Automated Image Captioning
Using Flickr8k Dataset

CIE 555

Prepared For
Dr. Mohamed ElShenawy

Prepared By

Omar Gaballah	201801697
Ahmed AbdelSalam	201801597
Ibrahim Hamada	201800739
Sohaila Zaki	201800998

Problem definition and motivation

The problem we are trying to solve is how to generate meaningful descriptions for images. It is a very important task in the text generation domain, In which the goal is generating meaningful sentences in the form of human-written text. Image captioning has plenty of applications such that using image captions to create an application to help people who have low or no eyesight.[1]

Literature Review

Many techniques have been proposed for the automatic generation of image captions. The models exploit the CNN network as an encoder for image information That can be used by a RNN model to generate meaningful text[3].

The development of these techniques has seen two approaches. First, Top-down approaches: In this approach, a top model of a CNN model is used for encoding and then LSTM is used for feedforward. Second, the Bottom-Up approach divides the problem into simpler objects and passes the feature vector of each object into an RNN model to produce the caption for that image[3].

1. **Explain Images with Multimodal Recurrent Neural Networks.**

- **Major achievements of the proposed papers:**
  This paper[6] proposed a multimodal Recurrent Neural Networks (m-RNN) model in order to generate captions to describe images. Before this paper, the image captioning task was tackled as a retrieval task such that the image and text features were both extracted and semantically mapped to the same space.Since this was modeled as a retrieval task, we were able to deal only with query images already existing in the datasets.

- **Hyper parameters and Architecture:**
  In this architecture, an m-RNN model has been proposed as a modification for RNN. The m-RNN model consists of an image model to get image features along with a language model language to represent the captions. CNN was used as an image model. For the language model, six layers were used, where one as an input layer, two layers for word embedding, one recurrent layer, one layer that merges image model with language model,and the softmax layer. Thus the m-RNN is much deeper. The layers' densities were annotated on the figure below.

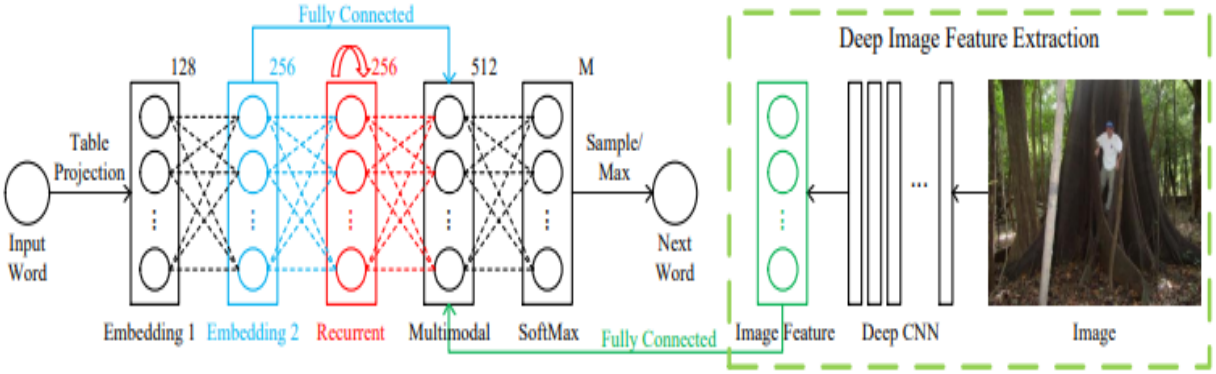

Figure 1: m-RNN layer

- **Accuracy:** They were able to reach a 0.54 BLEU-1 score which was the state of the art by that time.

	Flickr 8K				Flickr 30K			
	$\mathcal{PPL}$	B-1	B-2	B-3	$\mathcal{PPL}$	B-1	B-2	B-3
Ours-RNN-Base	30.39	0.4383	0.1849	0.1339	43.96	0.4699	0.1964	0.1252
Ours-m-RNN	**24.39**	**0.5778**	**0.2751**	**0.2307**	**35.11**	**0.5479**	**0.2392**	**0.1952**

Table 1. Reported scores for the implemented models

2- Deep Visual-Semantic Alignments for Generating Image Descriptions:

- **Major achievements of the proposed papers:** This paper was considered as the image captioning state of the art in 2015. It proposed a model that generated image captions based on a dataset that contains both images and captions to describe the images. It has generated descriptions of visual data that outperformed the retrieval baseline algorithms using a novel Multimodal Recurrent Neural Network architecture.

- **Hyper parameters:** In this paper, SGD was used 100 captioned images following the mini-batch approach and setting the momentum to 0.9. The paper then used cross-validation to investigate the best weight decay value, learning rate, and dropout rate in each layer in the model except the recurrent layers.

- **Accuracy:** Table 2 shows the obtained BLEU scores obtained in the paper on 1,000 test images. The paper got Flickr8K scores 16.7/31.8 and Flickr30K scores are 15.3/24.7

	Flickr8K				Flickr30K				MSCOCO 2014					
Model	B-1	B-2	B-3	B-4	B-1	B-2	B-3	B-4	B-1	B-2	B-3	B-4	METEOR	CIDEr
Nearest Neighbor	—	—	—	—	—	—	—	—	48.0	28.1	16.6	10.0	15.7	38.3
Mao et al. [38]	58	28	23	—	55	24	20	—	—	—	—	—	—	—
Google NIC [54]	63	41	27	—	66.3	42.3	27.7	18.3	66.6	46.1	32.9	24.6	—	—
LRCN [8]	—	—	—	—	58.8	39.1	25.1	16.5	62.8	44.2	30.4	—	—	—
MS Research [12]	—	—	—	—	—	—	—	—	—	—	—	21.1	20.7	—
Chen and Zitnick [5]	—	—	—	14.1	—	—	—	12.6	—	—	—	19.0	20.4	—
Our model	57.9	38.3	24.5	16.0	57.3	36.9	24.0	15.7	62.5	45.0	32.1	23.0	19.5	66.0

Table 2. Reported scores for the used models

3-Show and Tell: A Neural Image Caption Generator: a generative model was proposed which uses deep RNN to generate the correct captions that describe the images.

