Towards Empathy in Visiolinguistic Tasks

Young Se Kim youngsekim@umass.edu
Melnita Dabre mdabre@umass.edu
Chinmay Shirore cshirore@umass.edu
Seung Suk Lee seungsuklee@umass.edu

1 Problem statement

Recent developments in the field of Computer Vision and Natural Language Processing have allowed the two fields to be blended together. As a result, there is a growing number of research subfields that are emerging from the collaboration of the two fields. Image captioning and Visual Question Answering (VQA) are such sub-fields that require the integration of the visual analysis of the image into a text-generating model. In other words, the generated caption and answer must describe what the image shows, while also being a grammatical and fluent sentence.

One way that such image captioning and VQA model can be made more human is to allow it to analyze the sentiment or emotion from the image because as humans, we get the sentiment from the image, as well as factual information.

In this project, we try to develop an image captioning model and Visual Question Answering dataset that makes use of the sentiment information that the image expresses, in generating the caption for the image, so that the generated text could be more affective and exhaustive.

The major contributions that we propose in this project are 1) a new affective visual question answering dataset (SentiQA) 2) a new model evaluation strategy on affective visual image captioning 3) attempts of new approaches including GPT2, paraphrasing and controlled text generations and LSTM + LXMert in affective visual image captioning. You can find the detailed code and our contribution at our Github repo https://github.com/yskimCal/visualqa.

2 What we proposed vs. what we accomplished

In this section, we provide the list of things we proposed to do in our project proposal. The tasks we completed are crossed out and as for the tasks we failed to accomplish, the reason why we failed are briefly explained.

- ~~Using LSTM as the baseline algorithm~~
- ~~Using OmniNet to generate text for image captioning/question answering~~
- ~~Perform an error analysis to diagnose on what examples our model performs poorly and relatively well~~
- ~~Using the hidden states from LXMert for the task of Image Captioning - we did not propose this in the original proposal but we found this approach during our project~~
- Using BERT to generate text for image captioning/question answering - we concluded to not to use video data, so BERT was no longer needed
- ~~Using Faster R-CNN to detect sentiments from the images, which can be used for the OmniNet model - Faster R-CNN has now been deprecated but we use it's variant BUTD(Anderson et al., 2018) instead which has become very popular now.~~
- Using the descriptive captions generated by the SentiAttend model to improve the question answering - The code or the dataset for the SentiAttend model is not available and hence we chose to use other captioning models for our experiments.
- Testing our best model on the WikiArt Emotions dataset (Mohammad and Kiritchenko, 2018) to generate better image captions or question answers for artworks - we concluded that SentiCap dataset (Mathews et al.,

2015) is more suitable for our project to generate common sentimental image captioning and question and answering

- Using GPT2 to generate sentimental image captions - due to lack of enough dataset to train, could not figure out proper input to generate successful sentence

3 Related work and Our contribution

3.1 Visual Sentiment Analysis

The works in Visual Sentiment Analysis allow the machine to detect sentiment attributes using attention mechanisms (You et al., 2017). The attention mechanism allows the model to attend to specific regions of the image when analyzing the sentiment expressed in the image (You et al., 2017).

The visual sentiment analysis is applied in our evaluation of the model outputs and the error analysis, to investigate how much portion of our generated sentences contain negative or positive sentiment, and to compare whether each model is biased to generate captions with a particular sentiment over another.

3.2 Affective Image Captioning

There are several research that attempt to include the sentiment information in the generated caption about the image. For example, Senti-Attend (Nezami et al., 2018) uses the attention mechanism to deal with the fact that humans focus on different elements of the image depending on the sentiment they feel from the image. For example, given the same image, humans wrote a caption about the darker side of the image, when asked to generate a caption with a negative sentiment; and on the other hand, humans generated captions about the brighter side of the image, when asked to generate a caption with a positive sentiment (Nezami et al., 2018).

SentiAttend takes on the image and the desired sentiment category (positive, negative or neutral) as input and generate the image captions for the image. The model is trained on the MS COCO dataset (Lin et al., 2014) and then the SentiCap dataset, which has a positive, a negative and a neutral captions for each image (Mathews et al., 2015). The model performance is evaluated based on the standard metrics such as BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014), CIDEr (Vedantam et al., 2015), and ROUGE-L (Lin, 2004), as well as SPICE (Anderson et al., 2016).

We started from this approach and tried on several approaches to generate more accurate, rich and human readable sentimental sentences. The approaches include baseline of LSTM and OmniNet and GPT2, paraphrasing and controlled text generation and LSTM + LXMERT. The detailed approach will be introduced in Section 6.

