Sarcasm Detector using DynRT

Shreyas Patil
George Mason University
spatil28@gmu.edu

Anirudh Batibrolu
George Mason University
abatibro@gmu.edu

1 Introduction

1.1 Task / Research Question Description

The paper which we have reproduced is: Yuan Tian (2023). This research paper focuses on multi-modal sarcasm detection using a Dynamic routing transformer network. Most of the existing research that has been done in this domain majorly relies on the incongruity between image and text as an indicator that there is sarcasm being used. In this paper, they have taken inspiration from a routing-based dynamic network and proposed a Dynamic Routing Transfer Network (DynRT-net). At this point, sarcasm is such a widespread and dominant tool in communication the study of sarcasm detection has been prevalent for quite some time. This involves a model that not only linguistic cues but also understands visual cues.

1.2 Motivation and Limitations of existing work

The Study of sarcasm detection has been a hot topic in NLP for quite some time as it has a wide range of applications in understanding natural dialogue. There are other papers within the same domain of research and they have tried to solve the same issue but they had problems with efficiency.

Some of the earlier Research done in this domain was by (Tepperman et al., 2006) to understand natural dialogue, (Riloff et al., 2013) public opinion mining, (Tsur et al., 2010) social media analysis. All of this initial research led to rapid growth in the multimodal models for the detection of sarcasm in the recent few years. The authors also then list down the abundance of research that took place in recent years (Cai et al., 2019; Xu et al., 2020; Pan et al., 2020; Wang et al., 2020; Liang et al., 2021; Pramanick et al., 2022; Liang et al.,2022; Liu et al., 2022). The authors noticed that the most common way to identify sarcasm is if, in these multimodal data sets, there was an incongruity between the visual cues and the linguistic cues.

However, most of them ended up using models that were not dynamic and the architecture of these models ended up being very rigid with the method of detection irrelevant to the input. So to tackle this problem and to induce a dynamic component to identify cross-modal incongruity the authors came up with one possible solution which is to model a dynamic mechanism with the routing-based dynamic network. In this approach, a series of modules are used, and depending on the image and sentence pair we can dynamically use the suitable modules. The existing routing-based method in multimodal dynamic networks only works on single-modality data. This will not be sufficient to build and predict vast data of image-text pairs. Therefore they extended the routing scheme to work on multimodal data aiming to build a better model for sarcasm detection.

The novelty that is being introduced here is the Dynamic Routing Transformer Network. The model's router helps with the dynamic routing of the transformer modules with hierarchical co-attention to adapt to the cross-modal incongruity between different kinds of image-text pairs.

1.3 Proposed Approach

This section describes the proposed approach. To create a multimodal system that can predict sarcasm by identifying both text and images the model proposed is DynRT. The following is a brief description of the working of the model:

Encoding: In the encoding part we have used a pre-trained Roberta and Vit(Vision transformer) for initializing the embedding and this was done because of the robustness and the fact that RoBERTa was trained on large-scale data sets. Both these models are pretty robust and hence pro-

Modality	Method	*F1*	*Acc*
Image	ResNet (Cai et al., 2019)	61.53*	64.76*
	ViT (Dosovitskiy et al., 2021)	66.90 ± 0.09	68.79 ± 0.17
Text	TextCNN (Kim, 2014)	78.15*	80.03*
	SIARN (Tay et al., 2018)	79.57*	80.57*
	SMSD (Xiong et al., 2019)	79.51*	80.90*
	Bi-LSTM (Liang et al., 2022)	80.55*	81.09*
	BERT (Devlin et al., 2019)	81.09*	83.85*
	RoBERTa (Liu et al., 2019)	83.42 ± 0.22	83.94 ± 0.14
Image + Text	HFM (Cai et al., 2019)	80.18*	83.44*
	D&R Net (Xu et al., 2020)	80.60*	84.02*
	IIMI-MMSD (Pan et al., 2020)	82.92*	86.05*
	Bridge (Wang et al., 2020)	86.05	88.51
	InCrossMGs (Liang et al., 2021)	85.60*	86.10*
	MuLOT (Pramanick et al., 2022)	86.33	87.41
	CMGCN (Liang et al., 2022)	87.00*	87.55*
	Hmodel† (Liu et al., 2022)	88.92 ± 0.51	89.34 ± 0.52
	HKEmodel† (Liu et al., 2022)	89.24 ± 0.24	89.67 ± 0.23
	DynRT-Net†	**93.21 ± 0.06▲**	**93.49 ± 0.05▲**

Figure 1: Comparison of accuracy with other methods with DynRT at the end

vide a strong foundation.