The following figure shows the used LSTM model:

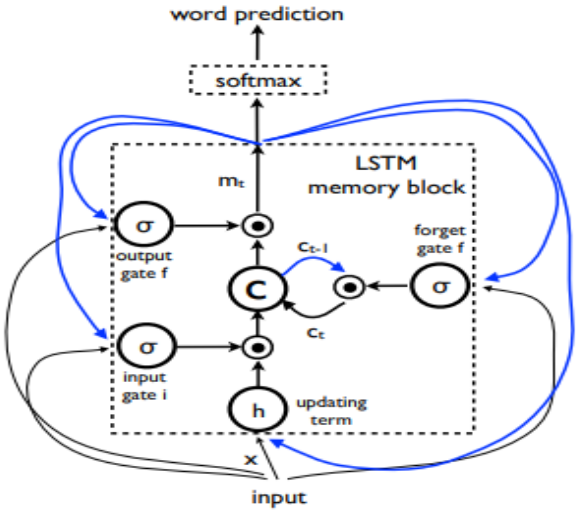

Figure 2. LSTM model architecture

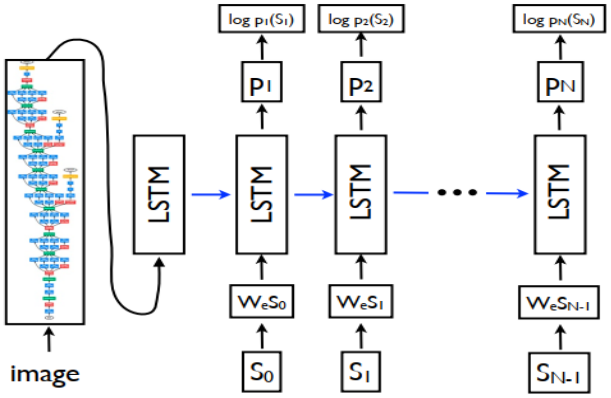

Figure 3. CNN image model with LSTM model

- **Major achievements of the proposed papers:** They used the NIC approach so that they were able to reach BLEU-1 score on the Pascal dataset 59, instead of 25. They were also able to obtain a BLEU-1 score of 66 as an improvement over the last obtained values which was 59 on Flickr30k, they also improved the score from 19 to 28 using SBU dataset. Additionally, they were able to achieve a BLEU-4 of 27.7 on the COCO dataset, which is the current state-of-the-art. [5]

- **Hyper parameters:** it was proposed that the log probability had to be maximized using the following optimization function.

$$\theta^{\star} = \arg\max_{\theta} \sum_{(I,S)} \log p(S|I;\theta) \qquad (1)$$

Such that the model parameters are θ, the input image is I, and the correct caption is S.

$$\log p(S|I) = \sum_{t=0}^{N} \log p(S_t|I, S_0, \ldots, S_{t-1}) \qquad (2)$$

Where the pairs of the example used for training is (S, I), and SGD is used with the samples of training data to help optimizing the summation of the logarithmic probabilities.

RNN was used to model p(S_t|I, S_0, . . . , S_{t-1}) such that the words are expressed using fixed size hidden memory h_t. The memory got updated based on the nonlinear function in (3) after knowing x_t, which is the new input.

$$h_{t+1} = f(h_t, x_t)\ . \qquad (3)$$

The used function in equation 3 depends on its ability to solve the problem of **vanishing and exploding gradients.**

- **Accuracy:** They were able to reach this score in BLEU-1 and SOTA which is the state of the art by that time.

Approach	PASCAL (xfer)	Flickr 30k	Flickr 8k	SBU
Im2Text [24]				11
TreeTalk [18]				19
BabyTalk [16]	25			
Tri5Sem [11]			48	
m-RNN [21]		55	58	
MNLM [14][5]		56	51	
SOTA	25	56	58	19
NIC	**59**	**66**	**63**	**28**
Human	69	68	70	

Table 3. Reported scores for the used models

Based on the implemented models in the three above mentioned papers, it can be concluded that the implemented model in paper (3) achieved the highest accuracy as the model was able to improve BLEU-1 score from 56 to 66 on Flickr30K Dataset and from 19 to 28 on SBU Dataset. It also achieved a BLEU-4 score of 27.7, which is the state of the art on the newly COCO dataset.

Dataset

The Flickr 8k dataset will be used in training and testing the proposed model. The Flickr 8k dataset has been widely used in the field of sentence-based image description. The dataset consists of 8000 images that capture people engaged in everyday events and activities, where each image is annotated using 5 different sentences provided by human annotators. Accordingly, the dataset contains around 40,000 captions for 8,000 images. The pictures were hand-picked to represent a diversity of events and circumstances from six different Flickr groups, and they usually don't feature any famous individuals or places. We will be using 6000 images for training, 1000 images for validation and another 1000 images for testing.

Figure 4. Some Dataset Examples

Figure 5. Some Dataset Examples with Captions

Data Preprocessing

We did the following Data Preprocessing :

1. Lowercase all the words
2. Remove Punctuations
3. Encapsulates the captions start and end sequences
4. Vectorize words with tokenze
5. Prepare a dictionary that has key = imageID and value= List of captions which are vectorized

- We encapsulated the captions so the model can know when the caption ends and when it starts.
- We also created a vocabulary which size was 8917 and after doing the necessary data cleaning it was 8357.
- In figure 6, we can see some statistics about the data set before data cleaning, as for example we can see that a is the most frequently appearing word and we are going to remove it since it does not add much information.

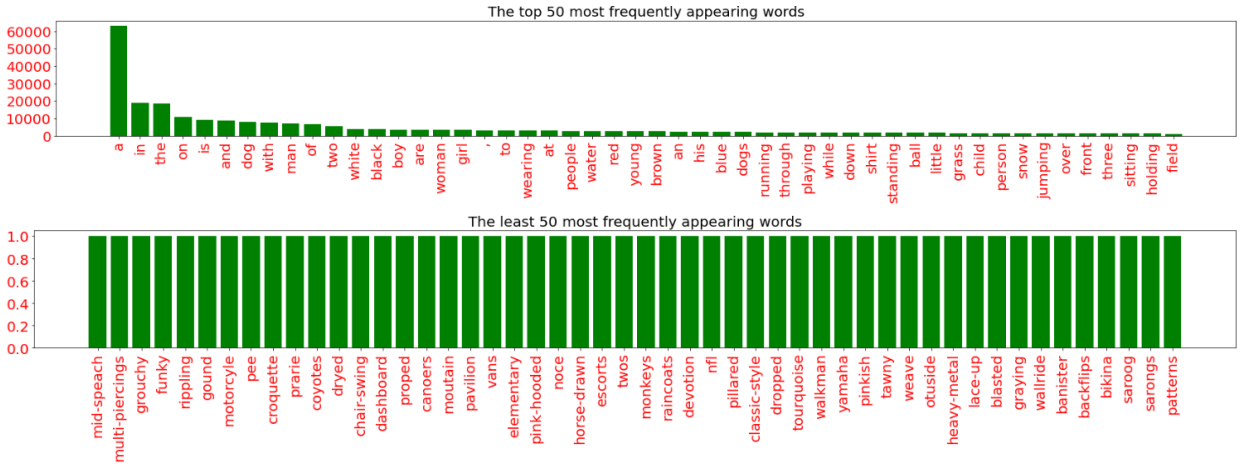

Figure 6. Statistics before cleaning the data

- In figure 7, we can see some images alongside their captions after all the data cleaning that we did.