3.3 Affective Visual Question Answering using Triple Attention Network

There is much research on Visual Question Answering and Visual Sentiment Analysis respectively, but there is little research that combines the two. Ruwa et al. (2019, 2018) are some of the few research that attempts to do that. There are three major contributions in their series of work.

First, they propose a Triple Attention Network model, in which the features for the question, the image, and the sentimental attributes of the image are embedded separately and then attended in a single attention mechanism (Ruwa et al., 2019), as an improvement from the Dual Attention model proposed in Xu and Saenko (2016).

Their second contribution is that they use the CNN visual multi-attribute detector to extract the multiple sentimental attributes from the image, which improves You et al. (2017). They find that extracting two features is optimal in improving the model accuracy (Ruwa et al., 2019).

The last notable contribution in their work is that they customize two different datasets, one for the VQA and the other for the VSA so that they can be used together for the purpose of training the model to do both at the same time (Ruwa et al., 2019).

3.4 Summary of our contribution

Our contribution includes:

1. We build a new dataset called SentiQA for tackling the affective VQA problem.

2. We design a new model evaluation strategy for cases where a dataset is not available for testing.

3. We also approach and analyse these problems from new perspectives of paraphrasing, attribute injection and ensemble methods.

4 Dataset

4.1 SentiCap with modifications - Generic SentiQA

Datafile - Generic SentiQA

We began by using the SentiCap dataset since it is one of few datasets that provides affective captions. We use SentiCap as our base dataset for our models we trained or fine-tuned. This dataset consists of images from the MS COCO 2014 validation dataset (over 2K test images) and provides sets of affective captions with positive and negative sentiments against to each image. The dataset also highlights the adjective noun pairs in each caption and provided the list. In addition to directly training and testing on our model using SentiCap dataset for visual question and answering, we created an affective visual questions answering dataset (since there is no such dataset currently available) using simple scripts to create questions such as ”What do you see in the picture?”, ”What is the girl doing?” or ”What is in the image?” based on the captions. Since SentiCap has 2,360 images with average 5-6 captions for each image, we were able to generate about 12K training QA pairs, 12K test pairs and about 1.2K validation pairs. However, one of the shortcomings of this dataset is the lack of variety in the questions. We call this dataset as SentiQA and have used it to train most of affective Visual question and answering models, which unfortunately did not return any promising results.

4.2 Visual 7W - Attempted but no success

We also attempted to create a dataset out of the popular visual 7W dataset that is used for visual question answering. Since this is more generic VQA dataset in public, we extracted questions and captions which is focused more on human actions or emotions. We extracted these pairs by using keywords containing emotions or nouns like ”girl”, ”boy”, ”child”, etc. Out of the total 27K images for telling QA, we found 2K that matched our requirements. As a next step, we put these images through multiple Mood detector models including DeepFace (Serengil and Ozpinar, 2020) (wrapper over Facebook’s deepface, Googles facenet and a few other emotion detection models) and a CNN based model (Priya and Dwivedi, 2018). However, these models were not able to identify faces and retrieve moods successfully and we got less than 600 images with a very low accuracy (this was checked manually for a few samples). We intended to update the answers with these mood + noun pairs to create sentiment QA pairs. We believe this approach may have been more effective if we have a better mood detector or with humans intervention for generation of mood labels for images.

4.3 Flickr 8K dataset

We used Flickr 8K dataset (Jain, 2020) to train our baseline models. Flickr 8K dataset consists of 8,000 images. Each image paired with 5 different captions which provide descriptions of the salient entities and events. The images are chosen from 6 different Flickr groups and annotated manually by annotators. We divided 6,000 data for training and 2,000 data for testing. We also encoded all these images using InceptionV3 model to be used as input for the visual head in the baseline model.

5 Baselines

5.1 LSTM

LSTM (Hochreiter and Schmidhuber, 1997) is a classic but dependable approach. We use this as our baseline. We initially planned to use the LSTM model from AVQAN (Ruwa et al., 2018). However, neither their code, nor their dataset has been released, we customly build it from scratch. Given the lack of reliable affective VQA datasets, we also worked on the generation of affective caption by ourselves.

