Meme Humour Classification: Attention-based Multi-modal Fusion with Humor Theory

Bohan Shu, Xiao Wei, Yiwen Tu
University of Michigan, Ann Arbor
{*shubohan, weixiao, evantu*}*@umich.edu*

I. Introduction

Memes have become an integral part of online communication, allowing individuals to express specific ideas and emotions in a concise and direct manner. This makes memes one of the most prevalent means of modern communication. As a result, IT enterprises and natural language processing practitioners have focused on developing various models for meme detection tasks. In this project, we aim to tackle meme humour classification, a crucial aspect of meme analysis, using the Memotion dataset provided by SemEval 2021 Task 7[1].

Specifically, our goal is to take in memes as input, and identify the emotion expressed in a meme and assign it a rate in each category: sarcastic, humorous, motivational, or offensive (which are four subcategories of humour). However, humor in memes are challenging to detect, as they rely on the complex interplay between text and images. Bringing images into consideration adds a lot of noise to the data, and the interplay is hard to exploit. While existing methods have shown promise, they often do not fully exploit the semantic structure of humor in meme text.

To address this, we propose a Attention-based Multi-modal Fusion with Humor Theory model that fuses text and image embeddings while also leveraging the structure of humor to extract text embeddings. By combining these approaches, we aim to enhance its ability to exploit the humor semantics of meme text and improve its conprehension of memes. The code of our project is published in https://github.com/tu-yiwen/EECS_487_proj.

Our contribution mainly lies in:

- Utilization of humour theory to fine-tune RoBERTa. We find that after fine-tuning, our model gets improved at humour classification. This indicates that humour-theory fine-tuning may have prospective applications in other downstream humour classification and humour generation tasks.
- Utilization of state-of-the art pre-trained model RoBERTa and VGG19 as embedding generator, and use transformer encoder as fusion method.
- Data preposessing of images.

Task allocations:

- *Yiwen Tu* Paper Reading, Text Embedding Extraction, Fusion Model, Model Training, Report Writing
- *Xiao Wei* Image Embedding Extraction, Presentation Slides Making, Report Writing
- *Bohan Shu* Data Query, Data preposessing, Model training, Report Writing

II. Related Work

Multi-modal analysis is a cutting-edge technique that can be applied to various modalities, thus being a desirable solution for meme-related tasks. However, current work in this area have certain limitations. The best-performing model for the Memotion task[2] utilizes multi-modal analysis, combining text and image embeddings generated from a large language model. Another team[3] uses deep learning to tackle the humor classification task. Also, a team[4] used VGG-16 followed by multi-head attention and dense layer along with residual connections to extract image features. For extracting text features, they used BiLSTM followed by attention mechanism and fully-connected layers along with residual connection. The text and image features are concatenated in a late fusion. The primary similarity between these approaches is that they directly generate sentence embedding from pre-trained models or use late fusion, which fails to consider the inner semantics of the humour in text.

To design a better structure, we aim to leverage humor theory. Understanding humor has always been a critical area of research in natural language processing. A humor classification metric proposed by **Rosso et al.**[5] contains four criteria: "incongruity, ambiguity, interpersonal effect, and phonetic style," which perform well in humor classification tasks. Additionally, researchers are increasingly focusing on humor in modern communication. For example, **Reyes et al.**[6] categorized tweets as humorous or ironic by considering text ambiguity and incongruity. Another study by **Gorham et al.**[7] investigated the communication process of humorous memes on Facebook using humor and virality theory. Both studies apply humor theory to modern communication and achieved promising results. Considering the promising result and the situation for processing memes, we propose to use the two of the four classification criteria[5] to form our combined text embedding, which is **incongruity and ambiguity**.

III. Data Collection and Preprocessing

A. Data Collection

*1) **Dataset**:* The Memotion 2.0 dataset is chosen for our meme humour classification task. The original **Memotion** dataset[1] was released in 2020 as part of the SemEval-2020 shared task on "Memotion Analysis". It consists of 8k annotated memes, with extracted text using Google OCR system, and aims to solve humour classification, which suits our project very well.