Dynamic Routing: The idea of dynamic routing is novel and tackles the problem of adapting to different image-text pairs. Static models can miss subtle nuances that are required in detecting sarcasm. DynRT shows potential in resovling that.

Hierarchical Co-attention: This mechanism, which allows for progressively more diverse attention masks with each layer of the transformer, seems promising for capturing various levels of interaction between the text and image features. It reflects a nuanced understanding that not all parts of an image are equally relevant to the accompanying text, and vice versa.

Routing Probability: The use of the Gumbel Softmax for routing probability is interesting, as it can help in sampling from a discrete distribution and thus deciding which co-attention masks to emphasize during the model's learning process. This allows the model to "focus" on more relevant parts of the multimodal input.

Loss Function: Utilizing cross-entropy loss is a standard choice for classification tasks, indicating that the model's outputs are being tuned to closely match the ground truth labels in a probabilistic sense.

1.4 Likely challenges and mitigations

Our project had a lot of challenges during the setup stage rather than problems with execution. The authors had stated their versions of modules used in the project in requirements.txt but we didn't have their version of Python so we had to execute using our versions and we downloaded the necessary versions of all modules without causing conflicts which took a lot of time and effort. After installing all the required software and datasets, the entire project folder was 21.7GB and we couldn't use Google Colab as it has a drive limit of 15GB so we decided to run it on Aniruddh's device which has a Nvidia GPU. However, some of the methods that were used in the model and TRAR were depreciated so we decided to change some of the methods and update them to the latest format. The optimal hyperparameters were already given so we did not encounter any significant challenges in the execution part of our project.

If our experiments did not go as planned then we had planned to use Hopper, our college provided us with a GPU server for NLP tasks if we could not run it on the device. We also decided on an option in which we uninstall all our pre-existing requirements to install the versions of the modules and software the author has specified. In case the project not running at all or if we encountered insurmountable challenges in executing the tasks, we had also selected other papers to reproduce in that case.

2 Approach

2.1 Models Used for Evaluation of Sarcasm Detection

The two models that we have used in our project for the detection of sarcasm are the xlm-RoBERTa-large model as well as the xlm-RoBERTa-base for multilingual RoBERTa. The project was previously working on RoBERTa-base and now to introduce multilingualism we went with the xlm-RoBERTa-base. We chose XLM-RoBERTa-base over xlm-RoBERTa-large as the code was not optimized for the larger model. The number of layers in the larger model is 24, while it is 12 for the base models. The code would have to be majorly altered to produce results.

XLM-RoBERTa is trained in 100 languages, and excels in multilingual contexts, making it ideal for global or diverse language datasets. xl-RoBERTa-large, though larger and potentially more powerful for English-centric tasks, demands greater computational resources and wasn't producing results for our project. XLM-RoBERTa-base is more efficient for training and deployment with limited resources. It's particularly effective in cross-lingual tasks and transfer learning across languages. If our focus is on multilingual applications or computational efficiency, xlm-RoBERTa Base is the better choice. However,

for English-specific tasks with ample resources, xlm-RoBERTa-large may be preferable.

