Hateful Memes Classification using Machine Learning

Jafar Badour
Artificial Intelligence in Games Design Lab
Innopolis University
Innopolis, Russia
j.badour@innopolis.ru

Joseph Alexander Brown
Artificial Intelligence in Games Design Lab
Innopolis University
Innopolis, Russia
j.brown@innopolis.ru

***Abstract*—Several studies produced sophisticated models for sentiment analysis of textual data, and many others tackled feature extraction from images. However, far fewer studies focus on the multimodal representation of data, namely the information that consists of multiple channels. In this work, we focus on the classification problem of multimodal data. Memes comprise a visual image and a textual caption. This work is dedicated to classifying hateful memes and this work proposes two approaches to solve the multimodal classification problem. First, converting the visual channel into a textual one and feed it to textual classifiers. The other approach, which yielded superior results, converted both channels into a vector representation and then combined them to represent the visual-textual context. This work is a consequence of the Facebook Hateful Memes challenge. The model developed in this work managed to rank 32 among 3172 competitors in the challenge. The model is implemented with no domain knowledge or understanding of hate speech. This model performed well in the Facebook Hateful Memes challenge dataset and a novel dataset that we created to prove the consistency of generic models over other models that are structured according to domain knowledge. In contrast to the top solution in the Facebook Memes Challenge, this work provides a generic approach, without hard-coding rules ahead of training or validation, that is able to learn the hatefulness definition from any dataset. A novel dataset that comprises hateful memes retrieved randomly from the web is described in this work, which is used as another dataset to test approaches generality.**

***Index Terms*—Machine Learning, Facebook Hateful Memes Challenge, AI, Artificial Intelligence, Deep Learning, LSTM, SBert, Multimodal, Classification**

I. Introduction

Humour has evolved following the considerable expansion of the internet. A meme, which stems from the Greek word mimema, meaning imitated, describes a humorous image exchanged between people. According to Newman et al. [1], the attention span of people has decreased following the technological revolution, boosting memes' popularity. A web surfer would prefer to see a meme rather than reading an article or watch the news. We can view memes as viral particles that are unable to be contain after an outbreak. Memes can be duplicated and replicated on the fly; there are even automated engines to spread memes on social media. Stopping such actions is expensive because we would need to hire millions of people to monitor the internet. The need for automating the detection of hateful memes becomes more urgent as memes supporting radical movements are growing exponentially.

A meme's structure usually includes a picture and a text. This hybrid structure makes processing and extracting information difficult since it adds another layer of complexity to the data. Classification of memes brings two classic problems to light. The first problem is image captioning which describes an image. The second problem is the classification of the textual information along with the data extracted from the image. The first problem is the hardest due to the limitation of popular datasets and the complexity of the field. It is hard because a picture can represent anything, and it is impossible to compile a dataset that contains every possible object or person. The objective is that given a meme, decision can be made as to whether it promotes hate speech toward a specific group of people or an individual. This work explains how deep learning can interpret visual and textual information for hate speech detection for memes. This paper is structured as follows: The second sections presents an overview on the related works about classification of multimodal data and the winning solution in the Facebook Hateful Memes challenge. The third section provides a context on understanding hate speech and the distinction from free speech. The fourth section presents two approaches to solve the multimodal classification problem. It introduces the creation of a novel dataset for the purpose of testing the ability of approaches to learn from different data. The fifth section showcases examples of hateful and benign memes as well as results of the implemented approaches on two different datasets. Machine Learning models that do not rely on pre-written rules for classification perform worse than general models that rely solely on the data.

II. Related Work

Memes comprise a strict structure, and the amount of information in the visual and textual parts is low [2]. Recent attempts to acquire domain knowledge in memes classification and clustering proved difficult because memes span human knowledge. Memes that contain technical or science knowledge are popular, i.e. those memes ridicule the differences between programming languages. Semantic analysis of textual tweets proved to be complex [3] because of having a tremendous amount of tweets irrelevant to the task at hand. For example, multiple tweets are formal and informative. Therefore sentiment analysis problem over restaurant reviews would not

benefit from such tweets. Datasets for informative and good-quality memes are scarce, making the research difficult due to the lack of resources. Multimodality has been a focus of multiple works, and certain methods try to solve the problem such as *fusion* and *pre-training*. Section **A** describes related works on the aforementioned methods. Section **B** focuses on approaches that are directly applicable in the domain of memes classification. Section **C** describes the recent Hateful memes challenge solution.