Memotion 2.0 that we choose is the updated version of it and is released in SemEval 2021 Task 7. It contains approximately 8500 annotated memes in English, with intensity levels for Humour, Sarcasm, Offensiveness, and Motivation. The dataset is split into two sets, the training set (7000 memes) and the validation set (1500 memes). Each column (humor, sarcasm..) is labeled with an extent, such as (not motivational, motivational).

We focus on task B and task C. For task B, the task is a binary classification and is aimed to quantify to which extent a particular emotion is being expressed in a meme. The intensities are on a scale of 0 or 1 for humour (e.g. 0 - not funny, 1 - others) , sarcasm and offensiveness and only 0 and 1 for motivation (0 - not motivational, 1 - motivational). For task C, the task is an ordinal regression and is aimed to quantify to which extent a particular emotion is being expressed in a meme. The intensities are on a scale from 0 to 3 for humour (e.g. 0 - not funny, 1 - funny, 2 - very funny, 3 - hilarious) , sarcasm and offensiveness and only 0 and 1 for motivation (0 - not motivational, 1 - motivational).

(a) Meme Example 1 (b) Meme Example 2 (c) Meme Example 3

[1 0 0 0] [2 0 1 0] [3 0 0 0]

Fig. 1: Example from Dataset: Memotion 2.0.

[humour sarcastic offensive motivational]

Though Memotion 3.0 is also available, its containing Hindi English is out of the scope of our current research, thus being not as suitable as Memotion 2.0 for our work.

*2) **Data Distribution Analysis**:* The distribution of the dataset presented in the following figure, and it shows that the dataset is quite imbalanced. For example, the motivation classification has 6714 positive labels, while it only has 286 negative labels, which may cause a great bias on prediction. Another example is the humour classification. The distribution of humour class of each label are shown in Fig. 3. The rough proportion is about 6:3:1:30, which is quite imbalanced. Training on such dataset may lead to a great bias towards the majority class and the minority class.

	Train data				Validation data				Test data			
Label	Negative	Neutral	Positive		Negative	Neutral	Positive		Negative	Neutral	Positive	
Sentiment	973	4510	1517		200	975	325		451	971	78	
Label	0	1	2	3	0	1	2	3	0	1	2	3
Humour	918	3666	1865	551	229	745	419	107	62	892	398	148
Sarcasm	3871	1759	1069	301	804	388	246	62	185	248	892	175
Offensiveness	5182	1107	529	182	1110	238	107	45	943	457	87	13
Motivation	6714	286	-	-	1430	70	-	-	1480	20	-	-

Fig. 2: Distribution of Memotion Dataset.

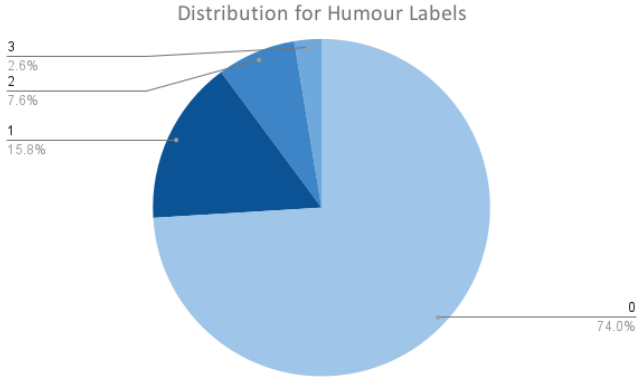

Fig. 3: Distribution of Humour Labels

Some authors, including Bucur et al. (2022) [2], suggest using **oversampling** to address the imbalance issue. However, as we have tested, oversampling won't lead to a better prediction, and thus we leave this problem aside for now.

B. Data Preprocessing

*1) **Remove Texts**:* After the extraction of textual information from memes, the presence of text may impede the extraction of meaningful image features. To address this issue, Optical Character Recognition (OCR) can be utilized to detect and locate text regions in the image. Then masks are defined around the text area to cover the texts, and the identified regions can be inpainted to effectively remove the text from the image. The result is shown in Fig. 4.