2.2 Motivation/intention

The study "Dynamic Routing Transformer Network for Multimodal Sarcasm Detection" offers a novel method for sarcasm detection in multimodal content—that is, content that incorporates text and visuals. This method's main component is the Dynamic Routing Transformer Network (DynRT-Net), which is made to respond dynamically to text-image inconsistencies, which are a crucial aspect of sarcasm. A hierarchical co-attention mechanism accomplishes this adaptation by aligning and focusing on particular text segments and matching visual regions that together express sarcasm. The method allows the network to dynamically choose the most relevant features from both text and images by extending routing notions in dynamic networks to the multimodal domain. This is important because sarcasm can appear in a variety of ways in diverse multimodal contents, and static networks might not be able to. This method's motives stem from the particular difficulties in detecting sarcasm in multimodal content. Sarcasm frequently results from a misalignment of textual and visual features, hence a system that can dynamically analyze and align these modalities is required. With the variety of snark, the technique must be flexible enough to handle a wide range of data formats. Moreover, the growing popularity of multimodal information on social media and other platforms emphasizes the necessity for efficient sarcasm detection systems.

2.3 Introduction of Multilinguality

To introduce multilinguality we chose to go ahead and introduce 3 different languages. Our model had a great performance in English but to see how well this model performs in different languages we added French, Spanish as well as Hindi. As sarcasm as a linguistic tool is used differently in different languages and is also different in different cultures, we tried to add the languages that are spoken by large masses.

Including Hindi in our multimodal sarcasm detection dataset enhances cultural nuance, expands linguistic diversity, and improves model robustness. It also broadens market applicability, especially in regions where Hindi is prevalent, and contributes valuable cross-linguistic insights to sarcasm research.

The reason we went ahead with French as one of our languages is because of its distinct phonetics—which were shaped by Frankish and Gaulish invasions—French has a distinct sound. It differs from its Romance equivalents in rhythm due to its usage of liaison and elision, nasal vowels, and silent letters.

Finally, we chose Spanish because incorporating Spanish in our multimodal sarcasm detection dataset enriches it with diverse linguistic and cultural contexts, enhances the model's applicability in Spanish-speaking regions, and provides valuable insights into the nuances of sarcasm across different languages and cultures.

2.4 Robustness

To identify whether our model is robust in terms of error handling is essential as it is pretty much possible that in real-world implementation of the model it might face human errors such as misspelled words or typos and also to asses how the model performs when there is a change in the wording or phrasing. Three main techniques were used to check the robustness of the model:

1) Introduction of misspelled words: In this, we added some misspelled words on purpose. We chose sentences that had a length of more than 4 words because if it's less than that it is kind of harder to understand the essence of the sentence. In every sentence that was chosen, we changed two words.

2) Change of voice: This was also a method to check whether the model has a deep understanding of sarcasm or is it just matching tensors. So in the given dataset we check whether the voice of the sentence is active or passive and change it to the corresponding opposite voice.

3)Adding synonyms: Adding synonyms was another crucial step to check whether the model works well on sentences that essentially mean the same but are different. The addition of synonyms was a tricky task though, as the simple replacement might change the meaning if it is not used in the right context. The changed words were adjectives and nouns. We didn't change any verbs or adverbs as that would drastically affect the meaning of the sentence and due to that, the labels of the sentence can change.

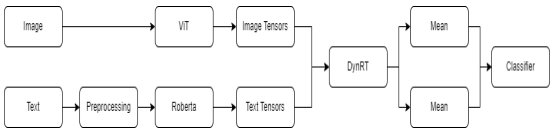

Figure 2: Flowchart of our approach

2.5 Our Approach

The approach that we took is to first introduce a multilingual version of our previously existing model. We went ahead with the XLM-RoBERTa-base. The reason we went ahead with that model is that XLM-RoBERTa Base is particularly useful in sarcasm detection for its ability to understand and process multiple languages effectively. This model, trained on a vast multilingual corpus, excels in capturing nuances and contextual meanings across different languages. In sarcasm detection, which often relies on subtle linguistic cues and cultural context, XLM-RoBERTa's multilingual capability allows it to more accurately identify sarcasm in diverse language datasets. This is especially valuable in scenarios where sarcasm is expressed in languages other than English or multilingual content, making XLM-RoBERTa Base a robust choice for global and culturally varied sarcasm detection tasks.

When it came to training the model in different languages we took a very minimalistic approach and used the previous data and labels to convert them to the respective languages using the translation modules in Python and gave the respective labels to the translated data. We then trained and tested the model on that data.