A. Fusion and Pre-Training: Approaching Multimodality

Baltrusaitis et al. [4] categorize the Visual-Linguistic (VL) multimodal fusion into three categories: early fusion, late fusion, and hybrid fusion. Early fusion combines the features of all the modes upon extraction. Late fusion combines the results for each mode [5]. Lastly, hybrid fusion fuse accumulates the results from individual unimodal predictors and early fusion. The pre-training comprises unimodally pre-trained and multimodally pre-trained. A unimodally pre-trained language and vision model combined with different fusion types is called a unimodally pre-trained multimodal [6].

B. State-of-the-art on Memes Classification

One of the classic tasks for VL multimodal exploration is to comprehend the correlation between multimodal feature spaces. A Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN), when trained together, learn a composite embedding space from combined multimodal VL data. This architecture is specifically common when viewing the literature on image captioning [7]–[9]. Visual Question Answering (VQA) merges both VL modalities to infer the correct answer rather than learning a mapping between spaces. Delicate Correlation Modelling [10] between image and question representation is required for the aforementioned problem. In hateful memes, we need similar accurate correlation modelling between visual and textual data to find a suitable mapping to both VL modalities and finally make a decision as to the classification. Recently, the focus diverged to cross-modality by multimodal pre-training approaches like Visualbert [11], and UNITER [12]. These approaches performed well on many recent approaches on multiple VL multimodal datasets such as VQA [10], Visual Commonsense Reasoning (VCR) [13], NLVR [14], Flickr30K [15], and many more.

Feature engineering domains influenced researchers to develop various techniques and classifiers. Traditional approaches in feature engineering consist of N-grams, BOW, POS, TF-IDF, CBOW, word2vec, and text features. These conventional techniques were extensively employed for hate speech detection and sentiment analysis in general. Some of the most recent and standard algorithms is Support Vector Machines (SVM), Random Forest, Decision Tree, Logistic regression, and Naive Bayes [16]. In contrast to different problems in Natural Language Processing (NLP), hate speech is subject to geographical, religious, and cultural backgrounds. Hate speech evolves with time, as well as a hierarchy in society. These implications can affect the view on a particular type of speech, whether it is hateful or not [17].

C. Facebook Hateful Memes Challenge

The following is a breakdown of the structure of the winning solution in the Facebook Hateful Memes Challenge.

1) Caption the image via Visual-BERT: First, a caption of the image is extracted using a pre-trained UNITER model. The captions are fed to a pre-trained ERNIE-ViL [18] model to create a knowledge graph. A knowledge graph is a graph that links objects with relations. For instance, "A girl is riding a horse". The information in the knowledge graph will be as follows:

- "girl" : object detected by the model
- "horse" : object detected by the model
- "is riding" : is a relation between two objects.
- "on top of" : is a relation extracted from the picture with no relation to the captioning text.

Keywords (objects) from ERNIE-ViL model are combined with race tags that are obtained from using a pre-trained FairFace [19] Classifier and web entities results from google cloud vision models. Each entity or identification is paired with a segment of the image. VL-Bert [20] is used to combine each entity with its accompanied component to produce a vector representation of the meme. The author then uses K-means clustering to cluster memes based on racism, sexism, etc. Similarity model is then used to classify memes.

III. Defining Hate Speech

This section contains: **A**) A summary description of the literary work on hate speech, **B**) A definition of hatefulness, which is used in the creation of the novel dataset used in this work, **C**) A context of how speech hatefulness varies with time and place, and **D**) A proposed distinction between free and hateful speech.

A. Related work on hate speech

Literary work on the definition of hate speech is scarce and not standardized. An early form of censoring hate speech on the world web is to have a blocklist of words where any text containing these words would be banned. This method is effective against concise hateful text (flames). Razavi et al. [21] looked into flames and produced a three-stage classifier. Their primary method is to use a dictionary of 2700 blacklisted words to detect flames. Flames usually comprise a short text, usually as a comment in a group chat, and it is different from the kind of texts in memes [21]. Definitions vary between researchers, Warner and Hirschberg [22] reduce the problem of hate speech classification into hateful and non-hateful. However, Malmasi and Zampieri [23] follow a different approach where they categorize any type of speech into three categories: hateful, offensive, and non-hateful. Warner and Hirschberg [22] debunk the obsolete keyword-based hate speech detection algorithms. They differentiate between sentences that have a potential hateful or insensitive word and those that carry a hateful meaning. Words such as the N-word

are usually flagged as inappropriate. In some conditions, some communities reclaimed such words. If the individual belongs to such a community, the word must not be deemed a sign of hatefulness. Other reclamation examples contain "queer", "nerd" and "nigga" [22].