<start> a little girl in a pink dress going into a wooden cabin <end>
<start> a little girl climbing the stairs to her playhouse <end>
<start> a little girl climbing into a wooden playhouse <end>
<start> a girl going into a wooden building <end>
<start> a child in a pink dress is climbing up a set of stairs in an entry way <end>

<start> two dogs on pavement moving toward each other <end>
<start> two dogs of different breeds looking at each other on the road <end>
<start> a black dog and a white dog with brown spots are staring at each other in the street <end>
<start> a black dog and a tri-colored dog playing with each other on the road <end>
<start> a black dog and a spotted dog are fighting<end>

<start> young girl with pigtails painting outside in the grass <end>
<start> there is a girl with pigtails sitting in front of a rainbow painting <end>
<start> a small girl in the grass plays with fingerpaints in front of a white canvas with a rainbow on it <end>
<start> a little girl is sitting in front of a large painted rainbow <end>
<start> a little girl covered in paint sits in front of a painted rainbow with her hands in a bowl <end>

Figure 7. Images alongside captions

Architecture

Our architecture receives two inputs which are image features and input words. Then it tries to predict the next word. Consequently, our architecture can be divided into three main parts, the first part is the image model, the second one is the language model, and the last one is a concatenation of both.

The image model is composed of 3 layers as shown in figure 8 below. We have used a dense layer as a way to learn what are the important features to be used. Then we followed it with a Repeat vector layer to repeat the image features and pass it to all the steps of the GRUs. Finally, we used a dropout layer as a regularizer with a rate of 0.4. The image model takes image features with a size = of 2048. We have tried to extract the features with two models one with Resnet 50 and the other with Xception.

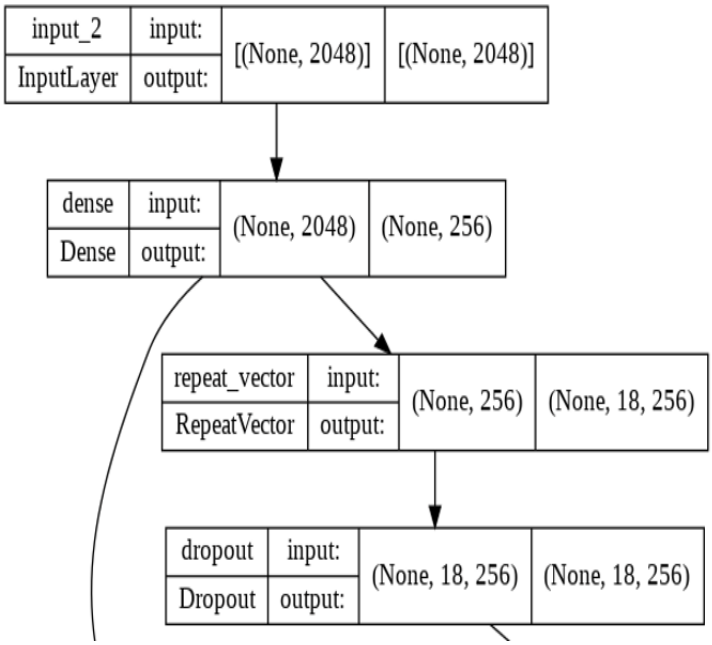

Figure 8. Images Model

The language model is composed of 2 layers as shown in figure 9 below. We have used an embedding layer to learn the text representation of our specified dataset. Then before going to the next layer we used a lambda function to remove masks from the embedding layers as this would cause an issue when we concatenate with the image model. Finally, we followed it with a dropout layer to act as a regular. Instead of passing the words to the language model we tried to initialize the values of the word with three different models and compared the results between them. We first used an embedding layer with dim=50. Then we tried using a glove model with dim= 50 and finally, we used another glove model with dim=300.

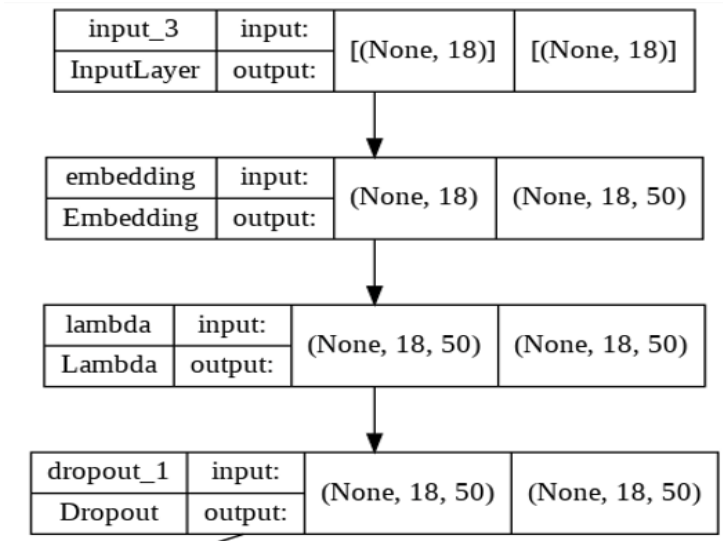

Figure 9. Language Model

The third part of our model is a concatenation between the two models. After concatenation, we had four layers which are a GRU followed by a dropout layer then another GRU layer, and finally a dense layer. Although most of the papers we have seen were using LSTMs[4][5][6], we thought that it would be better to use GRUs as we would have a smaller number of learning parameters. Consequently, we would have fewer calculations and a smaller model size with almost the same approximation results.