5.1.1 LSTM for generating affective captions

We began by using a simple LSTM model by referencing (Vadlamudi, 2019) which had been pretrained on RESNET50 (He et al., 2016). First, we used the features extracted by RESNET50 and trained it on Flickr 8K dataset from scratch to set it as our baseline model since there is no available dataset for evaluation in affective image captioning in public. We call it as Vanilla-LSTM. Vanilla-LSTM is trained on total 6,000 training data of Flickr 8K dataset with learning rate of 0.0001 and batch size of 3 for 30 epochs. Even though it was our baseline model, as we will see in evaluation section, it returned pretty good result at our evaluation metrics. After that, we trained it on SentiCap dataset from scratch which we call it as SentiCap-LSTM to compare the performance between those two models whether model trained on SentiCap dataset can generate more accurate, rich and human readable sentences. As we can see it at evaluation and error analysis section, the model trained on SentiCap dataset generated more accurate and rich sentences than model trained on regular caption dataset. The SentiCap-LSTM is trained on 1,217 training examples in SentiCap dataset which consists of average 3 negative and 3 positive descriptions of each image with the same parameter setting as Vanilla-LSTM.

5.2 OmniNet

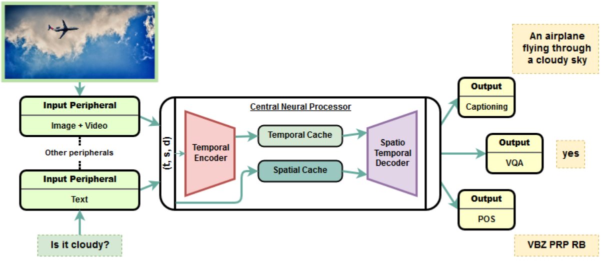

Figure 1: The structure of the OmniNet Model (Pramanik et al., 2019a).

Our next approach was to use the OmniNet model by (Pramanik et al., 2019a). The OmniNet accepts most types of data such as image, video, text and speech in peripheral networks which generate output representation with a spatio-temporal tensor (the author considers image or video as spatial data and text as temporal data). Peripheral networks use an existing pre-trained model like ResNet and BERT. After that, it combines and generates feature representation at the central neural processor (CNP) which is an attention based encoder-decoder. The CNP is originally designed for multiple language modeling tasks with sequence transduction similar to the Transformer (Vaswani et al., 2017) architecture. The CNP implements a generic encoding function to process and generate the spatio-temporal representations of the input. The encoder is called multiple times whenever there is each multi-modal input from the respective peripheral. The decoder consists of multi-head scaled dot product attention with multiple heads for the temporal cache and gated multi-head attention over the spatial cache, and decode predictions as softmax probabilities. Again here, we trained OmniNet twice for the 2 tasks by referencing their Github repo (Pramanik et al., 2019b):

1. Captioning: We trained OmniNet on SentiCap data for 15 epochs with a batch size of 64 and a learning rate of 0.001. The results look quite promising and we believe training it for longer and a larger data could help generate very good captions which could be used in few of our VQA approaches.

2. VQA: In addition to training it on SentiCap for Captioning we trained OmniNet on SentiQA data as well. Similar to the previous model we trained it for 15 epochs with a batch size of 64 and learning rate 0.001 as recommended by the author given their good results on the general VQA tasks. This model took over 12 hours to train was stopped before the entire training process could complete. However, the model like most LSTM based VQA models gives a single token output and hence similar to our attempt of trying to use an LSTM for this approach, we were unable to extract full sentence answers.

6 Our approach

6.1 GPT2 approach - Attempted but no success

First approach we tried to generate combined sentence with sentiments and captions, we applied GPT2 (Radford et al., 2019), which is consist of decoder only transformer and pre-trained on a massive dataset of web text in self-supervision fashion. GPT2 use auto-regressive language model which predict next token by given sequence of inputs and which proved the promising results on many text generation tasks. The pre-trained model was available at Huggingface library (Face, 2020a) but most of applications available used the input of one word of ”prompt” and generate sentence, so we modified the input which can insert sentiment and caption together and generate combined sentence. First, we separated adjective and caption in SentiCap (Mathews et al., 2015) dataset and fine-tuned on GPT2 model with input such as ”<bos> Adjective <caption> blah blah Noun blah blah <result> blah blah Adjective Noun blah blah” and then make GPT2 to predict result when we put input of prompt. However, our generated sentence was not successful and not human readable, and ended up being a sequence of repeated words. We concluded that the model is not trained well because of the input we created or because of the lack of training dataset and we ended up deciding to use alternative solution.