B. What is Hatefulness?

A speech is deemed hateful when a significant number of people find it so. Since it is hard to draw a line between what is hateful and what is not, we think this is the best definition so far that can be good enough for any time period. This is the definition of hatefulness used in this work. This logic stems from the judiciary system, where the action is considered a crime when the majority of the jury thinks it is. The laws of each country vary according to the major religion, ethnic background, economic status, historical background, and political landscape. For instance, the thumbs-up sign is deemed benign in most western cultures. However, in some middle-eastern cultures, such as Iran, it represents the western middle finger sign and is viewed as highly offensive. The hatefulness of an action or a certain text depends on the culture, society, community and the individual. Some people find certain speech hateful while others do not. We argue that if most people think of a specific type of speech as hateful, it is hateful. It is impossible to create a dataset where the annotators are a representative sample of the population. Different population clusters have a different definition of hate speech according to multiple factors, including the country or the ethnic identity. These different variations make the problem of defining a universal understanding of hate speech impossible.

A hateful word does not constitute a hateful tone of speech. For example, "nigga" is a word that conveys solidarity in the African American communities. We can see that the N-word here does not represent hate speech since it is present in an academic context and not pointed to attack anyone. It can be argued that this example is hateful, and "N-word" can replace the word "Nigga". We think that such a context is not considered hateful by the majority of the population. It is imperative to point out that we cannot determine what the majority of the population thinks ; therefore, in creating the **Innopolis Hateful Memes Dataset** such speech is considered benign.

C. Hate speech evolutionary nature

Christiansen et al. [24] found that languages go through a natural selection process, and the main concepts of the fitness of a word or an expression is its ease of pronunciation, contextual complexity, ability to be understood by the majority of the population. Mutations can be viewed in the lexical domain where a particular word can point to a different meaning and a word can be written or pronounced differently by other populations. Languages can influence one another, and a typical example is the influence of Arabic on the Spanish language, i.e. Spanish was changed due to the Moorish conquest [25]. A hateful sentence by modern-day standards can be considered benign in the nineteenth century since the historical consequences are crucial to provide a context to that speech. For instance, a picture of the Nazi army with a caption of "Ah shit, we did not turn on the gas" does not make sense unless the historical context is widely available. A great example of hate speech definition's evolutionary state is the Swastika. Swastika existed long before the Nazis as a religious symbol of Buddhism temples [26].

D. Hate speech distinction from free speech

Word reclamation is a common defence mechanism. Herbert and Cassie [27] argues that reclamation occurs when a group of people reclaim a word and strip it of its hateful nature. One famous example is the N-word, which has been reclaimed by the disseminated populations, namely the African American community in the United States. This word meaning evolved to show group solidarity. In the U.S., many people consider the N-word hateful when it comes out from a non-African American. Since we cannot determine the identity of the meme author, we will not consider it in our evaluation. In application, online platforms cannot consider the user's identity for many reasons, including privacy, identity-check of the users and different legislation between nations. These constraints affect the system's precision since reclaimed words can be taken out of context, therefore, deemed hateful. The recall, which is the fraction of true positives over true positives combined with false negatives, preference over precision would ultimately knock down free speech. The propagation of hate speech in social media is fast due to modern technological advancements, and it would damage minorities inciting hateful behaviour in individuals across virtual platforms. In reality, a person spreading racist propaganda on the street can be stopped by others speaking out to persuade the audience with the malicious intent of the propaganda. However, that is not how it works on social media where the author has access to tools of the environment where they can delete comments that they do not like or simply create fake users to simulate a Bandwagon effect [28], where people fall under the peer pressure to believe an untested hypothesis. In conclusion, achieving higher recalls on the classification problem of detecting hate speech would ultimately lead to higher false positives rates in free speech. Constraints on the contextual basis for memes detection also affect the false positives rate.

IV. Methodology

We examine the structure of the classification system as well as the datasets that we used in this work. The system implements two different approaches to solve the meme classification problem using state-of-the-art image and textual feature extractors.

A. Data Collection

1) Facebook hateful memes dataset: The data is provided as a part of the Facebook Hateful Memes challenge [29]. However, other datasets are used to pre-train the models

needed in the pipeline. The images from facebook consist of two channels:

- Visual: The image itself.
- Textual: The text extracted using state of the **ocr.space** object character recognition system.

2) COCO dataset: COCO dataset is a diverse and large dataset for the following purposes: object detection, segmentation and captioning. The set has 330,000 images with over a million captions supplied. There are five captions per image and other features such as Superpixel stuff segmentation and 250,000 people with *key points* [30].

3) The Innopolis Hateful Memes dataset: Hateful memes classification is a cutting-edge field. It is difficult to find huge datasets regarding hateful memes due to the novelty of the topic. That is why we decided to create our Innopolis Hateful Memes Dataset, which contains 23,000 Memes from multiple websites, including Memedriod, Twitter, and Duckduckgo. The dataset can be freely downloaded here [1].

