Meme Magic

1st Hrudaya Jinna
TA40935
hrudayj1@umbc.edu

2nd Aksheetha Muthunooru
ME52139
me52139@umbc.edu

3rd Yukta Medha
PD05177
yuktam1@umbc.edu

***Abstract*—In a time where social media and quickly changing internet culture rule the roost, it is critical to have fresh and interesting content. The MemeMagic is a random meme generator that makes use of sophisticated algorithms and machine learning methods to produce a wide variety of memes that satisfy the ever-changing preferences of internet users. The project leverages cutting-edge technology to seamlessly combine text and images, providing users with an endless source of amusement.**

I. Introduction

The MemeMagic is a cultural reflection of the internet age as well as an innovative piece of technology. By giving people access to a tool that goes beyond the boundaries of conventional content creation, it hopes to contribute to the vibrant and always growing world of memes. This study attempts to clarify the relevance and possible implications of the random meme generator in the larger context of internet culture by looking at the creation process, user interactions, and the influence on online communities.

This paper explores the idea, design, and execution of the memes, an initiative that uses deep learning and image recognition. The principal objective of this project is to provide users with a new and interactive tool for creating memes on the fly, in response to the increasing need for relatable and new content on the internet.

We will examine the underlying technologies that allow the Random Meme Generator to function as we delve into its inner workings. It can interpret textual inputs and choose images that enhance and complement the intended humorous effect thanks to machine learning algorithms and image recognition techniques. Furthermore, the ability of the application to adjust to user-specified parameters and thematic decisions guarantees a customized and entertaining meme-generation experience.

II. Related Work

A. *Dank Learning*

Dank Learning [4] has 3 types of CNN encoders + LSTM decoder with pre-trained GloVe embeddings. Dank Learning is the very first to combine pre-trained Global Vectors for Word Representation (GloVe) embeddings with Convolutional Neural Network (CNN) encoders and Long Short-Term Memory (LSTM) decoders. This method makes it possible to create memes that seamlessly combine textual and visual data. Dank Learning generates hilarious and contextually rich memes by using LSTMs for sequential text creation and CNNs for picture data processing. This pleasant balance between visual and textual aspects is achieved. Pre-trained GloVe embeddings improve the model's comprehension of complex text and help produce captions with a higher level of complexity of language.

B. *MemeBot*

MemeBot [5] is a "Transformer Models for Meme Captions Generation". Transformer models, which are known for their attention mechanisms and sequential processing powers, are applied by MemeBot to revolutionize the creation of memes. MemeBot's use of Transformers allows it to effectively capture long-range dependencies in captions and images. The attention processes improve the model's ability to concentrate on essential areas of a picture during the caption-generating process, resulting in a coherent and contextually appropriate output. The strength of this method is its ability to provide captions that are related to the visual material and adapt to a variety of meme formats.

C. *Memeify*

Memeify [7]is a Fine-tuned GPT-2 Language Model. Memeify takes an alternative approach by optimizing the powerful GPT-2 language model especially for the purpose of creating memes. GPT-2, known for its remarkable understanding of language, has been trained on a wide variety of online content, which makes it an excellent choice for identifying the casual and varied language included in memes. With Memeify, you can fine-tune the model to better understand and produce material that fits into the humor and context of online meme culture.

III. System Architecture

The user inputs data an image to be used for meme generation.The image provided by the user.The system uses a Flask API, which is a lightweight web application framework in Python, to handle requests. The system's output is a meme generated from the input image. In the backend, the dataset for training the machine learning models is sourced from memegenerator.net via web scraping. The images are then pre-processed, which could include resizing and normalizing. Data augmentation is applied to improve model robustness and prevent overfitting by transforming images in various ways. The dataset is split into training and testing sets, essential for model validation. Data preparation might include labeling and formatting data for the model. A trained model

is loaded, which is then used for making predictions, such as generating the meme text. There are options for choosing different types of word-based models for text generation. These include LSTM Decoders, which are recurrent neural networks suitable for sequences, and Transformers, which can either use a global image representation or consider the spatial aspects of an image. Similarly, character-based models are available, operating at the character level for potentially more nuanced text generation. These also include LSTM and Transformer models, with the Transformers considering spatial image features.