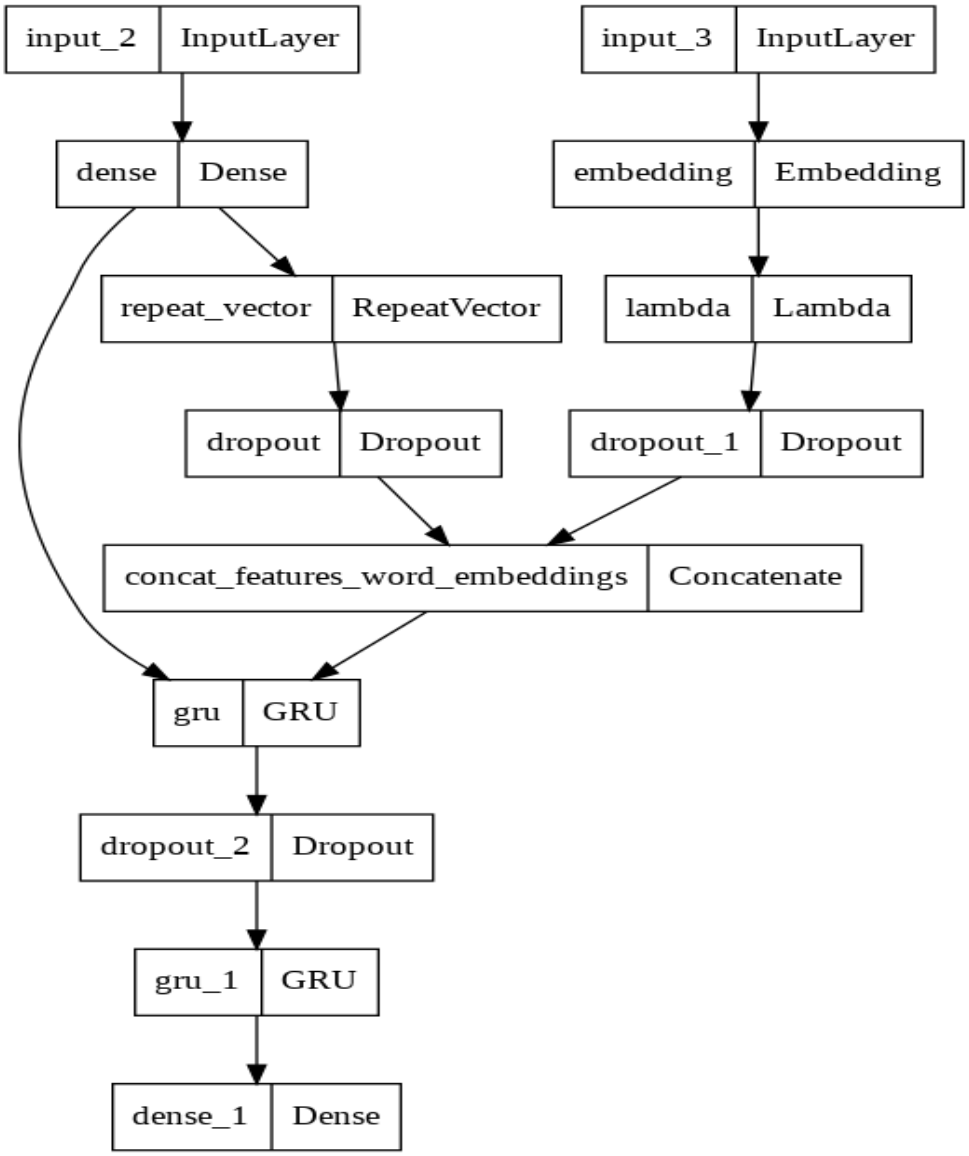

Figure 10. Full Architecture

BLEU Score

BLEU scores were used widely as a metric to test the performance of Image captioning and translation models. The metric gives an output between 0 and 1 to compare the similarities between a reference sentence with a translated one. There are some drawbacks with using BLEU scores in Image captioning models because sometimes the reference sentences will not include all the words that describe the image, accordingly BLEU Score will disregard some correct sentences. In order to overcome this drawback, the generation of sentences has to follow the same settings as in the generation of reference sentences[6].

$$BLEU\ Score\ =\ Brevit\ Penalty\ *\ e^{\frac{1}{N}\sum_{n=1}^{N} P_n}$$

$$Brevity\ Penalty = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c <= r \end{cases}$$

Brevity Penalty(Image by Author)

r: Length of the Reference Sentence.
c: Length of the Machine Translated sentence.
N: Number of n-grams used to evaluate BLEU
P : Modified precision

Brevity term is used to penalize the short sentences, since the modified precision tends to favor the very short sentences. When the translated sentence has a length that equals the length of the reference sentence, the Brevity Penalty will be set to 1, which is the "best match length" and the closest comparative sentence length. For short sentences, an exponential decaying penalty with increasing the length is multiplied by the obtained precision to penalize having very short sentences. Both the shortness penalty and the adjusted n-gram precision length only take the range of target language reference translation lengths into account, not the source length itself. A machine translation will receive a score of 1 if it is exact to one of the reference translations. Due to this, even a human translator may not always receive a score of 1. BLEU-1 score uses the precision score of the unigram model, while BLEU-2 score uses the average precision obtained from both unigram and bigram, and so on for BLEU-3 and BLEU-4.

Implementation

As explained in the above sections, the full architecture contains both image and language models. The image model is used to obtain features vectors that represent each image, while the language model is used to get the representation of words in captions. For the image model, we decided to use both ResNet50 and Xception models. Our choice was based on a comparison held between different deep CNN models [11]. Figure 13 shows that ResNet50 and Xception models showed least validation loss values when testing them on a classification problem.

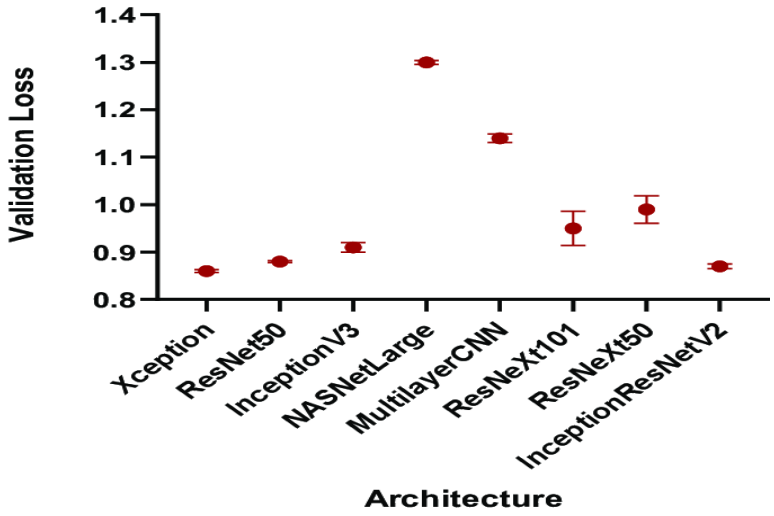

Figure 11. Comparison between different deep CNN models

Additionally, it can be clearly noticed from figure 14 that ResNet50 and Xception models satisfy the optimum trade-off between achieving higher accuracy and having low number of parameters based on the study conducted by (Zhang et al). [12]

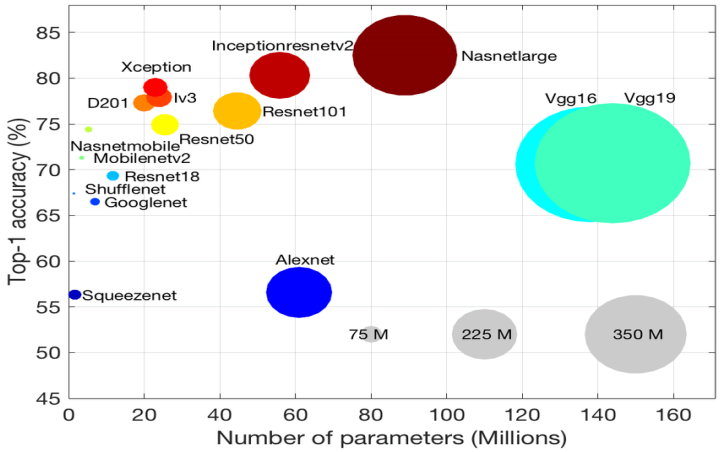

Figure 12. Comparison between ImageNet models (Accuracy vs. Complexity)

For the image model, embedding layers with GloVe (Global Vectors for Word Representation) are used. Global vectors for word representation are referred to as GloVe. By combining the global word-word co-occurrence matrix from a corpus, Stanford researchers created an unsupervised learning technique for creating word embeddings. Interesting linear substructures of the word are displayed in vector space by the resulting embeddings. Unlike Word2vec, which solely uses local language information. That is, only the words around a word have an impact on the semantics that are learned for it. By resolving three significant issues, GloVe accomplishes its mission. There are different versions of GloVe based on the number of tokens, words, and size of representation vector. GloVe 60B is used as a pretrained embedding layer in our implementation which contains 6 Billion tokens and the used representation sizes are 50, and 300.