The dataset labelling is carried out as follows:

- Five independent annotators visualize the meme and decide whether it is: 1- Hateful, 2- Not Hateful, 3- Not a meme. If an image was labelled as non-meme, we omit it from the dataset.
- The annotators were lectured extensively about what we consider as hateful.
- Each image is considered hateful if three out of five annotators classified the corresponding image as hateful.
- If an image is classified as "Non-meme" from at least one annotator it is omitted out from the dataset.
- If a meme contains non English text in a way that it is not understandable by non English speakers the annotators will classify it as non-meme.

This dataset is used to train multiple models including state of the art models and will be used in the following sections to compare different models.

B. System Pipeline

We tried two different approaches to solve the problem:

- Visual attention and Textual Classifiers.
- Object detection and MultiModal Classifiers.

1) Visual Attention and Textual Classifiers: In 2015, a paper by Vinyals et al. [31] proposed an approach to use the time series models for images. These time series models, especially Long Short Term Memory networks (**LSTM**), are used for problems that input textual data. Usually, the text is tokenized and embedded and then is fed into LSTMs in a token-at-a-time fashion. The LSTM network memorizes the states of the past iterations. LSTMs can be used in a multitude of applications. It is used in image captioning, where the model is fed an image and outputs a description of that image.

Their approach maximizes the probability of annotating the image with the correct description.

[1]https://github.com/JafarBadour/Hateful-Memes-Classification

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{(I,S)} p(S|I;\theta)$$

Where θ is the parameters for the model, S is the sequence (e.g. caption), and I is the image fed into the network. S is a caption where its length is unbounded. The chain rule is valid to calculate the probability for the sequence given an image.

$$p(S|I) = \prod_{t=0}^{N} p(S_t|I, S_0, ..., S_{t-1})$$

Since calculating the product is infeasable it is possible to get the logarithm of both sides of the equality. Getting the following equation:

$$log(p(S|I)) = \sum_{t=0}^{N} log(p(S_t|I, S_0, ..., S_{t-1}))$$

Training set consists of a collection of (S, I) pairs. Stochastic gradient descent is used to maximize $log(p(S|I))$

It is common to model $p(S_t|I, S_0, ..., S_{t-1})$ as a Recurrent Neural Network (**RNN**), where all the states that represent tokens from 0 to $t-1$ are expressed by the hidden state of memory h_t. This hidden state is updated every iteration where the model predicts the next token in the sequence. The new token x_t is fed into the model and the next hidden state is calculated as follows:

$$h_{t+1} = f(h_t, x_t)$$

Where f is the LSTM network, which has shown great performance on sequence tasks such as translation. Images are fed into a sequence of Convolutional Neural Networks (**CNN**) with batch normalization. The words are represented with **Glove** embeddings [32].

The core of the LSTM model is the memory cell c, and this cell captures the knowledge of the model. See figure 1.

There are components of the LSTM that control the behaviour of the memory cell called "gates." These layers can store either a value from the gated area of zero, depending on whether the gate is one or zero.

In figure 1; The memory cell c is controlled by three gates. Blue represents the recurrent connections. The m is the output for the time iteration $t-1$ is fed back to the memory at time t via the three gates, and the cell value is fed back through the "forget" gate. Both m at time t and predicted word at time $t-1$ are fed to the softmax layer for prediction purposes.

Training: multiple cells of LSTMs are stacked and fed the same input. In the beginning, all the cells are fed the image, then they are fed tokens of the output (predicted) sequence one at a time (the output of the previous time iteration is fed into each cell as well).

Inference: the image is fed in the beginning to the cells, and then the previous predictions are provided one at a time. The paper uses **beam search** which is to keep the best k sentences at time $t-1$ and then to predict for each of these inputs (the corrections are fed back into the network, and therefore the

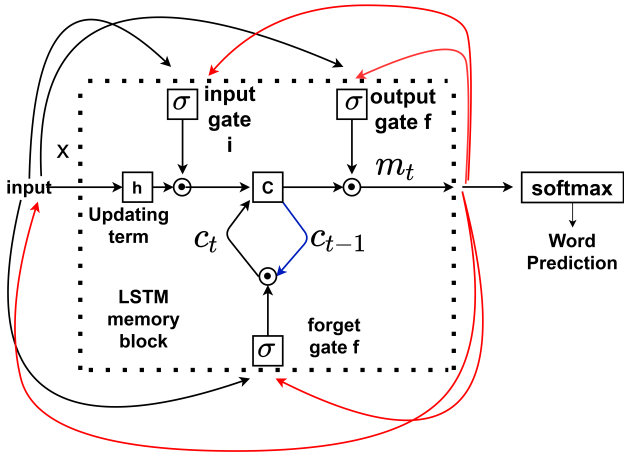

Fig. 1. LSTM: main architecture [31]