When it comes to testing we introduced a test file with the corresponding errors in the file to check the robustness of the model that we trained and as mentioned in the robustness section we made the necessary changes with the change of synonyms, the voice change, and the introduction of typos and then we saw how our model performed.

3 Experiments

3.1 Datasets

We utilized a multimodal sarcasm detection dataset from a renowned https://github.com/headacheboy/data-of-multimodal-sarcasm-detection, comprising 24,635 items of images and English text, sourced from Twitter. This dataset, widely used in the research community, serves as a benchmark for comparing multimodal sarcasm detection models. While the images reflect a global diversity, the text is uniformly in English, aligning with our research focus on the interplay between visual elements and English text.

The dataset's public availability facilitates widespread use and standardized model performance comparisons. We divided it into training, development, and testing segments in an 80/10/10 split, following meticulous preprocessing. We have 24635 images and text in total, with 19708 examples in the training set, 2464 examples in the testing set, and 2463 examples in the validation set. Our dataset consists of text, id, and label, where text gives us the sarcastic sentence, id gives us the filename of the image file and label is the classification(whether the sentence is sarcastic or not). After preprocessing and cleaning, there are 19557 sentences in the training set, 2373 sentences in the testing set, and 2283 sentences in the validation set. In preprocessing, we have removed all tags or words like "sarcasm", "sarcastic", "humor" or "joke" in the dataset as it tells us whether the sentence is sarcastic or not. This structure ensures comprehensive training and effective model evaluation, crucial for understanding sarcasm's subtleties in a multimodal context. Our study leverages this dataset to enhance the understanding and development of advanced sarcasm detection models, combining natural language processing and computer vision.

3.2 Metrics

The metrics that we use for measuring our model along with the baseline methods are accuracy and F1 score:

Accuracy: It is the difference between the true label(true prediction) and the prediction made by our model. It's a very straightforward term that is easy to calculate.

F1 score: It is the harmonic mean of the precision and the recall. We use the F1 score here as the labels might be imbalanced. There may be more examples of being classified as non-sarcastic rather than sarcastic. TP here means True Positive, FP - False Positive, and FN - False Negative

For our project implementation, we calculate other metrics such as precision and recall as well but for comparisons and final result, we state only the accuracy and F1 score as it represents preci-

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})}$$

Figure 3: F1 score formula

sion and accuracy as well.

3.3 Implementation

This is the GitHub repository of our implementation. https://github.com/RaydenDarkus/Sarcasm_Detector_Using_DynRT

We have also used the following resources:

1. The Dataset, image tensors, RoBERTa model, and xlm-RoBERTa models are uploaded in this folder in Google Drive: https://drive.google.com/drive/u/2/folders/1o-PLlhYN9AqFpJ8MHnuVfablV-_TFVF9

2. Originally, these were from https://github.com/headacheboy/data-of-multimodal-sarcasm-detect and https://huggingface.co/roberta-base/tree/main

At first, we downloaded the datasets for the images and the text. Then we preprocessed the texts and performed cleaning on them. The RoBERTa model was downloaded and put in a folder roberta-base. After that, we converted the images into image tensors using ViT. The image and text encodings were passed through several layers of the Dynamic routing transformer network. It performs routing on hierarchical co-attention of two modalities to capture cross-modal incongruity adapting to different image-text inputs. The model computes the hierarchical co-attention and the routing probability of the k^{th} layer(transformer layers) using the softmax function. This is then used for classification to tell whether the input is sarcastic or not.

The hyperparameters we used for Adam were a learning rate of 1e-6 and a weight decay of 0.01. The learning rate for RoBERTa was 3e-7 with a weight decay of 0.01. The hyperparameters were common for both testing and training. The results were then stored in the checkpoints folder which includes the classification results, the saved model, and the log file which gives complete details of each run.