(a) Before Removing (b) After Removing

Fig. 4: Comparison of Same Images before and after Text Removing.

*2) **Color Enhancement**:* Akbarinia et al.[8] have shown that color enhancement can significantly improve image visibility. Therefore, we applied color enhancement in our preprocessing approach to highlight the crucial parts of the image that may contain more humor information. Specifically, we applied color enhancement to improve the contrast of the images, which helps to reveal more details and improve the overall

image quality. We then fed these enhanced images through our multimodal model and selected the best color enhancement technique that maximizes performance. By using this approach, we aimed to enhance the visual quality of the images and achieve better results in our downstream tasks.

(a) Original Image (b) Color Enhanced

Fig. 5: Example of Combined Processed Dataset.

IV. Methodology

A. *Pipeline*

The pipeline of our project is demonstrated in Fig.6.

In brief, we feed the pre-processed image to the finetuned VGG to extract the image embedding. After that, we feed text and fine-tune RoBERTa with calculated labels of incongruity and ambiguity and achieve a final embedding of 768D. Besides, we extract the embeddings combining text and image through pre-trained CLIP.

After reaching all parts, we stack the embeddings and feed it into the transformer encoder and add a final layer final classification. The detailed process will be discussed in following parts.

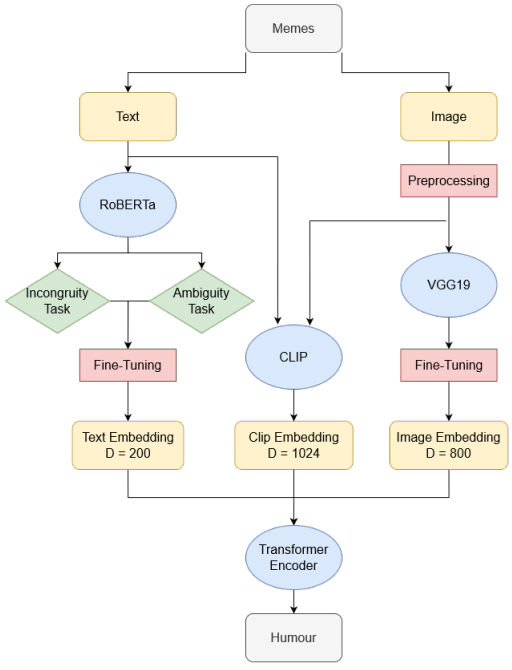

Fig. 6: Pipeline of Our Best Model.

B. *Meme Embedding Extraction*

At the very beginning, we start with representing a meme. To make a meme computational, we divide meme into text and image, and generate a unique embedding for each of them.

*1) **Text Embeddings**:* We have decided to use RoBERTa for text embedding. RoBERTa, a state-of-the-art NLP model introduced by Facebook AI Research in 2019, is an extension of BERT[9]. It excels at NLP tasks such as question-answering, sentiment analysis, and named entity recognition. It achieves improved performance through optimizations like a larger training corpus and dynamic masking. While it has achieved state-of-the-art performance on many NLP benchmarks, it may not accurately capture the incongruity and ambiguity unique to humor. Therefore, we choose to fine-tune RoBERTa for our humor classification task.

We model the fine-tuning task as a bi-task linear regression and use **Mean Square Loss** as the loss function to train the model. The two tasks are defined as:

- *Incongruity* is crucial for humor as it creates surprise by deviating from the audience's expectations. This generates cognitive dissonance, which can be resolved through laughter. Examples of incongruity include puns, wordplay, misdirection, and unexpected plot twists. Without incongruity, humour would be less enjoyable and predictable. We model the incongruity of a sentence as the largest meaning distance between words.
- *Ambiguity* is key to humor as it allows for multiple interpretations, leading to confusion and laughter. It appears in forms like puns, double entendres, and irony. These all rely on the audience's ability to interpret situations in various ways. Ambiguity creates a more engaging comedic experience as it requires the audience to actively engage with the text. To model ambiguity, we measure the smallest meaning distance between words, similar to how we model incongruity.