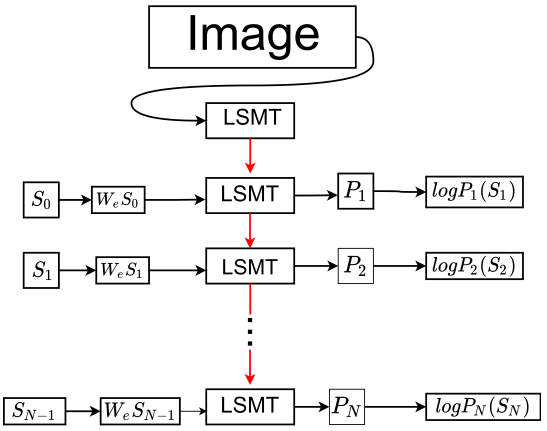

Fig. 2. LSTM [31]: Image is fed and then the tokens of the sentence are fed subsequently.

network could predict multiple outputs). The best sentence is $S = \text{argmax}_{S'}\, p(S'|I)$. The best sentence then is returned as an output.

2) Feeding captions to a text classifier: We established the image caption model as a good source for getting the information from the image and create a sentence that represents this information. The main idea for classifying hateful memes is to combine the visual content and the textual content and feed it to a textual classifier to understand whether the meme is hateful or not.

3) LSTM for text classification: We used the **Glove** embeddings to represent the words in the tokens from each channel: the textual channel and the caption from the visual channel. Both are concatenated and then fed into an LSTM layer which fed into another LSTM layer. Finally, the output is fed into a fully connected layer with sigmoid activation.

We do not create embeddings for the words of both image caption or the meme text. Instead, we import embeddings of the **Glove** embeddings which are trained using huge models on massive datasets. Mainly the purpose is to maximize the probability of hatefulness of the meme.

$$p(C, T|\text{hateful}) = \sigma(f(f(emb(C + T))))$$

where σ is the sigmoid activation, f is the LSTM net and emb is the embedding relation.

The loss function we chose is the binary cross entropy

$$L(\sigma(C, T, \theta), y) = ylog(\sigma(C, T, \theta)) + (1-y)log(1-\sigma(C, T, \theta))$$

We find the best parameters for the classification model using:

$$\theta^* = \underset{\theta}{\text{argmin}}\, L(\sigma(C, T, \theta), y)$$

It is crucial to note that using an LSTM rather than a Bidirectional LSTM resulted in inferior results that is why we chose to work with Bidirectional LSTMs. The main difference is that the activation for each cell becomes bidirectional instead of unidirectional. The input would be fed to the cell from left to right and from right to left.

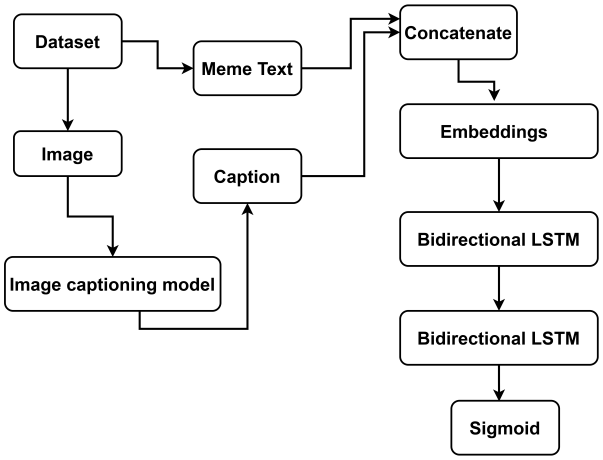

Fig. 3. An overview of the pipeline: First the meme is fed into the image captioning model then the caption and the meme text are concatenated and fed into a two layered bidirectional LSTMs which finally is fed into a fully connected layer with a sigmoid activation.

4) Training in the system: The image captioning model and the text classifier are trained independently:

- Image Captioning model: Is trained using datasets such as **COCO** [30] and **Flickr 30k** [15] datasets. The training process is as follows: randomly selecting an image and one of its annotations and train on the pair image, annotation.
- Textual classifier that comprises double bidirectional LSTM layers uses two datasets: The Facebook Hateful memes dataset [29] and the Innopolis Hateful Memes dataset. The input is the output of the image captioning model and the meme text, whereas the output is the probability of hatefulness.

C. Object detection, MultiModal Classifiers

We used another approach. This system consists of:

- Segmentation model.
- Xception Model.

- Sentence to embedding model.
- Fully connected model and sigmoid.

