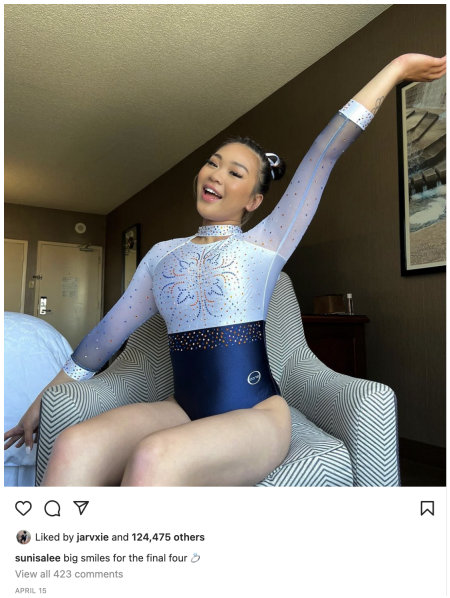
Social Media Image Captioning

Akash Karanam, Ayush Sachdeva, Nathan Powell

I. Introduction

Image captioning is one of the core tasks at the intersection of Computer Vision and Natural Language Processing. The standard deep learning approach for this task is to pass an image through a trained Convolutional Neural Network (CNN) to extract a feature matrix and then hand off this feature matrix to a recurrent neural network (RNN) to generate a sequence of words that form the caption. However, the task of social media image captioning is far more nuanced since a functional model needs to understand cultural context, sarcasm, and hyperbole in addition to the visual and semantic elements in the image. Furthermore, the grammar and syntax of social media captions differs from ordinary captions because social media captions try to elicit a response from the reader while ordinary captions are more objective and thus use a more consistent grammatical structure. For example, an image captioner might caption the image in Figure 1, ”This woman is seated and smiling”. However, the true caption for the image is ”Big smiles for the final four”. Note that while it is true that the image shows a woman that is seated and smiling, one would need information about the cultural and temporal context of the final four in addition to the semantic and visual elements of the image to make the true caption.

Fig. 1: Example Instagram Picture



II. Dataset

Our data is drawn from the Redcaps dataset which contains 12 million image-caption pairs collected from Reddit. This data was collected over 13 years of Reddit history and draws from 350 subreddits. [1] However, the Redcaps data set does not store the image directly, but rather a URL that contains the image. Thus, to wrangle this dataset and prepare it for modeling the creators of the dataset used a GPU cluster to get all the images from their respective web endpoints. Due to our compute limitations on Google Colab we chose to focus only on subreddits related to cat images which reduced the dataset to approximately 100,000 images. To determine which subreddits to focus on we passed a random sample of 100 images from each subreddit through Google’s ’Show and Tell’ image captioner and used BLEU (Bilingual Evaluation Understudy) to compare the ’Show and Tell’ caption to the caption in the Redcaps dataset. [2] The purpose of this experiment was to see how similar the Redcaps captions were to a literal, descriptive caption. We found that subreddits about landscapes, urban architecture, and food tended to have very literal captions while captions about cats tended to be more nuanced and more similar to social media captions. Moreover, the size of the reddit dataset limited the number of subreddits we could use to train our model. Hence, we chose the subreddit ”r/catpictures” and specifically trained our model to caption cat pictures from the time period 2017-2021. Even though this particular model focuses on captioning cat pictures, this modeling process in theory should be extendable to social media captions in general due to the similarity between the grammar and style of cat captions and social media captions.

III. Data Wrangling

The initial phase of our data wrangling pipeline involved using HTTP GET requests to get the image data from the URLs in the dataset and then converting these images to RGB tensors so the image could be passed into a Convolutional Neural Network (CNN). Since there is no uniformity in the size of images on Reddit we resized all of the images to 256×256 and thus our RGB tensors have dimension $256 \times 256 \times 3$. We also normalized the tensors on each of the three channels. This process of retrieving the image and converting it to a tensor is very RAM intensive and only ~ 6000 images could be processed before exceeding the 12GB RAM limit on Google Colab. Hence, we decided to process the 25000 images we were using for training into 5 different chunks of 5000 images. For each image in a given chunk we stored the RGB tensor representing the image, the image caption, and the image URL in a tuple and added the tuple to a list corresponding to the chunk. To prevent repeated reloading of this list we pickled the end state of the list and stored it on Box for each of the 5 training chunks as well as a chunk of images for the validation set. We made sure to include cat images from multiple different years to ensure that our training set and validation set are diversified.

IV. Keyword Generation

One important element of our project proposal that was missing in the Redcaps dataset and most social media datasets is keywords to help guide the caption. While the model will be tested with user provided keywords we needed to train our image captioner with keywords. Since the only textual information in our dataset was the captions we decided to generate keywords for each of the images in the dataset using the caption. Our aim was to identify the keywords that most closely matched the sentiment of the overall caption.