Accordingly, In order to reach the maximum BLEU Score, both ResNet50 and Xception models are used for image feature extraction, and GloVe 60B with both embedding size of 50 and 300 are used in the language model.

In total, 6 models were built and tested, then a comparison between their performance was held.

1) Model with ResNet50 and No GloVe
2) Model with ResNet50 and GloVe 50d
3) Model with ResNet50 and GloVe 300d
4) Model with Xception and No GloVe
5) Model with Xception and GloVe 50d
6) Model with Xception and GloVe 300d

Each of the 6 models was trained for 100 epochs, then the performance of each model is tested using samples from the test dataset following 5 approaches.

1) Greedy Approach
2) Beam Search with K = 3
3) Beam Search with K = 5
4) Beam Search with K = 7
5) Beam Search with K = 10

Predicting the word that has the highest likelihood in each location is a reasonably simple method. The greedy strategy frequently yields the right result, is quick to compute, simple to grasp, and does so. Over Greedy Search, Beam Search has two advantages. When using Greedy Search, we only used the top term in each slot. Beam Search, on the other hand, broadens this and selects the finest "N" terms. We also took into account each position separately. We didn't look at what came before or after the best term once we had chosen it for that slot. Beam Search, on the other hand, selects the 'N' best sequences so far and takes into account the likelihood of combining all of the words that came before the word in the present location.

Results

1) **Model with ResNet50 and No GloVe:**

In this model, the image model obtains a feature vector with a size of 2048 for each image in the dataset using ResNet50 Model. The language model does not use GloVe in its embedding layer, but it is initialized with zeros instead. After training the model for 100 epochs, the model obtained BLEU Scores as follows:

BLEU-1	**0.535028**
BLEU-2	**0.297881**
BLEU-3	**0.188988**
BLEU-4	**0.089115**

Table 4: Results of ResNet50 with No Glove

The model is tested using some samples from the test dataset and the obtained captions using each of the 5 approaches are shown in the following figures.

Figure 13: 4 Sample test results using ResNet50 and No GloVe

❖ Comment on the Results:

- Since the above model does not use a good initialization for word representation vectors in its embedding layer, we can see that the model stuck quickly and produces “a” repeatedly.

- This is because now we have to train the language model more and more to have some kind of understanding of the text representation. So, using only 100 epochs with this model was not sufficient at all.

- The above figures and the results that will be shown later that were obtained using the other models that depended on pretrained GloVe representation vectors in the embedding layers shows how it is important and effective to use pretraining in such a problem.

2) Model with ResNet50 and GloVe 50d:

In this model, the image model obtains a feature vector with a size of 2048 for each image in the dataset using ResNet50 Model. The language model uses GloVe in its embedding layer with a size of 50 dimensions.
After training the model for 100 epochs, the model obtained BLEU Scores as follows:

BLEU-1	**0.553598**
BLEU-2	**0.343159**
BLEU-3	**0.238214**
BLEU-4	**0.127202**

Table 5: Results of ResNet50 with Glove 50d

The model is tested using some samples from the test dataset and the obtained captions using each of the 5 approaches are shown in the following figures.

Figure 14: 4 Sample test results using ResNet50 and GloVe 50d

❖ Comment on the Results:

- From the image on the top left, we can see that the model in the greedy approach has considered the boy with a broken arm. We did not understand fully why the model considered the boy with a broken arm. However, our thoughts were that the model was either confused by the light color of the boy's t-shirt or maybe because of the weird shape of his hand as he tries to catch his leg.

- For the image in the top right, we can see that there was a mistake as the model considered a mountain to exist. Our interpretation is that it classified the shape of the far and blurry buildings as mountains. We can also see in the beam search that the people were considered “climbers” instead of “group of standing people”. This is because we are assigning probabilities to words. Consequently, if there were mountains then the probability of having climbers would be high..

- For the image on the left bottom, we can see that the model is kind of colorblind as it is considered the boy in a blue t-shirt instead of white. In the beam search, we can see the results were very far from the picture.

- For the image in the bottom right, the model has classified the dog as running despite it being sitting. Hence, the model is disabled for classifying actions correctly. In the beam search, we can see it has a little far interpretation as it considered the dog as white, black, and with soccer. This may be a consequence of having multiple images in the dataset where the dogs play soccer or have white or black colors.

3) Model with ResNet50 and GloVe 300d:

In this model, the image model obtains a feature vector with a size of 2048 for each image in the dataset using ResNet50 Model. The language model uses GloVe in its embedding layer with a size of 300 dimensions. The aim of using this model is to assess the effectiveness of increasing the dimension of embedding layer vectors. After training the model for 100 epochs, the model obtained BLEU Scores as follows:

BLEU-1	**0.517942**
BLEU-2	**0.316387**
BLEU-3	**0.212321**
BLEU-4	**0.103534**

Table 6: Results of ResNet50 with Glove 300d

The model is tested using some samples from the test dataset and the obtained captions using each of the 5 approaches are shown in the following figures.

Figure 15: 4 Sample test results using ResNet50 and GloVe 300d

❖ Comment on the Results:

- From the image on the top left, we can see that the model results in the greedy approach have done much better than the Resnet50 with glove 50. However, it considered the girl as a boy. Our thoughts are that the datasets may have many boys' images compared to girls.

- For the image in the top right, we can see that the model suffers from spatiality ordering as it considers the people riding bikes on the boat. The model has combined the bike next to someone and it thought that was the activity the group was doing. In addition, it combined this with the boat in the image. In beam search 3,5,7 the model tended again to consider the existence of a mountain or a hill and also suffered from losing the image spatiality. However, in beam search with k=10, the model gave a reasonable answer by considering the people were on the beach.

- For the image on the left bottom, we want to highlight the fact that the model considered the leaves as grass in both the greedy and beam search with k=3. This is because grass and leaves may have similar feature representation, but with more grass images in the dataset than leaves. In addition, the model still suffers from color blindness as it considered the boy's shirt as red.

- For the image in the bottom right, We believe the comments under the same image in the model that used ResNet50 with glove 50 hold here.

4) Model with Xception and No GloVe:

In this model, the image model obtains a feature vector with a size of 2048 for each image in the dataset using Xception Model. The language model does not use GloVe in its embedding layer, but it is initialized with zeros instead. After training the model for 100 epochs, the model obtained BLEU Scores as follows:

BLEU-1	**0.543330**
BLEU-2	**0.335017**
BLEU-3	**0.239280**
BLEU-4	**0.125075**

Table 7: Results of Xception with No Glove

The model is tested using some samples from the test dataset and the obtained captions using each of the following 5 approaches are shown in the following figures.