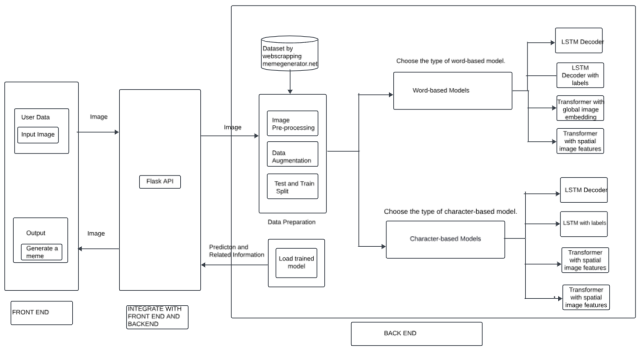

Fig. 1. Architecture of our service

IV. DataSet

We conducted the crawling and preprocessing of a extensive meme dataset, comprising 900,000 meme captions associated with 300 meme template images sourced from the MemeGenerator website. Throughout the data collection process, we ensured data cleanliness by removing obvious duplicates, outliers with excessively long captions, as well as non-ASCII symbols and templates in languages other than English.

A. Dataset Challenges

There are no established standard datasets for meme generation. The Dank Learning dataset by Peirson et al., 2018, includes about 400,000 captions but suffers from repeated memes, outliers, and imbalance. The Memeify dataset by Vyalla et al., 2020, comprises 1.1 million captions but is damaged and lacks special tokens for separating text blocks. Additionally, the Memebot dataset by Sadasivam et al., 2020, with 177,000 captions, has not been released.

B. Dataset Collection Process

The primary data source for the dataset is memegenerator.net. The collection involves web scraping and several cleaning steps. These steps include removing memes without text, captions with non-ASCII symbols, outliers with less than 10 or more than 100 characters, non-English text, and intentional punctuation repetitions. Captions with high similarity are also filtered out. In preprocessing, a '<SEP>' token is used for text block separation and an '<EMP>' token for empty text blocks.

C. Final Dataset Composition

The final version of the dataset shown in figure 2 contains 900,000 memes across 300 meme templates, averaging 3,000 captions per template. The data is divided into training, validation, and test sets, with 2,500, 250, and 250 captions per template, respectively.

Fig. 2. Final Dataset

V. Methodology and Implementation

A. Method

We planned to use various deep learning models but after researching a lot about which models are better for image processing and auto captioning of the image. We decided to explore Captioning LSTM, Captioning LSTM with Image-label Encoder, Base Captioning Transformer with Global image embedding, Captioning Transformer LSTM with Spatial image features and after through brushing of these models we decided to go ahead and use these models for our meme generator.

*1) **Captioning LSTM**:* A neural network architecture called the Captioned Long Short-Term Memory (LSTM) [2] model was created for natural language processing applications, especially the creation of evocative captions for pictures or videos. The model adheres to a methodical workflow by utilizing the sequential processing powers of LSTM, a member of