3.4 Multilinguality implementation

In this, we took the training texts, testing texts, and validation texts in the prepared clean folder in input and converted them to a text format as it was in a binary format. After that, we used a module called Helsinki from the huggingface library to convert it into French, Spanish, and Hindi. Then we converted them to their corresponding binary files to reduce the computation time and space complexity of the algorithm. We separately trained the model specifically on language and then tested the model to see its results on unseen data of the same language. We had constraints in the number of sentences as there were a total of 24635 such sentences. If we took a concatenated multilingual dataset, it would be more than 100000 sentences with linkage conflicts as some sentences would be linked to the same image and the code would not work for the data so we executed the training and testing of the dataset of each language separately.

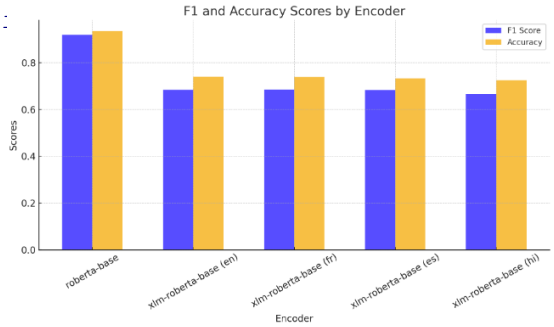

Figure 4: F1 score and Accuracy of baseline and other languages

The findings that we unveiled with the usage of the multilingual xlm-RoBERTa-base were pretty interesting, with the usage of Roberta-base we were able to achieve an accuracy of 93 percent. But with the use of xlm-RoBERTa-base dropped to almost 74 percent.

3.5 Robustness implementation

The robustness of an NLP (Natural Language Processing) model refers to its ability to maintain performance and make accurate predictions when faced with various challenging scenarios or unexpected inputs. Robustness is a critical quality for NLP models, as it determines how well they gen-

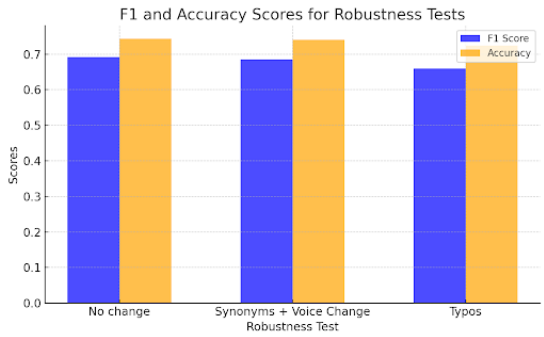

Figure 5: F1 score and Accuracy of Robustness tests

eralize to real-world data and handle different conditions that may differ from the training data.

The Robustness of the model was found to be pretty solid even after substituting nouns, and adjectives with their synonyms and changing the voice of the sentence. Next, we also introduced typos/mis-spellings in the sentences. The model still seemed to perform at more or less the same accuracy with a change of 1 to 3 percent while using xlm-RoBERTa-base as our encoder.

3.6 Results

The tables for our results compared with the results of the author is shown below:

Set	F1	Accuracy	Loss
train	0.9827	0.9831	0.3386
valid	0.9397	0.9518	0.3615
test	0.9193	0.9359	0.3750

Table 1: Our Run of the experiment

Set	F1	Accuracy
test	0.9321	0.9349

Table 2: Results of the author

We can see that the authors' claim is valid as we are getting a test parameter F1: 0.9193 while the authors got it as 0.9321 whereas our accuracy was 0.9359 while the author's was 0.9349. There is not much difference in the results.

For the given models, when we see the comparison between the F1 and the accuracy we see that there is not much difference in the scores.

When it comes to multilingualism, we can see that the metrics drop by almost 20 percent, and for robustness, the metrics drop by 1-3 percent.

Table 1 shows us our run of the experiment for Checkpoint 1 while Table 2 is the author's run of the experiment. In Table 3, we have our results for multilingualism in different languages. en indicates English, fr - French, es - Spanish, and hi - Hindi. In Table 4, we have the result of robustness testing. The encoder we have used there is xlm-RoBERTa-base. The tests show almost no change from the baseline metrics. This proves that our model is robust.