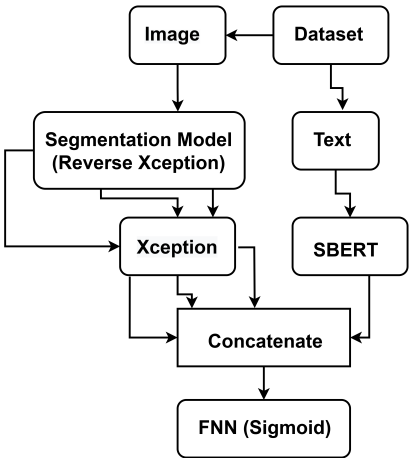

Fig. 4. A high level overview of the classification system of the second approach. It consists of 1) Segmentation model. 2) Xception Model. 3) Sentence to embedding model. 4) Fully connected model and sigmoid.

In figure 4 each meme's image is fed into a segmentation model and then the biggest three objects are fed into an Xception model for feature extraction. At the same time the textual channel of the meme is fed into Sentence Bert model and then all features from visual and textual channels are concatenated and fed into a fully connected Neural network.
We used segmentation model based on Xception [33] augmented pretrained model. According to He et al. [8] in their publication: Deep Residual Learning for Image Recognition where they found an intelligent way to train extremely deep neural networks by introducing skip connections.

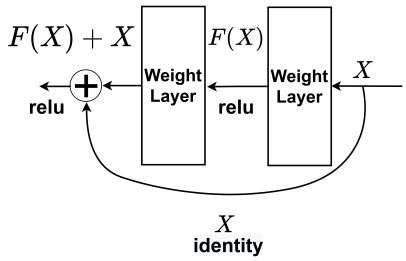

Fig. 5. A skip connection to provide the deeper layer with the sum of both the output of the its previous layer and an output of a shallower layer [31]

The model is based on the Xception model. We use a similar model for segmentation. However, we find the pixels that triggered the activations and provide a segmentation on these pixels.
The next step is to crop each connected component of pixels and feed it to the Xception model for object detection. Then the top 3 segmented images are fed into the pre-trained Xception model. The output of Xception from the segmented images and the original image are concatenated with each other and the output of **Sentence BERT** (SBert) [34].

The main idea of SBert is to input two sentences that denote the same meaning (paraphrased sentences). And SBert's goal is to find an embedding to each of these sentences such that the cosine similarity is minimal as possible. Mainly inputting u and v sentences, and the goal is to minimize.

$$L(u, v, \theta) = \frac{f(\vec{u}) \cdot f(\vec{v})}{||f(\vec{u})||.||f(\vec{v})||}$$

Finally, this output is fed into a fully connected layer, first with a Relu function and last with sigmoid activation. By doing so, SBert manages to find embedding of a sentence. This embedding is used in our system features of the sentence and concatenated with the features of the image and the segmented cropped images.

$$p(I, T|Hateful) = \sigma(Relu(Xc(I), Xc_3(I), Sb(T)), \theta)$$

Where Xc is the Xception net, Xc_3 is a tensor that comprises features of the top three segments by area, Sb is SBert net, and θ is the parameters of the rest of the network.

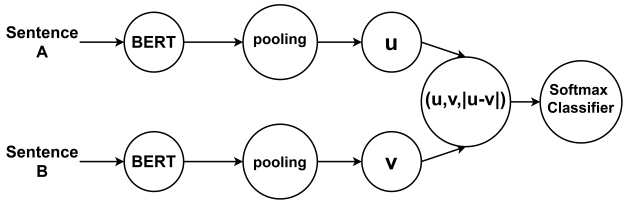

Fig. 6. The architecture sentence Bert that utilizes pooling to create an embedding space for phrases.

The features of the sentence are extracted using Sentence Bert and concatenated to the results of the earlier features. The textual and visual channels are split: The visual (the image) is fed into the segmentation model to retrieve the top segmented objects. These segmentations are split into different images. The features of the segmented images and the original image are concatenated in a long tensor. Then a few (three) fully connected layer with a Relu activation gets the concatenated results, and the final one has a sigmoid activation.

V. Evaluation and Discussion

Evaluation occurred using the two datasets available for this project. The Facebook hateful memes dataset and the dataset we created **"Innopolis Hateful Memes dataset"**. In table 2, results of the Sbert + Multimodal Classification Model. Table 3 contains the results of testing the model. Table 4 and Table 5 represent the training and testing results for the Visual Attention + Textual Classifier model.
A training iteration consists of:

1) Get a batch of images / image captions / text and load it to memory.
2) Train the model for two epochs

A. Evaluation of models

The primary metrics used are Accuracy and Area Under the Curve (AUC). The following table contains the results of running four iterations of training over the dataset.