6.2 Paraphrasing/Controlled Text generation - Attempted but no success

As we struggled to find any datasets for our specific task of Affective Question Answering, we were looking for ways to achieve this task without explicitly having to train on a task specific dataset. One of the methods we were interested in was paraphrasing as we only need one dataset such as SentiCap to train it on, And it would be able to adapt to other tasks. Conditional text generation(CTG) in particular deals with the task of manipulating sentences based on some control parameters such as sentiment or tense as shown in Hu et al. (2017). We used this paper as our reference. The open-source implementation is available under the Texar library (Hu et al., 2019) and it claims to produce very promising results. We fine-tuned this model on SentiCap but the sentences produced were not only completely out of context but also not coherent or readable. We also looked for other paraphrase models such as Krishna et al. (2020) which is a model for style transfer, for example changing the style of a tweet into a Shakespearean sentence. However, after consulting with the instructors, we decided to pursue other possible ideas.

6.3 LSTM + LXMert Approach - Attempted but with poor results

Recent models such as LXMert, VilBert and Uniter have been leading almost all the leaderboards for the computer vision with natural language tasks such as visual question and answering. These models are pretrained on masked language modelling on VQA dtasets such as VQA (Agrawal et al., 2017) and GQA (Hudson and Manning, 2019). Despite their success in all these tasks there is no available method or implementation utilizing these models for the task of Image Captioning, we trained a generator network that could utilize this latent representation from LXMert and generate affective captions. To this end we use the LSTM+FCN architecture as shown in figure, here we have an LSTM module that takes in a sequence of words as input and predicts the next word, along with this we have the visual representation from LXMert as the other input to a couple of dense layers that generate the output from the vocabulary.

Our motivation behind using this approach was to also allow the model to generate descriptive answers for questions passed to LXMert for tasks such as question answering. This model enables the LSTM to generate coherent sentence while taking into account the visual features from the vision head of LXMert. We tested if the model can caption the images correctly but unfortunately due to the lack of resources we could not train the model for long and hence could only get some mediocre results. Yet it is clear from the results that the model can correctly understand the main entity in the image and consistently mentions it in the captions albeit with some extra incoherent words.

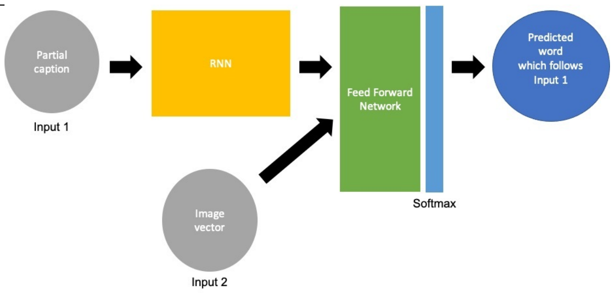

Figure 2: LSTM+FCN Architecture.

We also tried this model on our SentiQA dataset where the question and image features would be fed to the LXMert model and the rest of the training process would stay the same. Due to time and resource constraint we were not able to get the model to train for more than 7 epochs and this was not enough for the model to learn and generate good results.

We faced a myriad of issues when it came to training on Google Colab, the major one being the fact that we had to do a lot of preprocessing of data and hence we had exhausted the limits of using the GPUs pretty early. We also faced issues where our runtime environments would be reset randomly and we would end up losing a lot of our progress. Saving checkpoints onto our drive and proper scheduling was how we were able to use colab to its max limits.

7 Model evaluation

Here, we came up with a hybrid evaluation strategy as there is no simple way to compare generated text when there is no dataset available. For our particular task, if we extract the parts of the generated text which correspond to the added sentiment such as the adjectives, then we can directly use available standard metrics. We can then use coherence tests or sentiment analysis to check the

Model	B-1	B-2	B-3	B-4	ROUGE-L	METEOR	CIDEr	SPICE
Baseline: Vanilla LSTM	32.15	14.70	7.23	3.92	21.05	10.21	18.27	5.46
Baseline: SENTICAP-LSTM	33.97	15.65	7.90	3.70	24.43	11.73	28.37	7.78
SENTICAP-OMNINET	54.84	35.11	21.23	12.96	39.97	17.75	48.80	11.04

Table 1: Model comparison for scores of sentimental visual captions on standard evaluation metrics in percentages

validity of the entire generated sentence.

We evaluated our captions using the standard evaluation metrics following the works by Mathews et al. (2015) and Nezami et al. (2018), which include BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), METEOR (Denkowski and Lavie, 2014), CIDEr (Vedantam et al., 2015), and SPICE (Anderson et al., 2016). We implemented the evaluation referencing (Chen et al., 2018). However, the SentiCap dataset does not have the ground truth sentiment for each image; instead it provides captions for positive and negative emotions (Mathews et al., 2015), which is different from what we want to achieve. Therefore, a direct comparison between their models and ours would not be appropriate. Consequently, we excluded the sentiment part from our generated captions and check whether those captions are acceptable based on their ground truth captions in the MS COCO validation datset.

