ArchiMeDe: A New Model Architecture for Meme Detection

Jinen Setpal
RN Podar School
Mumbai, India
jinens8@gmail.com
jinen.setpal@rnpodarschool.com

Gabriele Sarti
University of Trieste & SISSA
Trieste, Italy
gsarti@sissa.it

Abstract

English. We introduce ArchiMeDe, a multimodal neural network-based architecture to solve the meme detection subtask of the DankMemes campaign at Evalita 2020. The system incorporates information from visual and textual sources through a multimodal neural ensemble to predict if input images and their respective metadata are memes or not. Each pretrained neural network in the ensemble is first fine-tuned individually on the training dataset, learned text and visual representations are concatenated and the final prediction is performed through majority voting.

Italiano. *Questo articolo presenta ArchiMeDe, un'architettura multimodale basata su reti neurali per la risoluzione del subtask di "meme detection" per lo shared task DankMemes a Evalita 2020. Il sistema unisce informazione visiva e testuale attraverso un insieme multimodale di reti neurali per predirre se immagini in input e i loro metadati corrispondano a meme o meno. Ogni rete neurale preallenata all'interno dell'insieme è inizialmente messa a punto individualmente sul dataset di training; in seguito, le rappresentazioni di ogni rete per immagini e testo vengono concatenate e la previsione finale è effettuata tramite un voto di maggioranza.*

1 Introduction

This paper describes ArchiMeDe, the multimodal system we developed for participating in the DankMemes shared task at the Evalita 2020 campaign. Specifically, we participated in the first subtask of DankMemes, involving meme detection given input images and their metadata. Memes were extracted by task organizers from the Instagram platform, and data available from each dataset entry – actors, user engagement, manipulation, etc. – were leveraged to train an ensemble of multimodal models performing meme detection through majority-vote.

In recent years, the democratization of deep learning approaches in the fields of computer vision and natural language processing is arguably the result of two main factors: first, the cost of AI-dedicated hardware for consumers have been steadily decreasing [Gurbaxani and Mendelson, 1987] and second, the availability of pre-trained open-source models have greatly reduced the computational threshold required to obtain state-of-the-art results in multiple language and vision tasks [Devlin et al., 2019]. Pre-trained systems are often leveraged in a two-step framework: first, they undergo an unsupervised or semi-supervised pre-training to learn general knowledge representations, then they are fine-tuned in a supervised way to adapt their parameters in the context of downstream task. This transfer learning approach stems from the computer vision literature [He et al., 2018] but has been recently adopted for natural language processing tasks with positive results [Howard and Ruder, 2018, Devlin et al., 2019, Liu et al., 2019].

Following recent transfer learning approaches, our system leverages pre-trained visual and word embeddings in a multimodal setup, obtaining strong results on the meme detection subtask.

The following sections present our approach in detail, focusing on the main modules of the system and the features we leverage from the dataset. We then present our results, and conclude by discussing the problems we faced with some inconsistencies in the data.