There are multiple methods for measuring the distance between words. We decided to use *The Wu&Palmer similarity:*: This method measures semantic similarity between two words based on their depth in the WordNet hierarchy and the depth of their lowest common ancestor. It is also commonly used in natural language processing tasks such as text classification, information retrieval, and word sense disambiguation. It is calculated as:

$$\text{Sim}_{wup} = \frac{2d_{LCS}}{d_1 + d_2}$$

where d_{LCS} is the Wordnet depth of the least common subsumer of w_1 and w_2, and d_1 and d_2 are the depths of w_1 and w_2, respectively.

*2) **Image Embeddings**:* We utilized the renowned VGG (Visual Geometry Group) model for generating image embeddings. Introduced by Simonyan and Zisserman in 2014[10], VGG is a convolutional neural network architecture widely used for image classification, object detection, and segmentation tasks. It is known for its simplicity and uniformity, consisting of convolutional, pooling, and fully connected layers.

To suit our practical needs, we fine-tuned the VGG19 model and framed the training process as a regression task. We used

the **MSELoss** loss function and the **Adam** optimizer. The detailed training process is outlined below:

i) We added four new layers to the VGG model, including fully connected layers of size (1000, 800) and (800, 5), and two ReLU layers after each. We froze the parameters from the pre-trained model.
ii) For each dimension - humour, sarcastic, offensive, motivational, and sentimental - we assigned an integer to each class and trained the model based on the loss of our predictions and the label vectors.
iii) After 40 epochs of training, we saved the model up to the third-last layer to obtain 800D embeddings from our input. For reference, the training loss was 0.0834, and the training accuracy was 91.38%.

*3) **CLIP Embedding**:* CLIP (Contrastive Language-Image Pre-Training)[?], is a state-of-the-art neural network architecture that can simultaneously process images and natural language text. Developed by OpenAI, CLIP is trained to recognize visual concepts and match them to corresponding textual descriptions, without the need for explicit object detection or segmentation. This approach has resulted in remarkable performance across various downstream tasks, including image classification, object detection, and even zero-shot image generation.

Based on pre-trained CLIP model, we propose a process to extract embedding of our memes:

i) We first tokenize our text with the attributed tokenizer of CLIP.
ii) After that, we apply CLIP text encoder to the tokens and CLIP image encoder to the image to receive 512D embeddings for each.
iii) Finally we concatenate the embeddings to get the final 1024D embeddings.

Since the CLIP is good at relating text to the corresponding image, the CLIP embedding can act as a bridge between the text and image of meme, thus revealing the deeper relationship between them.

*4) **Embedding Fusion, Model Training**:* We employ a fusion approach by stacking the three embeddings, which are of dimensions 800D (Image Embedding), 768D (Text Embedding) and 1024D (CLIP Embedding), respectively. Since they are of different dimensions, we first project them into 512D using a projection layer and stack them together. The stacked embeddings are then fed into the final fusion model. Note that since the labels are integers, we round the final results to get predictions, but not in calculating loss.

We tried these models for final fusion, and we use transformer encoder as the final fusion method:

- *Transformer Encoder(Multi-headed attention with feed-forward network)*. The encoder consists of identical layers, each with two sub-layers: a multi-headed self-attention mechanism that attends to the input sequence and a multi-headed attention mechanism attending to the output sequence. These sub-layers are followed by a position-wise feed-forward network. The activation function between the TransformerEncoderLayer is gelu.
- *Late fusion*. Each Embedding goes through 4 layers: from original dimension to 512D, from 512D to 128D, and from 128D to 32D, then from 32D to 4D (final prediction). Then we use mean pooling to get the final prediction. The activation function between each layer is ReLU.

*5) **Training Details**:* The code of our project is published in https://github.com/tu-yiwen/EECS_487_proj.