Figure 16: 4 Sample test results using Xception and No GloVe

❖ Comment on the Results:

- Although the results in the images above were poor as a result of not using a good word representation as initialization, we can see that the usage of Xception features has given us way better results than the Resnet50. As we can see the model does not only get stuck at “a” but it gets stuck in different words as “in” and may not get stuck in some cases as in the image on the bottom left.

5) Model with Xception and GloVe 50d:

In this model, the image model obtains a feature vector with a size of 2048 for each image in the dataset using Xception Model. The language model uses GloVe in its embedding layer with a size of 50 dimensions. After training the model for 100 epochs, the model obtained BLEU Scores as follows:

BLEU-1	**0.605276**
BLEU-2	**0.406098**
BLEU-3	**0.305233**
BLEU-4	**0.176107**

Table 8: Results of Xception with Glove 50d

The model is tested using some samples from the test dataset and the obtained captions using each of the following 5 approaches are shown in the following figures.

Figure 17: 4 Sample test results using Xception and GloVe 50d

❖ <u>Comment on the Results:</u>

- We can see that the model results were overall very accurate and the results have very high details that were not captured by resnet50. For instance, the model could see that there were two people in a cannon. Moreover, it could have detected that it was aligned by trees. This kind of interpretation says that the image features have some kind of sense regarding the image's spatiality.

- For the image in the bottom right, The model has mistakenly considered there were two surferers instead of one. If we have considered the figure above, we will see that "two" was the eleventh most frequent word. Consequently, this is maybe why the model decided there were two surferers, not one.

6) Model with Xception and GloVe 300d:

In this model, the image model obtains a feature vector with a size of 2048 for each image in the dataset using Xception Model. The language model uses GloVe in its embedding layer with a size of 300 dimensions. After training the model for 100 epochs, the model obtained BLEU Scores as follows:

BLEU-1	**0.617114**
BLEU-2	**0.418422**
BLEU-3	**0.317608**
BLEU-4	**0.187548**

Table 9: Results of Xception with Glove 300d

The model is tested using some samples from the test dataset and the obtained captions using each of the following 5 approaches are shown in the following figures.

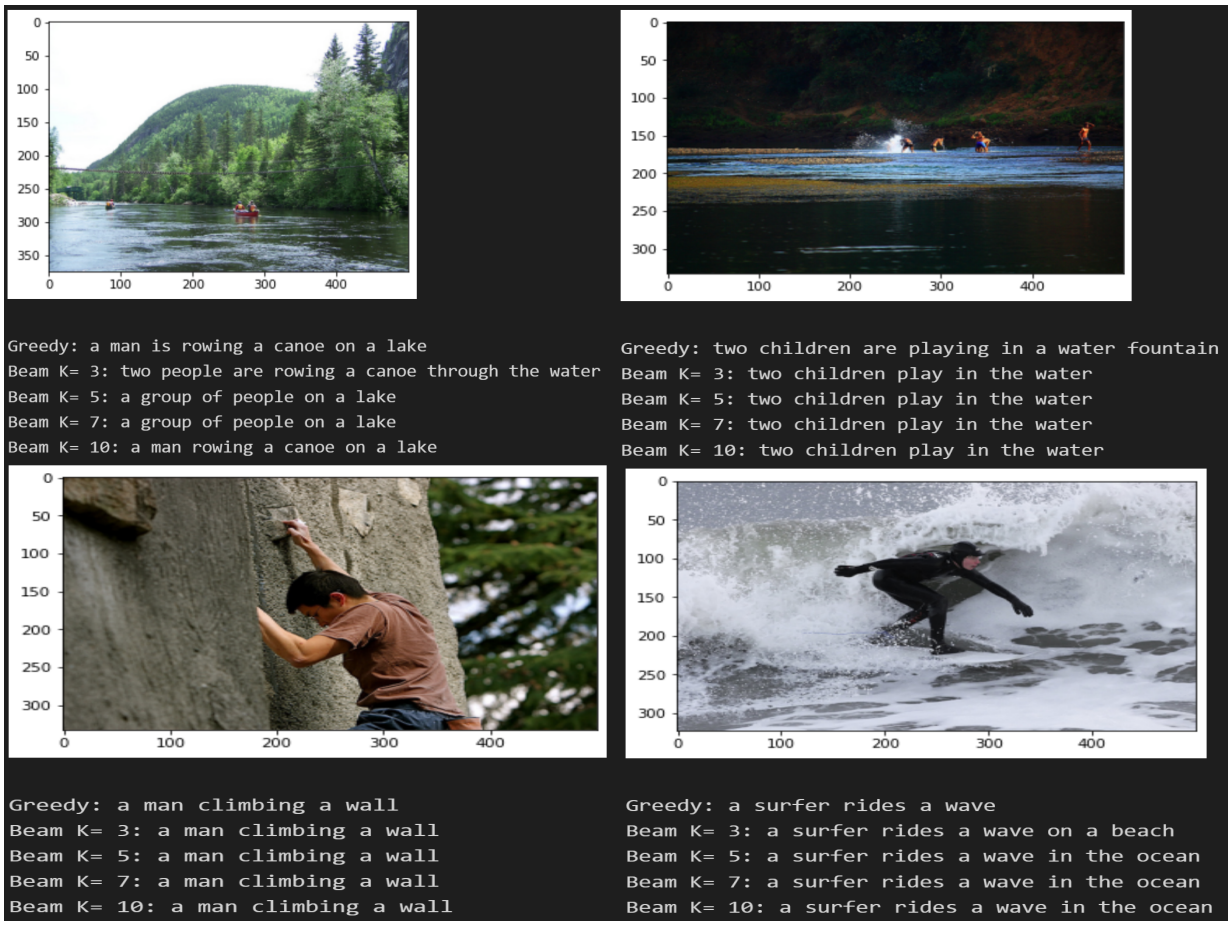

Figure 18: 4 Sample test results using Xception and GloVe 300d

❖ Comment on the Results:

The above result is the best result we have got so far. We can see that it has a very accurate representation of all the images above. Moreover, it did not mistakenly consider there were two surfers as in Xception with Glove=50.

- **Testing the 6 Models on Images Not in the Dataset**

The images below were retrieved randomly from the internet. We can see in general that any model that did not use Glove gave very poor results. Models that used Resnet were trying to mention objects in the scene rather than trying to describe them. While models with Xception were trying to describe the images generally not literally. For instance, in test image 3, we will see Xception with Glove=50 says "a group of children is playing soccer in the park" while in Resnet 50 ith Glove= 50 and beam search of k=7, says "a group of men in matching outfits are standing in mountain range".