We have three models that we evaluated: Vanilla LSTM, SentiCap-LSTM and OmniNet. Vanilla LSTM model is trained on Flicker 8k dataset and SentiCap-LSTM is trained on SentiCap dataset as same as OmniNet. We did not include the evaluation of LXMert, partly due to the time constraint, but also because they were notably worse than our Vanilla LSTM outputs. We do provide a further analysis in Section 9 about the results of LXMert model.

Table 1 shows our result for the three models in details, and it seems that OmniNet model trained on the SentiCap dataset outperformed our baseline models. From this, we claim that OmniNet is well developed with better architecture for visual captioning.

Model	Positive	Negative
SENTICAP-LSTM	0.32	0.68
SENTICAP-OMNINET	0.65	0.35

Table 2: Sentiment Tendency of our models as in proportion of captions generated with positive/negative sentiment

Furthermore, we checked whether our models returned more positive captions or negative captions on both SentiCap LSTM and OmniNet models. The result is summarized in Table 2. We conducted sentiment analysis implemented by referencing (Face, 2020b) on the generated sentimental visual captions and found that overall, LSTM generated captions with negative sentiment more often than with positive sentiment; but Omninet generated captions in the opposite way, that is it generated captions with positive sentiment more often than with negative sentiment.

We do not have a clear explanation as to why this would happen. We look into this model behavior in more detail in the following section about the error analysis.

8 Error analysis of Baseline models

In this section, we provide an in-depth error analysis on some examples in order to further examine the model behavior. We use the CIDEr score as our threshold to pick the “Bad cases” and the “Good cases”. “Bad cases” are examples where both models scored the CIDEr score of 0 and there were 13 such example sentences out of 100 sentences we tested. “Good cases” is defined as the examples where both models scored the CIDEr score above 0.6. We compare the models on the basis of what sentiment do the generated sentences have, and whether the generated captions are grammatically acceptable. Lastly, we offer some examples for both bad and good cases and discuss the model behavior in more detail.

8.1 Sentiment Comparison

As discussed in Section 7, the captions generated by the LSTM were more negative than positive; whereas those generated by the OmniNet were more positive than negative. We look into the Bad cases and the Good cases to examine whether this observation still hods for each of the two kinds of cases. We examined the generated captions one by one for the Good and Bad cases, and counted the number of positive adjectives and negative adjectives. Positive adjectives included words like "happy", "pretty" and "nice"; and negative adjectives included words like "dead", "dirty", "crappy". If a sentence only contained negative adjectives, it was labeled as a negative caption; if it only contained positive adjectives, it was labeled as a positive caption. If a sentence ever contained a positive and a negative adjectives together, it was considered as a neutral caption. An example of a neutral caption is "a *good* game of *stupid* people around a tennis ball".

8.1.1 Bad cases

Model	Positive	Negative	Neutral
SENTICAP-LSTM	0.31	0.62	0.08
SENTICAP-OMNINET	0.46	0.46	0.08

Table 3: Sentiment Tendency of our models, for the cases where both models performed poorly (CIDEr score = 0), as in proportion of captions generated with positive/negative/neutral sentiment

For the Bad cases where both models scored the CIDEr score of 0, the sentiment tendency of the generated captions was similar for the LSTM model but different for the OmniNet model, compared to the overall sentiment tendency described in 2. As for the LSTM model, the generated captions were again more negative than positive. As for the OmniNet model, however, there were equal number of positive and negative captions. For each model, there was one neutral caption out of 13 captions in the Bad cases.

To sum up, given the same image, the LSTM was more likely to generate a negative caption and the OmniNet was equally likely to generate a negative or a positive caption. There were 6 out of 13 times where the two models matched in their generated sentiment, that is they both generated negative or positive captions.

8.1.2 Good cases

For the Good cases where both models scored the CIDEr score above 0.6, the sentiment tendency of the generated captions were even more biased than the overall tendency described in 2. As we discussed earlier, the captions generated by the LSTM were more negative than positive; whereas those generated by the OmniNet were more positive than negative. This tendency was exaggerated in the Good cases as in Table 4. below.