- *RoBERTa Finetuning* We use the value of incongruity and ambiguity discussed above as labels, append a classify layer at the end of RoBERTa and do a supervised learning. The learning rate is 0.00001, loss function is MSE loss, and optimizer is Adam. We train for 5 epochs. We then use the last hidden layer as the final sentence embedding.
- *Final model-Task C* The problem is modeled as an ordinal regression problem. We use the Adam optimizer and L1 loss function. We choose L1 loss function to induce sparsity in our data (because the labels ranges from 0 to 3, if we use MSE, the results will cluster together and hinder performance.) The tuning grid is as follows:
  - *learning rate* [0.1, 0.01, 0.001, 0.0001]
  - *weight decay* [0, 0.001, 0.00001]
  - *transformer layer number* [2, 4, 6, 8]
  - *attention head* [4, 8]
- *Final model-Task B* The problem is modeled as a binary classification problem. We use the Adam optimizer and cross entropy loss function. The tuning grid is as follows:
  - *learning rate* [0.1, 0.01, 0.001, 0.0001]
  - *weight decay* [0, 0.001, 0.00001]
  - *transformer layer number* [2, 4, 6, 8]
  - *attention head* [4, 8]

V. Model Evaluation

A. *Result Demonstration*

We submitted our predictions to codalab and compare our results to those groups participating in the contest before and the baselines given by the committee. The baselines and our model performance on codalab is listed below: (The scores are all **weighted F1 scores**

*1) **Task B**:* We calculate the F-1 score of our model on Task B. Note that we include the result of our model, both trained on multitask and single task for each humour subcategories. The scores of the leading groups and baselines are in light color, as shown in Fig.7.

TABLE I: Task B

	Single Task	Multi Task	Little Flower	Baseline
Humor	0.9384	0.9381	0.9384	0.7944
Sarcastic	0.8191	0.7011	0.819	0.6575
Offensive	0.4853	0.5181	0.554	0.5346
Motivational	0.98	0.98	0.98	0.9575

It can be observed that our model reaches the same as the highest score on Task B comparing with the current leading groups on humour, sarcasm and motivational. But our model did not perform well in the prediction of offensiveness. We

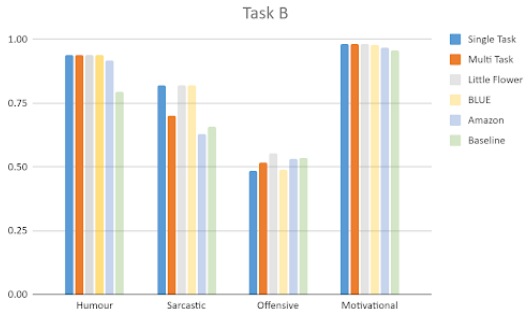

Fig. 7: Comparison of Performance of Leading Groups and Baselines on Task B.

lags the baseline by 0.05. This may be because the imbalanced dataset (Though oversampling did not do well as we tried). The detailed data is listed in TABLE 1.

*2) **Task C**:* The performance on Task C is listed below. The F1 scores are demonstrated in Fig.8, with detailed data in Table 2. Similarly, the scores of the leading groups and baselines are in light color.

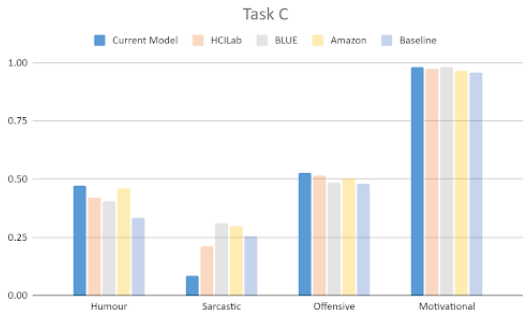

Fig. 8: Comparison of Performance of Leading Groups and Baselines on Task C.