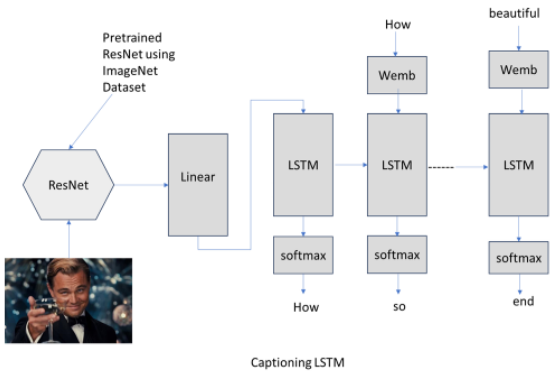

Fig. 3. Captioning LSTM

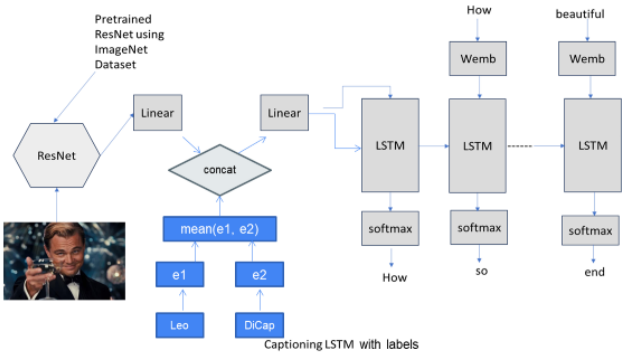

Fig. 4. Captioning LSTM with labels

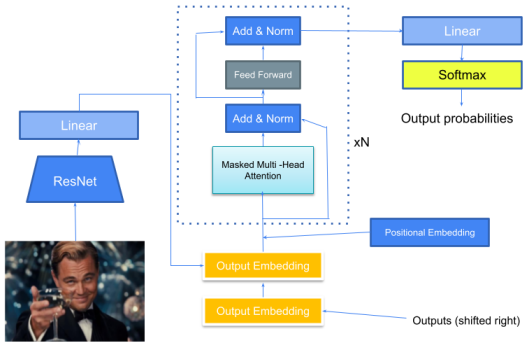

Fig. 5. Captioning Transformer LSTM with Global Embedding

the recurrent neural network (RNN) family. First, it uses a Convolutional Neural Network (CNN) to extract key elements from the visual data after receiving an image as input. The LSTM receives these properties as its initial inp ut. The LSTM unit is then used for sequential processing, repeatedly producing words for the caption while retaining a recollection of words that came before it to record contextual associations.

A probability distribution over the vocabulary is generated for the following word in the caption generation process, and this procedure is repeated until an end token or a predetermined maximum length is achieved. The model's parameters, such as weights and biases, are iteratively changed to reduce discrepancies between the generated and actual captions while it is trained on pairs of photos and their accompanying captions. Notably, captioned LSTM models are used in image captioning tasks, showcasing their ability to generate descriptions that are human-like on their own by identifying linguistic relationships and contextual subtleties in sequential data, such sentences or captions.

*2) **Captioning LSTM with Image-label Encoder**:* "Captioning LSTM with Image-label Encoder" [6] refers to a particular neural network architecture that is intended to be used in the process of creating picture captions. This paradigm refers to the encoder that processes the image input as a "Image-label Encoder." This encoder probably takes into account both the image's visual characteristics and extra information encoded in labels linked to the image.

The "Image-label Encoder" is a neural network architecture designed specifically to provide captions for images. It processes an image together with the labels that go with it in two steps. The Image-label Encoder uses a neural network to extract relevant characteristics, possibly combining a Convolutional Neural Network (CNN) for visual data and extra layers for label information. The purpose of combining the visual and semantic information is to provide the features a more comprehensive, contextual understanding. The resulting output consists of a representation that incorporates both picture visual features and label information. Next, integrated features from the Image-label Encoder are processed by the Caption Generator, which is driven by a Long Short-Term Memory (LSTM) network.

In order to produce captions during this sequential processing, the LSTM considers both the visual features and label-related data. Based on the integrated features from the Image-label Encoder and the words that came before it, the result is a logical string of words that make up the caption. The model is given image-label pairings and captions during the training phase. Then, the parameters of the LSTM and the Image-label Encoder are iteratively changed to minimize the discrepancy between the generated captions and the real captions in the training dataset. This architecture is used in picture captioning tasks, where the addition of more semantic information encoded in labels improves the model's ability to generate more complex and contextually rich captions that reflect a deeper grasp of the content.

*3) **Captioning Transformer LSTM with Global Image Embedding**:* An image captioning neural network architecture called "Base Captioning Transformer with Global Image Embedding" [3] was created specifically for this purpose. This model uses a transformer-based method to create captions for photos. It is based on the transformer design, which is well-known for its effectiveness in processing sequential data. Interestingly, it includes a Global Image Embedding method that processes an input image to produce a thorough representation that captures the context and overall information. This global picture embedding is smoothly incorporated into the transformer-based model, helping the caption generation process comprehend the image.

In order to reduce discrepancies between the generated and actual captions, the transformer and global image embedding components' parameters are iteratively adjusted during the training process, which uses pairs of images and captions. By providing a comprehensive representation of the input image, this design seeks to improve contextual awareness while utilizing the advantages of transformers for sequential processing.

