experiments, it is observed that pre-trained embeddings can be used effectively and with little effort to represent information conveyed by visual and textual components. While we haven't explicitly included irony or other distinctive aspects derived from text or image among the features, it is understood that the vectors generated by the embeddings express them implicitly.

Starting from the simple model used, it could be interesting to conduct in-depth analyzes to understand which of the basic features are most important. Furthermore, we could build saliency maps (Simonyan et al., 2013) to understand which areas of the images are most relevant for the meme detection task.

The proposed model could be improved. With more time and computational resources, a broader experimentation campaign could be conducted, using Bayesian hyperparameter optimization; we could try different numbers of neurons in hidden layers and other neural network architectures. To improve the classifier without much effort, we could also make an ensemble of our best performing models.

In our classifier, we used BERT powerful language model to get text vectors. We could do BERT fine tuning, in order to obtain better textual embedding, aimed at meme detection task.

Finally, to overcome the limits of this simple model, we could look for a more explicit way to encode the irony present in the text, drawing inspiration from IronITA (Cignarella et al., 2018).

References

Valerio Basile, Danilo Croce, Maria Di Maro and Lucia C. Passaro 2020. *EVALITA 2020: Overview of the 7th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian.* Valerio Basile, Danilo Croce, Maria Di Maro and Lucia C. Passaro (eds.). Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)

David Beskow, Sumeet Kumar and Kathleen Carley 2019. *The Evolution of Political Memes: Detecting and Characterizing Internet Memes with Multimodal Deep Learning* Information Processing & Management, volume 57

Piotr Bojanowski, Edouard Grave, Armand Joulin and Tomas Mikolov 2016. *Enriching Word Vectors with Subword Information* arXiv:1607.04606

Alessandra Teresa Cignarella, Simona Frenda, Valerio Basile, Cristina Bosco, Viviana Patti and Paolo Rosso 2018. *Overview of the EVALITA 2018 Task on Irony Detection in Italian Tweets (IronITA)* Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2018)

Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova 2018. *BERT: Pre-training of Deep Bidirectional Transformers for Language* arXiv:1810.04805

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin and Tomas Mikolov 2018. *Learning Word Vectors for 157 Languages* Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)

Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun 2016. *Deep Residual Learning for Image Recognition* 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Martina Miliani, Giulia Giorgi, Ilir Rama, Guido Anselmi and Gianluca E. Lebani 2020. *DANKMEMES @ EVALITA2020: The memeing of life: memes, multimodality and politics.* Valerio Basile, Danilo Croce, Maria Di Maro and Lucia C. Passaro (eds.). Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)

Piero Molino, Yaroslav Dudin and Sai Sumanth 2019. *Ludwig: a type-based declarative deep learning toolbox* arXiv:1909.07930

Benet Oriol Sabat, Cristian Canton Ferrer and Xavier Giro-i-Nieto 2019. *Hate Speech in Pixels: Detection of Offensive Memes towards Automatic Moderation* arXiv:1910.02334

Karen Simonyan, Andrea Vedaldi and Andrew Zisserman 2013. *Deep inside convolutional networks: Visualising image classification models and saliency maps* arXiv:1312.6034

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest and Alexander M. Rush 2019. *HuggingFace's Transformers: State-of-the-art Natural Language Processing* arXiv:1910.03771