TABLE II: Task C

	F1 Score				
	Current	HCILab	BLUE	Amazon	Baseline
Humour	0.473	0.4212	0.4036	0.4598	0.3349
Sarcastic	0.0866	0.2109	0.3083	0.2979	0.2533
Offensive	0.5268	0.5144	0.485	0.5021	0.4799
Motivational	0.98	0.974	0.98	0.9638	0.9575

It can be observed that our model reaches the highest score on Task C comparing with the current leading groups in humour, offensive and motivational. We outperforms the bert group on humour by 0.013, and on offensiveness by 0.013 But our model did poorly on sarcasm, which even lags behind baseline by 0.16. This may be because of the difficulty of multi-tasking. (We also tried single tasking for task C and it did not perform well.). Task C is more difficult than task B because the labels range from 0 to 3.

*3) **Ablation Study**:* Finally, we conduct ablation study on our current model. To be specific, we experiment on several methods: late fusion, single modality and base model.

Single modality means that we only consider one part of embedding, such as image only or text only. Base model shows our work of directly using embedding from pre-trained model without finetuning to our purpose. The comparison and scores are shown in Fig.9. The detailed data is listed in TABLE 3.

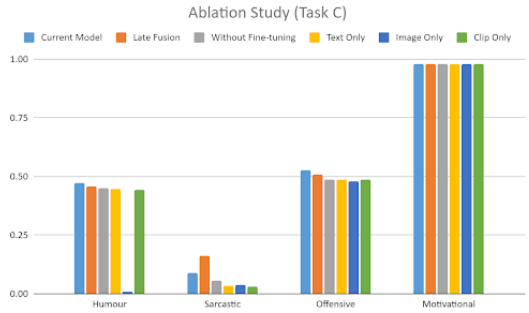

Fig. 9: Demonstration of Ablation Study Result.

TABLE III: Ablation Study

	F1 Score					
	Current	LF	No FT	Text	Image	Clip
Humour	0.47	0.46	0.45	0.44	0.01	0.44
Sarcastic	0.09	0.16	0.06	0.03	0.0376	0.03
Offensive	0.53	0.51	0.49	0.49	0.48	0.49
Motivation	0.98	0.98	0.98	0.98	0.98	0.98

It can be observed that our current model performs better than only-text, only-image and only-clip baselines. Also, fine-tuned model performs better than non-fine-tuned methods, and transformer method perform better than late fusion (structure discussed above).

*B. **Discussions***

As can be seen, we reaches approximately the best scores for task B and task C, except offensiveness in task B and sarcasm in task C. Overall, we cannot say our model performs really well. We can see that the difference on offensiveness in task B and sarcasm in task C with the baseline and the best models are not small. We reflect on the whole process and summarizes several possible reasons:

*1) **'Trick' of High Score**:* Though our model reaches the performance level of other leading groups, we still possess a doubt "whether our model did capture the structures of the dataset", which is deepened through our inspection of the prediction details. As detailed in the dataset part, Memorion is severely imbalanced in training set and even test set. For example, consider the distribution of "motivational" in dataset: there are 6714 '0's and 286 '1's in training set and 1480 '0's and 20 '1's in test set. This means that if the model predict all '0's for all images, it still can reach a really good score! In fact, all the high-performing model in the contest, including ours, tend to predict so and get a weighted F1 score of 0.98. However, this does not mean that the models actually "learn" something from the data. It's better to test on a more balanced data. It's possible that a better model which actually "learns something" have a lower F1-score compared to this cheating model.

*2) **Are we really learning the relationship between text and image?**:* During the process of model training, we often feel doubtful: has our model really learned the relationship between images and text? On one hand, we are limited by the dataset, which makes it difficult to evaluate the performance of our model. On the other hand, there is a question mark over whether CLIP embeddings have really learned the relationship between images and text. This may be the reason why our predictions on certain classes are not as good as expected. We believe that early fusion may be better at learning the relationship between the two.

*3) **Model Training**:* During the model training process, we found that most of our models did not converge well. Either they converged and oscillated back and forth at a high loss, or they had a very high validation loss. Even after hyperparameter tuning, we were unable to achieve significant improvement. This may be related to the models we have chosen. After all, the transformer encoder is more like a black box, and we may need a more extensive hyperparameter search, but this would be too time-consuming for us. On the other hand, we find that even though we use L1, loss, the results for task C still seem to cluster together. This may arise from the noise of data.