*4) **Captioning Transformer LSTM with Spatial image features**:* A neural network setup called the "Captioning Transformer LSTM with Spatial Image Features" [8] was created specifically for the purpose of creating captions for

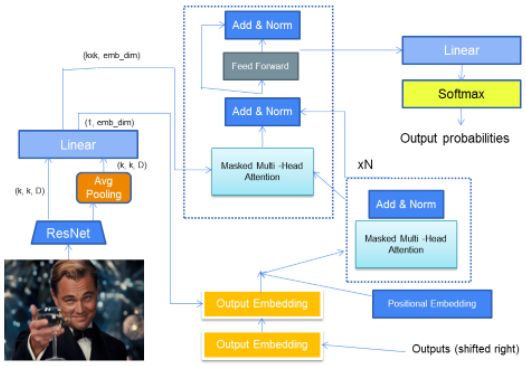

Fig. 6. Captioning Transformer with Spatial Image features

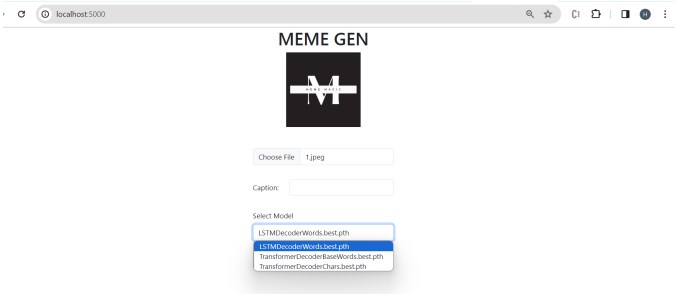

Fig. 7. Front End of our website

pictures. This model is composed of both transformer and LSTM elements, taking use of the transformer's expertise in handling sequential data and the LSTM's capacity to identify dependencies over long sequences. The model gains an essential dimension with the addition of spatial image characteristics. Convolutional Neural Networks (CNNs) are one type of feature extraction mechanism that is used to extract spatial features and patterns from images through input processing. The spatial picture characteristics are then concatenated with tokenized sequences or added as an additional input stream, and smoothly integrated into the transformer-LSTM architecture [1].

By taking into account both the sequential context and the spatial arrangement of visual elements throughout the caption generation process, this integration improves the model's contextual awareness. The model is trained on image-caption pairings, and iterative parameter adjustments are made with the goal of minimizing differences between the produced and real captions. To summarise, this architecture provides an all-encompassing method for captioning images by merging the benefits of transformer-LSTM sequential processing [9]with the spatial insights obtained from picture characteristics, resulting in enhanced captioning performance.

VI. Front End

Users can interact with the system by uploading images through a user-friendly interface, which includes an upload button linked to RESTful APIs. The front-end is programmed in HTML, allowing users to input text and select from multiple methods before submitting. The system then generates and displays a meme image based on the input text, offering an option to download the generated image.

VII. Back End

The interface provides users with the functionality to select an image from a designated folder, as illustrated in Figure 8. After choosing the image, users have the flexibility to opt for one of three methods: LSTM Decoder Words, Transformer Decoder Base Words, or Transformer Decoder Chars. Upon making their selection, they can proceed by clicking the submit button. For seamless interaction between the front-end and back-end, enabling image uploads and retrieval of generated captions, the development of APIs using Flask is essential. Figure 9 shows the generated meme after interacting between the backend using Flask API.