1) **Test Image 1:**

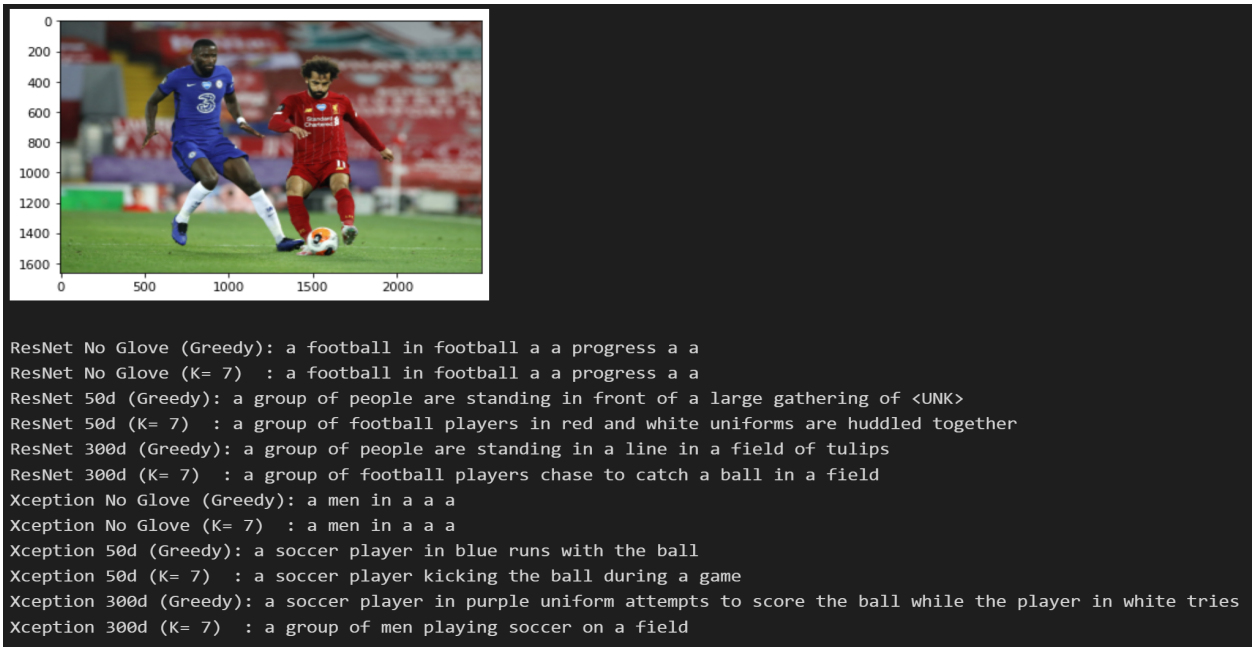

Figure 19: Performance of 6 Models on Image that is not in the dataset

2) Test Image 2:


```
ResNet No Glove (Greedy): a man of children in a a
ResNet No Glove (K= 7)  : a man of children in a a
ResNet 50d (Greedy): a group of people are standing in front of a large gathering of <UNK> <UNK>
ResNet 50d (K= 7)  : a group of people standing in front of a graffitied wall
ResNet 300d (Greedy): a group of people are standing in front of a building with a bunch of people in
ResNet 300d (K= 7)  : a crowd of people are standing in front of a colorful building
Xception No Glove (Greedy): a young and is a a a a a a
Xception No Glove (K= 7)  : a young and is a a a a a a
Xception 50d (Greedy): a girl and girl are playing on a field
Xception 50d (K= 7)  : a group of children are running on the grass
Xception 300d (Greedy): a girl is running with a child in a park
Xception 300d (K= 7)  : a little girl is chasing another child on the grass
```

Figure 20: Performance of 6 Models on Image that is not in the dataset

3) Test Image 2:


```
ResNet No Glove (Greedy): a group of children children children children cream cream cream cream cream
ResNet No Glove (K= 7)  : a group of children children children children cream cream cream cream cream
ResNet 50d (Greedy): a group of men in camouflage pants and <UNK> are standing in a line of a busy
ResNet 50d (K= 7)  : a group of men in matching outfits are standing in front of a mountain range
ResNet 300d (Greedy): a group of people are standing in front of a carnival
ResNet 300d (K= 7)  : a group of greyhound dogs racing around a track
Xception No Glove (Greedy): a boys in soccer soccer soccer soccer soccer soccer soccer soccer
Xception No Glove (K= 7)  : a boys in soccer soccer soccer soccer soccer soccer soccer soccer
Xception 50d (Greedy): a group of children are playing soccer in a park
Xception 50d (K= 7)  : a group of children are playing with a soccer ball
Xception 300d (Greedy): a little boy in a blue shirt chasing a soccer ball
Xception 300d (K= 7)  : children playing soccer in a field
```

Figure 21: Performance of 6 Models on Image that is not in the dataset

Conclusion

In this project, we have investigated the conventional methods that have used a global CNN as a feature representation for image and conventional word embeddings. Although these models can give acceptable results, they have shown poor performance in classifying the actions in the images, relating the spatial information of the image with the generated text, and classifying colors. We present below the results obtained from each of the 6 models using our 5 approaches and the results of testing the models on randomly selected images from the internet.

	ResNet50, No GloVe	ResNet50, GloVe 50	ResNet50, GloVe 300	Xception, No GloVe	Xception, GloVe 50	Xception, GloVe 300
BLEU-1	0.535028	0.553598	0.517942	0.543330	0.605276	0.617114
BLEU-2	0.297881	0.343159	0.316387	0.335017	0.406098	0.418422
BLEU-3	0.188988	0.238214	0.212321	0.239280	0.305233	0.317608
BLEU-4	0.089115	0.127202	0.103534	0.125075	0.176107	0.187548

Table 10: Comparison between BLEU Scores of the 6 models.

Comparison to the state of the art

Image captioning has developed much in recent years. A paper in 2018 proposed that the traditional ways of extracting image features with global CNN lose a lot of spatial information[13]. As a result, they proposed semantic representations of images and updated the language model by using semantic element embedding. Their semantic embedding was employed to investigate local features in the image and generate captions for it. This methodology has overcome the state of the art by then scoring B1 of 0.75 on the COCO dataset.

The state of the art nowadays is based on transformer architecture. An M2 architecture was proposed by this paper[14], improving both the image encoding and the language model. The key idea was to learn a multi-level representation of the relationship between image region and prior knowledge. After that, mesh-like connectivity was used at the decoding stage to investigate low and high-level features.This architecture has managed to score BLUE-1 equals to 81.6 and BLUE-4 equals to 39.7 surpassing the state of the art by then.