TABLE I
COMPARISON WITH TOP CHALLENGE SOLUTION

Model	Dataset	Acc	Auc
Top Solution in FB challenge	Fb	**0.89**	**0.8827**
SBERT+ Multimodal Classifier	Fb	0.75	0.794
Visual Attention + Textual Classifier	Fb	0.642	0.648
Top Solution in FB challenge	Innopolis	**0.735**	**0.8209**
SBERT+ Multimodal Classifier	Innopolis	0.722	0.8
Visual Attention + Textual Classifier	Innopolis	0.648	0.64

The percentage of dataset for training is 80% for all approaches for each dataset.

B. Examples

An example of a False Positive using the textual classifier in the model "Visual Attention + Textual Classifier" in figure 7. The image captioning model predicted a wrong caption which made the textual classifier to predict as "hateful". In figure 8, the activations worked properly. In figures 8 and 7 below the red highlight denote the probability that the model will classify the meme as hateful whereas the blue highlight denotes the probability that the model will classify the meme as benign.

In figure 7, the word red triggers a benign response despite it is preceded by *skull* due to the bidirectional nature of Bidirectional LSTM. It linked the adjective red to the *cup* instead of *skull*.

Fig. 7. One example of a false positive meme.
Activations of LSTM on the right after each word. No skull appears but car is detected. The word skull triggers the model to classify the image as hateful whereas red, which is related to cup, triggers a benign response.

C. Discussion

1) SBERT+Multimodal outperforms VA+Text Classifier: The image captioning model needs more training to be able to capture better captions. The results of the "Visual Attention + Textual Classifier" model are inferior to the "SBERT+Multimodal Classifier" model. Transforming the meme into textual content is a complex task, and the discrete nature of text increases the difficulty of training and negatively affects accuracy. However, using Sentence Bert to retrieve a vectorized sentence representation combined with other features extracted from the image provides the model with a continuous data flow. It increases the general accuracy of the Area Under Curve metric.

Fig. 8. One example of a hateful meme and a true positive prediction. Activations of LSTM on the right after each word

2) Facebook Hateful Memes Challenge Retrospect: The Facebook Hateful Memes Challenge Dataset is over-sampled with instances of racism, sexism, Islamophobia, etc. However, the natural flow of memes online is different. The data collected in the tables of the Innopolis Hateful Memes Dataset is retrieved without oversampling from various websites that do not censor hateful memes. In the winning solution, [35] in the Facebook challenge. The author developed different classifiers for different classes of hate speech and relied on the dataset's oversampling nature to significantly increase accuracy. The solution still managed to outperform the top model provided in this work, however, the accuracy and AUC gaps diminished drastically. This is because of the added error of implicit bias due to hard-coding different types of hate speech.

VI. Conclusion and Future Work

This work provides proof that generic models that do not consider the domain knowledge of the problem can perform as well as crafted models structured after a deep understanding of the domain. This work introduces a new dataset composed of a randomized sampling of websites that are famous for being meme-hubs. This dataset consists of well over 20,000 memes, with 13% of them being hateful. The dataset also offers researchers more detailed information about each meme. In contrast to the Facebook Hateful Memes dataset, our dataset provides words bounding boxes in the image itself. The newly collected dataset is crucial to test the hypothesis: Models created to detect certain hate speech types perform poorly when applied upon images randomly retrieved from the internet. A few solutions of the winners in the contest [35] used this mechanism of relying on detecting types of hate-speech forcing implicit bias in their models.

Using segmentation to split the image into multiple images with lower complexity produced superior results over-

generalized image captioning and textual classifiers. This is because the meme's meaning is usually related to the most visible object in the meme. It was extracting the most salient objects that produced superior results. SBert provided a vectorized format of the meme's text. SBert helped make satisfying results since we used the paraphrase encoding for sentences. This is due to observing the meme's life cycle, where an individual meme can be replicated and have its text paraphrased.

This work also provides a free software solution [2] that helps to annotate images, as well as, software to collect images from the web efficiently. The same link contains a description of the novel dataset used in this work. It is highly recommended for other researchers to use this dataset and propose feedback to improve it.

Future work can expand multiple dimensions of this work. The Innopolis Hateful Memes dataset can contain more samples and be labelled by more annotators. Sentence BERT can be replaced by another model that encodes texts and provides a vectorized text representation. The multimodal classifier can take more objects and concatenate their Xception representation before feeding them into fully connected layers.

ACKNOWLEDGMENT

Many thanks to the analysts that helped establish the Innopolis Hateful Memes Dataset: Mohammad Ahmad, Salim Hanna, and Suliman Badour.

REFERENCES

[1] M. Z. Newman, "New media, young audiences and discourses of attention: from sesame street to 'snack culture'," *Media, Culture & Society*, vol. 32, no. 4, pp. 581–596, 2010.