Model	Positive	Negative	Neutral
SENTICAP-LSTM	0.11	0.89	0.00
SENTICAP-OMNINET	0.11	0.78	0.11

Table 4: Sentiment Tendency of our models, for the cases where both models performed better (CIDEr score $\geq$ 0.6), as in proportion of captions generated with positive/negative/neutral sentiment

8.2 Grammatical Acceptability Comparison

Another important way to evaluate the model behavior is whether the generated captions were grammatical, regardless of whether it correctly describes the image. We checked the generated captions for the Bad and the Good cases manually and rated the captions as either grammatically acceptable or utterly ungrammatical. We do admit that it is perhaps a bit subjective but we tried to be as remain consistent in assessing the grammatical acceptability. Some of the grammatically acceptable captions are as follows: "a boy is being a frisbee in mid air" or "a happy man is sitting on a beautiful tree". These were labeled as grammatically acceptable, regardless of whether they correctly describe the image that they were supposed to describe. Some of the utterly ungrammatical sentences were as follows: "the angry child was was the crying baby he is the crying" or "the food was used look at all".

8.2.1 Bad cases

In general, for the Bad cases, the OmniNet was much better than the LSTM in generating grammatically acceptable captions. While the LSTM generated only 5 out of 13 grammatically acceptable captions, the OmniNet only generated one caption that was unacceptable. The only ungrammatical OmniNet caption was: "a person riding a horse pastdle is ready to ride a ride a past". Based on the counts of captions, we argue that even when we look at the Bad cases where the two models failed to generate captions relevant to the the OmniNet outperformed the LSTM model in generating grammatical captions, the Omninet at least generated captions that were more grammatically acceptable.

8.2.2 Good cases

On the other hand, as for the Good cases, both models were much better in generating grammatically acceptable captions. 4 out of 9 LSTM generated captions were acceptable and 6 out of 9

OmniNet captions were acceptable. However, ungrammatical captions still existed for both models even though all of these captions scored good CIDEr scores. In other words, we observed the limitation of the automatic evaluation metrics that they do not necessarily check the grammatical acceptability of the captions because they only check how many words were matched with the original captions. This point will be addressed in more detail with specific examples in the following section.

8.3 Manual analysis on some examples

Finally, we introduce some hand-picked examples that are interesting for both Bad cases and Good cases and discuss the model behaviors on them. We discuss two Bad case examples and suggest potential reasons why the models have failed on those examples. We also discuss two Good case examples and in doing so, we argue that the automated metric we used in Section 7 has some shortcomings in evaluating the image captions. We argue that the automatic evaluation metric sometimes can mistake a bad caption as a good one because it tries to match the N-grams.

8.3.1 Bad cases

We introduce two examples where we can intuitively infer why the models generated the captions that scored the CIDEr score of 0 because it makes in a way that the models "perceived" the images in the way they did.

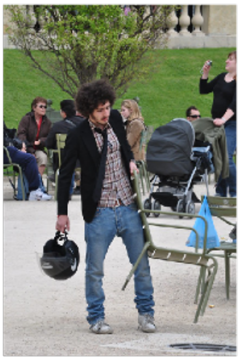

Figure 3: Example of an image for which both LSTM and OmniNet performed poorly (CIDEr score = 0)

The first one is in Figure 3. The ground truth image captions generally describe a man in the center holding a helmet and a chair. In the distant background however, there are a few people sitting on chairs which none of the caption mentions. While the LSTM generated caption mentions "large teddy" which is not understandable, the OmniNet caption mentions "a group of people sitting together on a bench", which might refer to the people in the background, rather than the man in the center. This suggests that the OmniNet could sometimes miss the most salient object in the picture and attend to the background more.

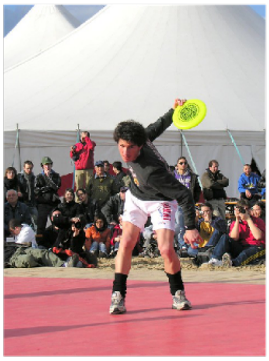

Figure 4: Example of an image for which both LSTM and OmniNet performed poorly (CIDEr score = 0)

The second example is in Figure 4. The image depicts a man playing a Frisbee in a competition. However, only one of the five ground truth captions mentions that the Frisbee that the man is holding is "neon yellow", which is really similar to the color of a tennis ball. From this perspective, it is somewhat understandable that the OmniNet generated a caption like "a game of people around a tennis ball". This example suggests that although the model failed to identify the object as a Frisbee, it might have attended to the right attribute of the object and found the most similar word that has the attribute.