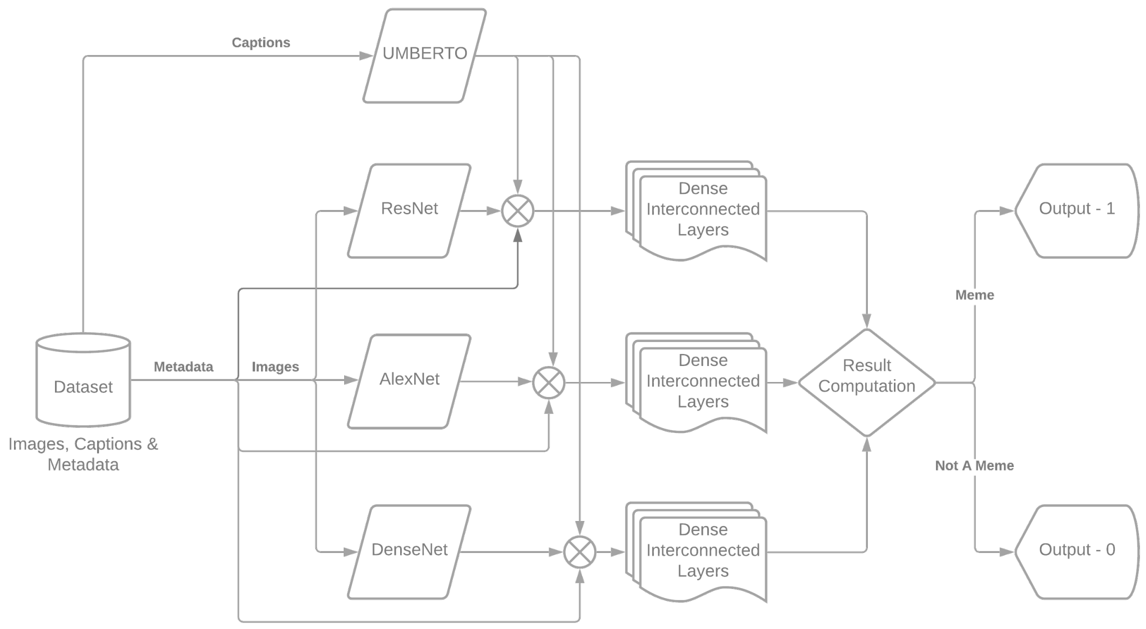

Figure 1: System Architecture

2 System Architecture

The system is based on a 3-model learning ensemble, with the final output being the result of a majority vote. Each individual model incorporates a 8-layer feed-forward neural network taking as input sentence embeddings, image metadata and image vector representations (further discussed in Section 3.1.6) to predict the meme status of an image.

The system incorporates a series of dense layers, with a single dropout layer used to prevent overfitting and improve generalization performance. Multiple experimental iterations showed that an increase in depth followed by a reduction in layers' width led to increased accuracy scores. Each model was trained with a batch size of 64 sets, 100 epochs fitted with test accuracy callbacks, using an early stopping strategy with a patience of five epochs. Each model utilized the Adam optimizer [Kingma and Ba, 2017] with a learning rate of 0.001 and the binary cross-entropy loss function.

3 Fitting and Optimization

This section details the training processes involved in the development of the system. It includes descriptions on how each individual data element has been modified in order to best suit learning, transfer learning specifications and respective system integration. It also includes strategies researched alongside respective implications to system accuracy.

3.1 Input Manipulation

The test dataset is pre-processed before being fed into the model, and the following subsection details individually each element and the preprocessing strategy implemented.

3.1.1 Engagement

User Engagement is expressed as a numeric integer value. The engagement per post is expressed a numeric integer value. This is scaled and standardized, with values centered in 0 with $\sigma = 1$.

3.1.2 Date

In the training dataset, dates for each post has been presented in the yyyy-mm-dd format. This date was compared with the predetermined date 1st January 2015, in order to derive a numeric value representing the number of days from the date of reference. Min-max scaling is then applied to the numeric values further deriving float numeric values between in the range [0,1] which was subsequently fed into each model for training.

The test dataset expressed dates in the dd/mm/yyyy format, however this did not impede the data, as minor changes to the date interpreter allowed for the numeric conversion to dates. Data normalization remains unchanged throughout both processes.

3.1.3 Manipulation

The manipulation column of the dataset expresses an integer representing boolean true/false. If the image posted has been manipulated before being posted, the field displays 1, else 0. Data representation packages such as matplotlib showed the data to be noisy and a weak predictor; therefore it was dropped as input.

3.1.4 Visual Actors

Each entry was additionally provided with the visual actors present in the frame. If there was no individual of note in the frame, the value expressed for the entry was 0, representing boolean false. Each Visual element described in the training dataset has been utilized in order to build a binary field representing the element via a column. This was represented by boolean true/false; if the person of note mentioned in the column was present in the image, the entry expressed value 1. If no such individual was present, the binary field represented 0 across it's length.