[2] F. Yus, "Multimodality in memes: A cyberpragmatic approach," in *Analyzing digital discourse*. Springer, 2019, pp. 105–131.

[3] X. Liu, K. Li, M. Zhou, and Z. Xiong, "Collective semantic role labeling for tweets with clustering," in *Twenty-Second International Joint Conference on Artificial Intelligence*. Citeseer, 2011.

[4] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 2, pp. 423–443, 2018.

[5] T. H. Afridi, A. Alam, M. N. Khan, J. Khan, and Y.-K. Lee, "A multimodal memes classification: A survey and open research issues," *arXiv preprint arXiv:2009.08395*, 2020.

[6] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine, "The hateful memes challenge: Detecting hate speech in multimodal memes," *arXiv preprint arXiv:2005.04790*, 2020.

[7] D. Gurari, Y. Zhao, M. Zhang, and N. Bhattacharya, "Captioning images taken by people who are blind," in *European Conference on Computer Vision*. Springer, 2020, pp. 417–434.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, "A comprehensive survey of deep learning for image captioning," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–36, 2019.

[10] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2425–2433.

[11] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Visualbert: A simple and performant baseline for vision and language," *arXiv preprint arXiv:1908.03557*, 2019.

[2]https://github.com/JafarBadour/Hateful-Memes-Classification

[12] Y.-C. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Learning universal image-text representations," *arXiv preprint arXiv:1909.11740*, 2019.

[13] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi, "From recognition to cognition: Visual commonsense reasoning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6720–6731.

[14] A. Suhr, S. Zhou, A. Zhang, I. Zhang, H. Bai, and Y. Artzi, "A corpus for reasoning about natural language grounded in photographs," *arXiv preprint arXiv:1811.00491*, 2018.

[15] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, "Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2641–2649.

[16] P. Fortuna and S. Nunes, "A survey on automatic detection of hate speech in text," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–30, 2018.

[17] A. Schmidt and M. Wiegand, "A survey on hate speech detection using natural language processing," in *Proceedings of the Fifth International workshop on natural language processing for social media*, 2017, pp. 1–10.

[18] F. Yu, J. Tang, W. Yin, Y. Sun, H. Tian, H. Wu, and H. Wang, "Ernie-vil: Knowledge enhanced vision-language representations through scene graphs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, 2021, pp. 3208–3216.

[19] K. Kärkkäinen and J. Joo, "Fairface: Face attribute dataset for balanced race, gender, and age," *arXiv preprint arXiv:1908.04913*, 2019.

[20] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai, "Vl-bert: Pre-training of generic visual-linguistic representations," *arXiv preprint arXiv:1908.08530*, 2019.

[21] A. H. Razavi, D. Inkpen, S. Uritsky, and S. Matwin, "Offensive language detection using multi-level classification," in *Canadian Conference on Artificial Intelligence*. Springer, 2010, pp. 16–27.

[22] W. Warner and J. Hirschberg, "Detecting hate speech on the world wide web," in *Proceedings of the second workshop on language in social media*, 2012, pp. 19–26.

[23] R. Kumar, A. K. Ojha, S. Malmasi, and M. Zampieri, "Benchmarking aggression identification in social media," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, 2018, pp. 1–11.

[24] M. H. Christiansen and S. E. Kirby, *Language evolution.* Oxford University Press, 2003.

[25] D. Rorabaugh, "Arabic influence on the spanish language," *Research paper, Seattle Pacific University. https://mast. queensu. ca/˜ rorabaugh/docs/ArabicInfluence. pdf*, 2010.

[26] M. Pant and S. Funo, *Stupa and Swastika: Historical Urban Planning Principles in Nepal's Kathmandu Valley*. NUS Press, 2007.

[27] C. Herbert, "Precarious projects: the performative structure of reclamation," *Language Sciences*, vol. 52, pp. 131–138, 2015.

[28] R. Schmitt-Beck, "Bandwagon effect," *The international encyclopedia of political communication*, pp. 1–5, 2015.

[29] D. Kiela, H. Firooz, A. Mohan, V. Goswami, A. Singh, P. Ringshia, and D. Testuggine, "The hateful memes challenge: Detecting hate speech in multimodal memes," *ArXiv*, vol. abs/2005.04790, 2020.

[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[31] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156–3164.

[32] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[33] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[34] B. Wang and C.-C. J. Kuo, "Sbert-wk: A sentence embedding method by dissecting bert-based word models," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2146–2157, 2020.

[35] R. Z. https://ai.facebook.com/blog/hateful-memes-challenge winners/, "The hateful memes challenge winners: Detecting hateful memes in the facebook hateful memes challenge," *ArXiv*, 2020.