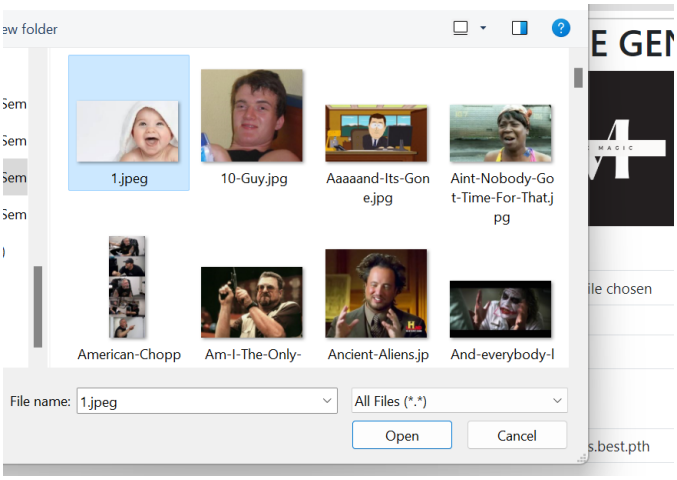

Fig. 8. User has the option to upload an image from their own PC

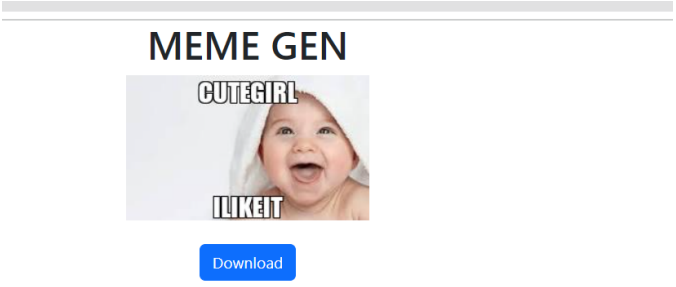

Fig. 9. Genertaing a meme From our model

Fig. 10. Display a meme after model training

VIII. Results

A. Models based on LSTMs outperform models based on transformers

After experimenting, it was found that LSTM-based models performed better than Transformer-based models in tasks involving the creation of memes. The LSTM architecture performed better at creating consistent and fitting meme captions. It is well-known for its sequential processing and capacity to detect long-term dependencies in data. Memes' complex vocabulary, which is frequently characterized by humour and cultural references, fits in nicely with the advantages of LSTM models, which helps explain why they perform better than Transformer-based models.

B. Similarities between the Image Only and Image Label LSTM Models

Upon comparing models built for image-only and image label inputs, one important observation was how similar their results were. In the creation of memes, both LSTM models performed similarly irrespective of the kind of input. This shows that memes' visual components may be handled by the LSTM architecture, either directly through images or indirectly through labeled data.

C. Base Transformer performed better than one with spatial features

When it comes to Transformers, we noticed that the base Transformer model performed better than its counterpart enhanced with spatial features.

Overall, this system integrates front-end design, back-end API development, and advanced machine learning models to create a dynamic and interactive meme generation platform.

The Figure 11 and Figure 12 show the random memes generated from the models.

Fig. 11. Random Meme generated from our model from the input text "Hi"

Fig. 12. Random Meme generated from our model from the input text "Cute"

IX. Future Scope

Firstly, there's potential to enhance meme generation accuracy using diverse model sets, given that LSTM models have shown promising results. The project aims to generate more coherent captions and potentially surpass the quality of human-generated memes. Additionally, exploring transformer-based architectures alongside LSTM models could offer new insights and improvements. Another aspect is the effectiveness of text labels on LSTM-based models and the comparative performance of word-level versus character-level tokenizations. The project also plans to refine its dataset, possibly expanding beyond the current 200 meme templates and 900,000 memes, to create a more comprehensive and balanced collection for training and testing. The future work will likely involve more complex experiments with hyperparameters and model structures to achieve these goals.

X. Conclusion

In conclusion, our experiments shows how well the LSTM model performed in capturing patterns and producing coherent captions, highlighting their accuracy in meme generation. But it's important to remember that human-generated memes still have a distinct charm despite the advances in AI-driven meme production. Current models are unable to adequately reproduce the natural creativeness and grasp of context that humans can provide. Even while our models show promise, human-generated memes' inventive and dynamic quality continues to set the standard for humor and cultural relevance.

References

[1] Guang Li, Linchao Zhu, Ping Liu, and Yi Yang. Entangled transformer for image captioning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8928–8937, 2019.

[2] Xiangyang Li, Shuqiang Jiang, and Jungong Han. Learning object context for dense captioning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8650–8657, Jul. 2019.

[3] Hashem Parvin, Ahmad Reza Naghsh-Nilchi, and Hossein Mahvash Mohammadi. Transformer-based local-global guidance for image captioning. *Expert Systems with Applications*, 223:119774, 2023.

[4] Abel L Peirson V and E Meltem Tolunay. Dank learning: Generating memes using deep neural networks. *arXiv preprint arXiv:1806.04510*, 2018.

[5] Aadhavan Sadasivam, Kausic Gunasekar, Hasan Davulcu, and Yezhou Yang. Memebot: Towards automatic image meme generation. *arXiv preprint arXiv:2004.14571*, 2020.

[6] Jinsong Su, Jialong Tang, Ziyao Lu, Xianpei Han, and Haiying Zhang. A neural image captioning model with caption-to-images semantic constructor. *Neurocomputing*, 367:144–151, 2019.

[7] Suryatej Reddy Vyalla and Vishaal Udandarao. Memeify: A large-scale meme generation system. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pages 307–311. 2020.

[8] Chi Wang, Yulin Shen, and Luping Ji. Geometry attention transformer with position-aware lstms for image captioning. *Expert systems with applications*, 201:117174, 2022.

[9] Jincheng Zheng and Chi-Man Pun. Hybrid-spatial transformer for image captioning. In *Proceedings of the 5th International Conference on Control and Computer Vision*, pages 22–28, 2022.