The test dataset included samples where the image's visual elements was neither 0 nor fully represente by the training binary field, i.e, new visual actors were introduced by the test dataset. Pseudo columns if created to represent these individuals would consistently express 0 during training increasing noise and were hence dropped, displaying a binary field consisting of elements described during training only. While this was technically accurate, it did not allow ideal representation of the testing data, potentially impacting accuracy.

3.1.5 Text

Textual analysis of the data is one of the most crucial avenues of development and is critical to the success of the system. However, this problem cannot be approached similar to standard textual analytic frameworks. Memes produce a degree of complexity, primarily as they are elucidated in short concise phrases, and do not necessarily comply to standard grammatical rules. They also tend to use slang, which while conveying intended meaning to the reader, greatly increases the need for high model capacity. For this reason we selected UmBERTo, a RoBERTa-based neural language model pretrained on Italian texts extracted from the OSCAR corpus [Ortiz Suárez et al., 2020], for producing text representations.

SentenceTransformers Using the standard UmBERTo implementation we could only produce word-level contextual embeddings. To adapt the model for sentence-level classification without having to rely on the [CLS] token it produces, we use the SentenceTransformers framework [Reimers and Gurevych, 2019]. SentenceTransformers allows to produce full sentence embeddings by averaging all word embeddings produced for tokens of each sentence. We take fine-tuned representations for textual data and use them as components of our end-to-end system.

3.1.6 Image

While we have so far discussed simply the use of metadata in order to predict our results, it is important to address the core of a meme; the image itself. We can internally distinguish a meme from a standard image through the aforementioned broken sentence structure, meme 'templates', quick and messy edits, among others. Certain memes also look like standard images but it is the context that we see them in that distinguishes them between categories.

Due to this variance, it is impractical to expect a singular framework to effectively describe each distinguishable feature and utilize it to classify an entry. Hence, the learning is split between various pre-trained model architectures. Each of them uses a fundamentally different approach to feature extraction and as a result allows for a high variance with minimal sacrifice toward generalization performance.

ResNet This is the default vector representation provided by the task organisers. This is left unchanged and directly passed to model training. Each vector embedding has shape 2048.

AlexNet AlexNet is a powerful convolutional neural network built with 5 layers of CNN and 3 fully connected layers. AlexNet specialises in the identification of depth; objects such as keyboards and a large subset of animals are effectively classified by the network architecture. This makes it a good predictor as features such as depth are generally lacking in memes as they are either accompanied by text boxes. This is described by image vector embeddings from the AlexNet network allowing for a highly accurate classification tool. Each vector embedding has shape 4096.

Team Name	**Run**	**Precision**	**Recall**	**F1-Score**
UniTor	2	0.8522	0.848	0.8501
SNK	1	0.8515	0.8431	0.8473
UPB	2	0.8543	0.8333	0.8437
UniTor	1	0.839	0.8431	0.8411
SNK	2	0.8317	0.848	0.8398
UPB	1	0.861	0.7892	0.8235
DankMemesTeam	**1**	**0.8249**	**0.7157**	**0.7664**
Keila	1	0.8121	0.6569	0.7263
Keila	2	0.7389	0.652	0.6927
baseline	1	0.525	0.5147	0.5198

Table 1: Task 1 Results. The system score is reported in bold, with the team name DankMemesTeam.

DenseNet Pre-trained models such as ResNet and AlexNet use a large number of hidden layers. While the increase in depth allows for better feature extraction, the transmission pipeline is prone to data leakage, losing essential features before reaching fully connected layers. DenseNet employs novel techniques, fully connecting each hidden layer to one another. Consequently, resultant vector embeddings from pre-trained DenseNet neural networks consist of features sometimes missed by deep CNNs. Each vector embedding has shape 1000.