8.3.2 Good cases

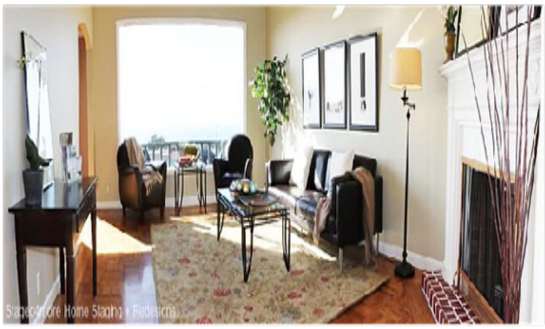

Figure 5: Example of an image for which both LSTM and OmniNet performed better (CIDEr score ≥ 0.6)

We introduce two examples that achieved higher evaluation metric result even though there was critical issue on the captions.

The first example is in Figure 5 which depicts a living room with multiple furniture. Both LSTM and OmniNet just provided a list of furniture which is not grammatically acceptable. LSTM returned "bad view of decorated living room dining area to kitchen area"; and OmniNet returned "a living room with a television table table, chairs, television, rotten wood table". This example suggests that as long as a caption contains all of the right object, it will score high on the automated metric, regardless of its grammatical acceptability.

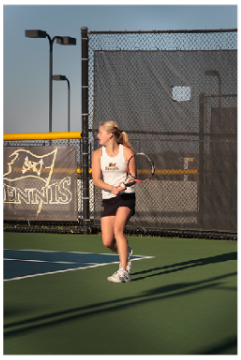

Figure 6: Example of an image for which both LSTM and OmniNet performed better (CIDEr score ≥ 0.6)

The second example is in Figure 7 which is an image of a woman playing tennis at court. Both LSTM and OmniNet returned grammatically correct captions but LSTM got higher score in evaluation metric even though it predicted the person playing tennis was a man. The generated caption of LSTM was "the tough guy is playing tennis player with the tennis match" and the CIDEr score was 0.7. The generated caption of OmniNet was "a pretty woman hitting a tennis ball with a tennis racquet" and CIDEr score 0.6. This clearly shows that even though we know that OmniNet returned a better caption, correctly identifying the person as a woman, it scored lower than LSTM because it matched a shorter sequence of N-gram.

9 Error analysis of LSTM + LXMert

There were two cases where we could see some promising results that made us believe there is some room to improve our custom LSTM + LXMert model in the future work. The first example is in Figure 7 below. The model generated caption for this image is"two zebras stand in field grass dead" which successfully captured the field, the grass and the two standing animals, though they were not detected correctly as giraffes. For another image that depicts several sheeps, the model generated the following caption: "two zebras stand in in the". This suggests that our model at least successfully captured when there are animals in the image. The generated captions are not grammatically acceptable however, since we did not train long enough.

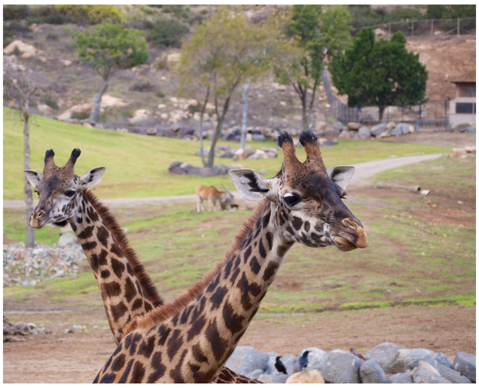

Figure 7: First example of an image for LSTM + LXMert

The second example is the same image as in Figure 4 and it is another instance that our model successfully captured the important features in the picture. The generated sentence was "dead of of people tennis court serve serve". Even though this is a picture about playing Frisbee, the caption still captured that it is a kind of sport activity. Also, as we discussed previously, the color of the Frisbee does look similar to that of a tennis ball. For another image that depicts a scene of a baseball game, the model also generated "dead man of to tennis tennis tennis tennis". In a similar way that the model was able to capture the presence of animals, it seems it can also capture a sport activity when there is one in the image. These examples suggest that if we can train on more dataset and fine-tune on our architecture, we expect there will be more promising results.

10 Contributions of group members

List what each member of the group contributed to this project here. For example:

- Young Se Kim: Data pre-processing, training baseline models, model evaluation and error analysis, building model

- Melnita Dabre: Data pre-processing, creating datasets, training baseline models, building model

- Chinmay Shirore: Training baseline model, data pre-processing, building model, model evaluation, error analysis, testing alternative approaches

- Seung Suk Lee: Training baseline models, model evaluation and error analysis, a lot of writings

11 Conclusion

The area of visual affective question answering has a lot of potential and remains relatively unexplored. Not only did we struggle to find related work, but there were no implementations, models or even simple datasets that could handle this task.

One important step we plan to proceed in this direction is to improve the SentiQA dataset. We plan to create more QA pairs, with more adjectives and emotions and replace many of the SentiCap answers with more coherent answers and a diverse question set.