Encoder	language	F1	Accuracy
RoBERTa	en	0.9193	0.9359
xlm-RoBERTa	en	0.6920	0.7434
xlm-RoBERTa	fr	0.6859	0.7391
xlm-RoBERTa	es	0.6833	0.7328
xlm-RoBERTa	hi	0.6660	0.7252

Table 3: Results on Different Languages for Testing Set

Test	F1	Accuracy
No change	0.6924	0.7434
Synonym+voice change	0.6845	0.7408
Typos	0.6594	0.7240

Table 4: Robustness results on Testing Set

The log file of the results are found in exp/checkpoints folder in our repository.

3.7 Discussion

We faced challenges in setup and experienced hardware limitations as the dataset was huge. We could have implemented the task even earlier if not for the limitations in Google Drive space. In the future, we will make sure to think about solving such issues early on and explore other alternatives like Hopper for our project.

Our results do not differ much from the authors who have published the paper. We did not make many changes to the dataset or the hyperparameters as of Checkpoint 1 as some models are very sensitive to even minute changes which can decrease the accuracy or the efficiency of the model majorly.

For checkpoint 2, we chose 3 other languages to translate for the tests regarding multilingualism and 2 cases for robustness. The details of our implementation are provided in the results above.

3.8 Resources

Our team has two members, Shreyas and Anirudh. We were able to successfully download all the requirements of the project and run them in op-

timal time. We cloned the folder directly from GitHub and then downloaded the images from the dataset which had 24635 images in total. The entire download and extracting using 7zip took around 20 minutes. Then we downloaded the RoBERTa model from the HuggingFace library. Using ViT, we converted the images into tensors which took around 1 hr for all the images. Anirudh downloaded the dataset while Shreyas adjusted the libraries and versions of other modules to suit Python 3.11.4 and downloaded Cuda-toolkit 12.1 which was needed for the project. Shreyas performed training on the dataset with the hyperparameters that we were given. Anirudh then ran testing and performed error analysis, while Shreyas uploaded the folder to the repository with specified instructions. Shreyas then performed a literature survey to identify papers that had research topics in the same domain(multimodal sarcasm detection). Next, Shreyas did the translation of our dataset into Spanish, Hindi, and French. After translation, Anirudh modified the testing set and performed the experiments for robustness as well as performed training and testing on the translated sets. Our training took around 2 hours for 15 epochs on Nvidia GeForce 3050 RTX, while testing was completed in under 1 minute. Overall, the project took around 17 days to reproduce and implement, with regular calling and meetups, both online and offline.

As we are achieving less accuracy on the multilingual encoder xlm-RoBERTa-base, this is because our model is very sensitive to even minute changes. The hyperparameters that we have used before will not work for a different text encoder. As such, we need to perform a grid search or a random search using multithreading to find the optimal hyperparameters for the best results. We can also use optuna library in Python to perform this task. This works by taking a range of learning rates and weight decays for the different encoding models and saving the best results or the best model.

3.9 Error Analysis

We found that while the differences in metrics for the test set are not much, there is a minute difference in the F1 metric, we have a score of 0.9193 while the authors are getting an F1 metric score of 0.9321. While the model does not fail in its entirety, it incorrectly classifies some examples as not sarcastic due to differences in linguistics as sarcasm or even humor differs from region to region. Some images had captions in them in Hindi or other languages while the accompanying text with them was in English so the model also needs more examples of images with captions in other languages or more examples of sarcasm from different parts of the world.

4 Related Work

Here's a brief description of the relevant papers as requested, including how the proposed work differs from these:

1.Rossano Schifanella (2016) Schifanella et al., 2016: This study combines textual and visual embeddings through simple concatenation for multimodal sarcasm detection. The proposed approach goes beyond simple concatenation by using dynamic routing to account for the nuanced interactions between modalities.

2. Nan Xu (2020) Xu et al., 2020: This research captures cross-modality contrast and semantic association for sarcasm detection by looking at both simultaneously. In contrast, the proposed approach employs a dynamic routing transformer to adaptively capture the intricacies of cross-modal incongruity.

3. Bin Liang (2022) Bin Liang (2021) Liang et al., 2021, 2022: These papers utilize graph convolution networks to map both in-modal and cross-modal relationships for sarcasm detection. The current work differs by incorporating a multimodal dynamic network that dynamically adjusts to the diversity in multimodal samples, rather than a fixed graph-based structure.