The objective of multiple vector embeddings was to cumulatively cover a significant portion of possible meme combinations and templates. As a result, the obtained system while being significantly computationally expensive, it accentuates test accuracy from 75% to 83% on validation data.

3.2 Data Augmentation

The testing dataset provided 1600 entries. Out of this, only 80% or 1280 images were available for training, based on train-test splits. This is a relatively small dataset, as popular classification models often train on thousands if not millions of training images before running accurate predictions. An efficient and effective method of increasing this training data involves using data augmentation strategies. In order to augment data, random changes are introduced in each image, modifying at random brightness, rotation and zoom. A permanent fill strategy nearest is applied throughout each augmentation. 9 augmented images are produced for every default image entry. As a result, the training dataset is increased from 1280 to 12800 images, leading to improved system effectiveness.

Every augmented image is associated to the same metadata as the original, varying only in the visual embedding itself. The ideal result would be an increase in the generalization performance as the model fits better to the general rule of recognizing memes. However, results show the opposite: unmodified metadata cause the system to overfit instead. This was in part due to augmentations not pertaining to the general meme template and a significant increase in the number of column entries with correlated training values.

A large set of augmentation strategies were applied to the dataset, modifying factors, ranges and augmentation count. No iteration significantly and consistently improved the performance of the system, and the augmentation process being computationally expensive, both for GPU usage and storage, was determined noisy, relatively inconclusive and therefore dropped from training.

4 Results

A k-fold cross validation strategy is utilized in order for the trained system to generate the challenge submissions. The dataset has been divided in 5 folds and 5 different instantiations of the same system compromised the final result. The system placed 7th in the singular task in which it participated, impeded primarily by inconsistent recall scores. Consequently, the F1-score is also significantly impacted. Table 1 describes the results and the standings of the system in comparisons to other participating systems.

A direction to improve the current system would be to modify the recall threshold, building a better precision-recall balance allowing for more accurate overall results. The final output can also be computed as a better iteration of majority-vote, focusing more on model confidences to influence the evaluation of the final output. The ensemble could also include more varied models with differing architecture to better accentuate differing feature extraction. The system could also be trained on additional data derived through web scraping; although necessitating complex automation systems, the massive influx of data will surely compensate, allowing for large strides in accuracy over an extended timeframe.

5 Discussion

In this paper we described ArchiMeDe, our multimodal system used for participating in the DankMemes task at Evalita 2020. The results produced by the system are promising, even if the systems does not encode inductive biases that are specific neither for multimodal artifact recognition, nor to meme detection in particular. The entry is not far behind at precision from the best performing systems, and there are several avenues for growth that display considerable potential. The paper highlights effectively the crucial impact of transfer learning to the success of this system. Implementations involving combinations of differing text-based analysis and image resolution also provide an interesting research direction.

Memes today are one of the most formidable modes of portraying one's idea while maintaining strong interpersonal connect. The informality of memes combined with their ease of making and distribution has only accentuated their growth. To be able to effectively interpret memes is a task far deeper than what can be intuitively thought. As humans continue to unravel their minds and derive ingenious methods of computationally recreating them, we realise the importance of slang, and how it relates directly toward core human principles. A piece of our culture, memes are the best represented and documented slang we have today, and to effectively interpret them means to cross a significant milestone with lasting impacts to the field of NLP.

References

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

V. Gurbaxani and H. Mendelson. Software and hardware in data processing budgets. *IEEE Transactions on Software Engineering*, SE-13 (9):1010–1017, 1987.

K. He, R. Girshick, and P. Dollár. Rethinking imagenet pre-training, 2018.

J. Howard and S. Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL `https://www.aclweb.org/anthology/P18-1031`.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2017.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

P. J. Ortiz Suárez, L. Romary, and B. Sagot. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online, July 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.acl-main.156`.

N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL `https://www.aclweb.org/anthology/D19-1410`.