Another potential application we mentioned in our initial proposal for such a mood-aware VQA/visual captioning model is on artworks (Garcia et al., 2020). While the existing work on the VQA on artwork is designed to generate answers about the image itself and the background knowledge of the image, such as the information about the painter, it has not yet incorporated the emotions expressed by the image into their model (Garcia et al., 2020). We can potentially test our model on the WikiArt Emotions dataset (Mohammad and Kiritchenko, 2018) which contains artworks and the corresponding emotions evoked in an observer.

One thing we were surprised about was that for some images, the generated captions were inappropriate and even rude. We realized that the sentiment of an image is arguably subjective and potentially a source of inducing a human bias into the dataset. A future work which investigates ways to minimize the ethical problems that could be introduced by human annotators is in order, before we can use such sentiment-aware captioning or question answering technologies in actual downstream tasks, for instance to help visually impaired people.

References

Agrawal, A., Lu, J., Antol, S., Mitchell, M., Zitnick, C. L., Parikh, D., and Batra, D. (2017). Vqa: Visual question answering. *Int. J. Comput. Vision*, 123(1):4–31.

Anderson, P., Fernando, B., Johnson, M., and Gould, S. (2016). Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

Chen, X., Fang, H., Lin, T.-Y., and Vedantam, R. (2018). *Microsoft COCO Caption Evaluation*.

Denkowski, M. and Lavie, A. (2014). Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380.

Face, H. (2020a). *Huggingface Tranformers Library: OpenAI GPT2*.

Face, H. (2020b). *Pipelines*.

Garcia, N., Ye, C., Liu, Z., Hu, Q., Otani, M., Chu, C., Nakashima, Y., and Mitamura, T. (2020). A dataset and baselines for visual question answering on art. *arXiv preprint arXiv:2008.12520*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hu, Z., Shi, H., Tan, B., Wang, W., Yang, Z., Zhao, T., He, J., Qin, L., Wang, D., et al. (2019). Texar: A modularized, versatile, and extensible toolkit for text generation. In *ACL 2019, System Demonstrations*.

Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., and Xing, E. P. (2017). Toward controlled generation of text. *arXiv preprint arXiv:1703.00955*.

Hudson, D. A. and Manning, C. D. (2019). Gqa: A new dataset for real-world visual reasoning and compositional question answering.

Jain, A. (2020). *Flickr 8k Dataset*.

Krishna, K., Wieting, J., and Iyyer, M. (2020). Reformulating unsupervised style transfer as paraphrase generation.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

Mathews, A., Xie, L., and He, X. (2015). Senticap: Generating image descriptions with sentiments. *arXiv preprint arXiv:1510.01431*.

Mohammad, S. M. and Kiritchenko, S. (2018). An annotated dataset of emotions evoked by art. In *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference (LREC-2018)*, Miyazaki, Japan.

Nezami, O. M., Dras, M., Wan, S., and Paris, C. (2018). Senti-attend: image captioning using sentiment and attention. *arXiv preprint arXiv:1811.09789*.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Pramanik, S., Agrawal, P., and Hussain, A. (2019a). Omninet: A unified architecture for multi-modal multi-task learning. In *ICLR 2020 Conference*.

Pramanik, S., Agrawal, P., and Hussain, A. (2019b). *OmniNet: A unified architecture for multi-modal multi-task learning*.

Priya, A. and Dwivedi, P. (2018). *Face and Emotion Detection*.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ruwa, N., Mao, Q., Song, H., Jia, H., and Dong, M. (2019). Triple attention network for sentimental visual question answering. In *Computer Vision and Image Understanding 189*.

Ruwa, N., Mao, Q., Wang, L., and Dong, M. (2018). Affective visual question answering network. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 170–173.

Serengil, S. I. and Ozpinar, A. (2020). Lightface: A hybrid deep face recognition framework. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 23–27. IEEE.

Vadlamudi, B. (2019). *Use Pytorch to create an image captioning model with pretrained Resnet50 and LSTM and train on google Colab GPU (seq2seq modeling)*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *NIPS 2017 Conference*.

Vedantam, R., Lawrence Zitnick, C., and Parikh, D. (2015). Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Xu, H. and Saenko, K. (2016). Dual attention network for visual question answering. In *ECCV 2016 2nd Workshop on Storytelling with Images and Videos (VisStory)*.

You, Q., Jin, H., and Luo, J. (2017). Visual sentiment analysis by attending on local image regions. In *Proceedings of the thirty-first AAAI conference on artificial intelligence*, pages 231–237.