4. Shraman Pramanick and Johns (2022) Pramanick et al., 2022: The authors use self-attention to model intra-modal and optimal transport for cross-modal relations. The proposed DynRT-Net instead dynamically routes attention to more effectively capture varying degrees of multimodal incongruity.

Each of these papers contributes to the field of sarcasm detection using both images and text, but the proposed approach by the user aims to address the limitations of static computation mechanisms by introducing a dynamic, adaptable architecture tailored to the complexity of multimodal sarcasm detection.

5 Conclusion and Future Work

The paper, "Dynamic Routing Transformer Network for Multimodal Sarcasm Detection" is reproducible. The results we reproduced are not very different from those of the authors so the reproducibility part was successful. RoBERTa model was used for text encoding while ViT model was used for image encoding. The research done in this paper combined previous work in Multimodal Sarcasm Detection with dynamic routing networks. For detecting sarcasm, the sentiment conveyed in the text is the complete opposite of what is conveyed in the image. Our understanding is that many of the previous methods only adapt to one type of data at a time, either images or text, but not both together. The authors extended this concept to cover both images and text simultaneously within a dynamic network, which will help grasp the complexities of sarcasm better, even when it involves unconventional combinations of images and text. Overall, the result of this paper has adequate accuracy as we can see in the results. Additionally, this is a state-of-the-art model which indicates that this is the latest paper in this domain of multimodal sarcasm detection that has the best results in metrics of F1 parameter and accuracy when compared with previous methods.

For multilingualism, our results while using xlm-RoBERTa-base are adequate but it's not an optimal result as there is a difference of around 20 percent from using RoBERTa-base on English sentences. This is because the model is very sensitive and a different encoder gives a different result so we need to tweak the hyperparameters for xlm-RoBERTa-base For robustness, we can observe that our model is robust, as for both test cases the difference in the metrics is only 1-3 percent. Our model does not exhibit a different behavior or performance in edited test sets.

The authors state in their limitations that, the current co-attention design in their method is constrained to only four types, which can limit its versatility. Also, due to the lack of publicly available datasets for multimodal sarcasm detection, the experiments are conducted on a single dataset, which restricts the evaluation of the generalizability of our approach.

As for future work, we can create a user interface or a way to input a sample image and text and check if it is sarcastic or not using Django or Flask. This would prove to be much easier to test new cases or find out more errors in which the classifier can be incorrect.

References

Xiang Li Lin Gui Min Yang Ruifeng Xu Bin Liang, Chenwei Lou. 2021. InCrossMG: Multi-modal sarcasm detection with interactive in-modal and cross-modal graphs. In *In Proceedings of the ACM International Conference on Multimedia*, page 4707–4715.

Xiang Li Min Yang Lin Gui Yulan He Wenjie Pei Ruifeng Xu Bin Liang, Chenwei Lou. 2022. CM-CGN: Multimodal sarcasm detection via cross-modal graph convolutional network. In *In Proceedings of the Annual Meeting of the Association for Computational Linguistics*, page 1767–1777.

Wenji Mao Nan Xu, Zhixiong Zeng. 2020. DR-Net: Reasoning with multimodal sarcastic tweets via modeling cross-modality contrast and semantic association. In *In Proceedings of the Annual Meeting of the Association for Computational Linguistics*, page 3777–3786.

Joel Tetreault Liangliang Cao Rossano Schifanella, Paloma de Juan. 2016. Detecting sarcasm in multimodal social platforms. In *In Proceedings of the ACM International Conference on Multimedia*, pages 1136–1145.

Aniket Roy Shraman Pramanick and Vishal M. Patel Johns. 2022. Multimodal learning using optimal transport for sarcasm and humor detection. In *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 546–556.

Ruike Zhang Wenji Mao Yuan Tian, Nan Xu. 2023. DynRT: Dynamic routing transformer network for multimodal sarcasm detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, Volume 1 (Long Papers)*, page 2468–2480.