VI. Future Improvements

Due to time limit, we don't have time to test all our proposed methods. The following is a list for further improvements of our model that will be implemented after the course.

A. New Pipeline

In our current approach, we extract the embeddings independently from text and image, relying mainly on the pretrained models' ability. Although we have dived into theory of humour as reference for our finetuning, it still lacks certain interpretability and does not explore the underlying relationship between text and image. We want to explore how each parts of the image relate to that of the text.

Therefore, we want to design a new pipeline similar to that of the CLIP model[?], as shown in Fig.3. In this pipeline, each pair of text and image is considered a positive sample, and all other combinations are negative samples. This setting enables us to fine-tune the process of embedding extraction, yet the detailed design of the new model is on the progress.

Overall, our method can be classified as "early fusion" because we fuse the image and text early in the pipeline to obtain our final embeddings. Moreover, we expect that this method will provide better interpretability by exploring the relationship between text and image and treating them as a cohesive unit rather than separate components.

B. Early Fusion

Besides late fusion and intermediate fusion method, we can also feed text and patches image to models like VisualBERT at the very beginning. We don't have time to test how this model work, but we expect it to perform better than our current model because it can learn more of the interplay between two modalities.

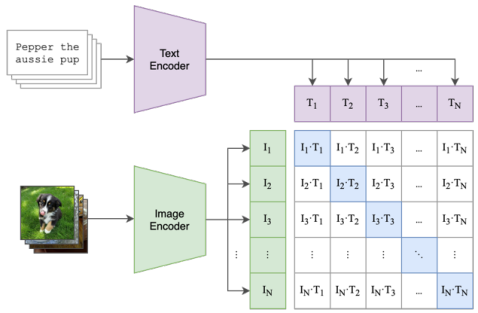

Fig. 10: Pipeline of CLIP, Inspiring Our New Model.

References

[1] C. Sharma, D. Bhageria, W. Scott, S. PYKL, A. Das, T. Chakraborty, V. Pulabaigari, and B. Gambäck, "Semeval-2020 task 8: Memotion analysis - the visuo-lingual metaphor!" *CoRR*, vol. abs/2008.03781, 2020. [Online]. Available: https://arxiv.org/abs/2008.03781

[2] A.-M. Bucur, A. Cosma, and I.-B. Iordache, "Blue at memotion 2.0 2022: You have my image, my text and my transformer," 2022. [Online]. Available: https://arxiv.org/abs/2202.07543

[3] D. S. Chauhan, D. S R, A. Ekbal, and P. Bhattacharyya, "All-in-one: A deep attentive multi-task learning framework for humour, sarcasm, offensive, motivation, and sentiment on memes," in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 281–290. [Online]. Available: https://aclanthology.org/2020.aacl-main.31

[4] K.N.Phan, G.-S. Lee, H.-J. Yang, and S.-H. Kim, "Little flower at memotion 2.0 2022 : Ensemble of multi-modal model using attention mechanism in memotion analysis," 2022.

[5] D. Yang, A. Lavie, C. Dyer, and E. Hovy, "Humor recognition and humor anchor extraction," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 2367–2376. [Online]. Available: https://aclanthology.org/D15-1284

[6] "From humor recognition to irony detection: The figurative language of social media," *Data & Knowledge Engineering*, vol. 74, pp. 1–12, 2012, applications of Natural Language to Information Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169023X12000237

[7] "Humour 2.0: Styles and types of humour and virality of memes on facebook," *Journal of Creative Communications*, vol. 10, no. 3, pp. 288–302, 2015. [Online]. Available: https://doi.org/10.1177/0973258615614420

[8] A. Akbarinia and K. R. Gegenfurtner, "How is contrast encoded in deep neural networks?" *arXiv preprint arXiv:1809.01438*, 2018.

[9] 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.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.