Ethical Considerations

The model could learn toxic or rude or offensive language which could lead to generating offensive captions to given images. This could be because of having bad words like curse words or violent language in the training data which would be something we need to consider. There are methods that can be categorized as data-based or decoding-based strategies that aim to reduce toxicity. Data-based strategies are considered computationally expensive due to adding pre-training to the model and changing the parameters. Decoding-based methods have the advantage of being more accessible to practitioners and less expensive due to leaving the parameters unchanged and only modifying the decoding algorithm.[2]

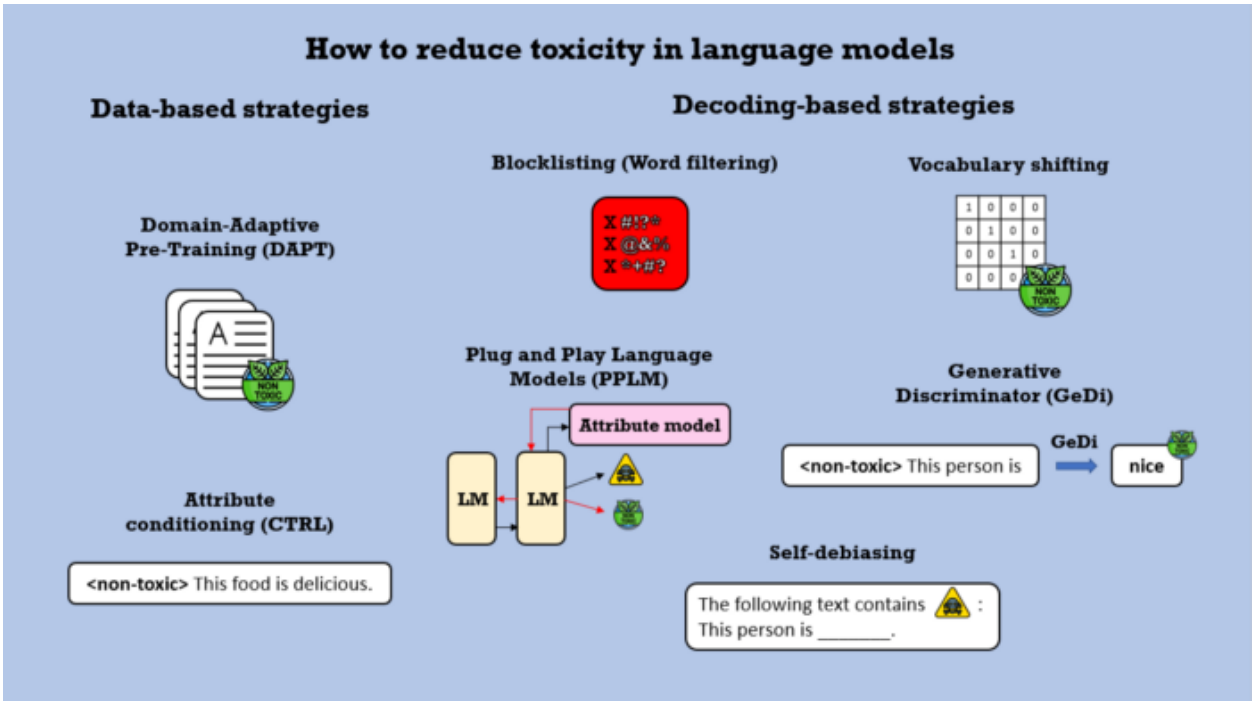

Figure 22. Toxicity in language models

Work Distribution

Name	Work Distribution
Ibrahim Hamada	- Contributed greatly Improving the model performance: by adding Glove and Xception - Apply model with no glove and Xception - Apply glove with dim= 50 and Resnet 50 - Apply glove with dim= 50 and Xception as feature extractor - Adding BLEU-Score
Sohaila Zaki	- Apply glove with dim= 300 and Resnet 50 - Adding BLEU-Score - Choosing the dataset - Investigating the dataset - Greedy approach to generate sentence
Omar Gaballah	- Apply glove with dim= 50 and Resnet 50 - Choosing the dataset - Investigating the dataset - Wrangling and cleaning the dataset
Ahmed Abdelsalam	- Defining the base architecture and input pipeline - Contributed in interpreting the results - Apply model with no glove and Resnet50 - Apply glove with dim= 50 and Resnet 50 - Apply glove with dim= 300 and Xception as feature extractor

References

[1] A. Roy, “A guide to image captioning,” *Medium*, 09-Dec-2020. [Online]. Available: https://towardsdatascience.com/a-guide-to-image-captioning-e9fd5517f350. [Accessed: 21-May-2022].

[2] J. Nikulski, “Toxicity in AI text generation,” *Medium*, 13-Sep-2021. [Online]. Available: https://towardsdatascience.com/toxicity-in-ai-text-generation-9e9d9646e68f. [Accessed: 21-May-2022].

[3] P. Waghmare and S. Shinde, “International Conference on Communication and Information Processing,” in *Artificial Intelligence Based On Image Caption Generation*, 2020.

[4] A. Karpathy and L. Fei-Fei, “Deep visual-semantic alignments for generating image descriptions,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[5] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, “Show and tell: A neural image caption generator,” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[6] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille, “Explain images with multimodal recurrent neural networks,” arXiv.org, 04-Oct-2014. [Online]. Available: https://arxiv.org/abs/1410.1090v1. [Accessed: 22-May-2022].

[7] T. Ganegedara, “Light on math ML: Intuitive Guide to Understanding Glove embeddings,” *Medium*, 15-Nov-2021. [Online]. Available: https://towardsdatascience.com/light-on-math-ml-intuitive-guide-to-understanding-glove-embeddings-b13b4f19c010. [Accessed: 24-Jun-2022].

[8] J. S. Chawla, “Word vectorization using glove,” *Medium*, 06-Jul-2020. [Online]. Available: https://medium.com/analytics-vidhya/word-vectorization-using-glove-76919685ee0b. [Accessed: 24-Jun-2022].

[9] R. Khandelwal, “Bleu - bilingual evaluation understudy,” *Medium*, 26-Jan-2020. [Online]. Available: https://towardsdatascience.com/bleu-bilingual-evaluation-understudy-2b4eab9bcfd1. [Accessed: 24-Jun-2022].

[10] K. Doshi, “Foundations of NLP explained visually: Beam Search, how it works,” *Medium*, 21-May-2021. [Online]. Available: https://towardsdatascience.com/foundations-of-nlp-explained-visually-beam-search-how-it-works-1586b9849a24. [Accessed: 24-Jun-2022].

[11] Thirumalaraju, Prudhvi & Kanakasabapathy, Manoj Kumar & Bormann, Charles & Gupta, Raghav & Pooniwala, Rohan & Kandula, Hemanth & Souter, Irene & Dimitriadis, I. & Shafiee, Hadi. (2021). Evaluation of deep convolutional neural networks in classifying human embryo images based on their morphological quality. Heliyon. 7. e06298. 10.1016/j.heliyon.2021.e06298.

[12] Zhang, Youshan & Davison, Brian. (2020). Impact of ImageNet Model Selection on Domain Adaptation.

[13]X. Zhang, S. He, X. Song, R. Lau, J. Jiao and Q. Ye, "Image captioning via semantic element embedding", *Neurocomputing*, vol. 395, pp. 212-221, 2020. Available: 10.1016/j.neucom.2018.02.112 [Accessed 24 June 2022].

[14] M. Cornia, M. Stefanini, L. Baraldi, R. Cucchiara;”Meshed-Memory Transformer for Image Captioning”, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10578-10587.