The first step of the keyword extraction process involved identifying candidate keywords. To do this we used sci-kit learns CountVectorizer module and iterated through the captions using a trigram(three words at a time) model. Next we used Distilbert to convert the caption and the candidate keywords to numerical data because it is optimized for similarity tasks which is critical in keyword extraction. After creating the BERT embeddings we still needed to reduce the set of candidate keywords to the actual keywords. The key observation here is that just because a word occurs with high frequency in a caption does not mean that it is a keyword. The word must also be similar to the overall caption. Note that for our caption this idea of similarity to the overall caption is critical since captions are so short and usually don't have repeated words. To determine which candidate keywords most closely represent the caption we computed the cosine similarity between the caption and the candidate keyword embedding vectors and chose the top keywords. We chose cosine similarity because it performs well in high dimensionality.

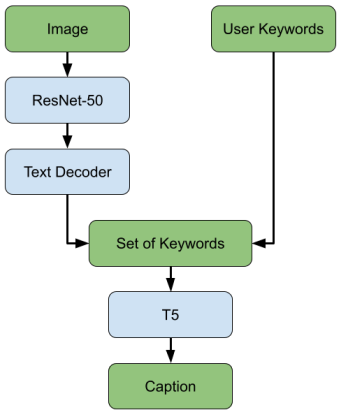
Unfortunately, when we ran this model on the captions in our dataset the keywords were very unbalanced and most of the keywords were synonyms of cat or related to common features of cats. This was particularly concerning because many of these words could be generated just by using the image and thus were not necessary as keywords. The main drawback of using cosine similarity to generate keywords was that the keywords were not very diversified since words that closely represent the overall meaning of a caption tend to be very similar to one another. To increase the semantic diversity of our keywords without generating keywords that did not represent the captions we used a max sum similarity algorithm. Specifically, our goal with this algorithm was to maximize the candidate keyword similarity to the caption while also minimizing the similarity among the candidate keywords. Even with this improvement our keywords were heavily skewed towards nouns and words related to cats so we decided to use a pretrained model to utilized many of our techniques. We specifically used the KeyBERT model since it used BERT embeddings and has a very similar architecture to the BERT transformer that we discussed in class.

KeyBERT also ranks the keywords from most important to least important and had the option to return the n most important keywords where n is between 0 and the number of words in the caption. This made it very easy to augment the image URL, caption, RGB tensor tuples in our dataset with a list of keywords. The ranking of the keywords was also very helpful when we evaluated our captions with 0,1,2,3,4, and 5 keywords. [7]

TABLE I: Keyword to Caption Generations

Keywords	True Caption	Generated Caption
oliver, collar, adopted, dashing, halloween	we adopted little oliver today. dashing is he with his new halloween collar.	adopted oliver in a dashing halloween collar.
tiger tibby	my tibby tiger	tibby tibby tiger
alice, meet	meet alice!	meet alice! she's so alice.
zara, sister, 16, lovely, post	as i posted her sister i feel it only fair to also post a picture of zara. also 16 and just as lovely as her sister.	my sister's 16-year-old zara. i'm so happy

Fig. 2: Full Model Pipeline



V. Modeling

We divide our modelling process into two different sections. The first section focuses on generating keywords purely based on the image. These keywords are usually nouns or verbs. For the second section of the modeling process, we use the keywords generated by our image model and keywords entered by the user and generate the final social media caption for the image.

A. Image to Keywords Modeling

We modeled this problem as the classic caption generating pipeline. The goal of this model is to receive an image and generate keywords based on this image. Hence, our training data for this model are in the form of pairs of images and keywords from their corresponding captions.

We start with a Convolutional Neural Network (CNN) pretrained on the Resnet-152 [6]. We use the transfer learning in order to ensure that our image masks are effective in identifying objects from our dataset. Even thought we start with the pretrained weights from the base model, we retrain this model on our dataset in order to ensure a feature map which is particularly useful for our dataset. For example, one of the most common characteristics in our dataset is the image of a cat yawning. Hence, by retraining the model, our model gets better at recognizing that particular actions.

We pass the feature map of size 20 which is the CNN's output to a Text Decoder which is comprised of a Recurrent Neural Network (RNN). Note that this RNN is not pretrained and is only trained on our data. This is integral as keywords for cat captions are very different from keywords for usual descriptive captions. The output of this RNN is a string comprising of the keywords in decreasing order of importance.

Finally, while training this combined model, we use the cross entropy loss while comparing the predicted keywords and the actual keywords.

B. Keywords to Caption Modeling

In order to allow our model to use both a trained model on the image and input keywords from the user to generate a caption, a model that could take strings as input and output a sentence was a necessity. In order to meet this requirement, we underwent extensive research to find the best keyword-to-caption generation model. One of the more promising libraries that we found for this purpose was the KeyToText library [3], which would allow the loading of pre-trained models that had been very well fitted for this particular task. However, we ultimately chose to use the more generalized transformer models that the KeyToText library was built on, the T5 Transformer models trained by Google.

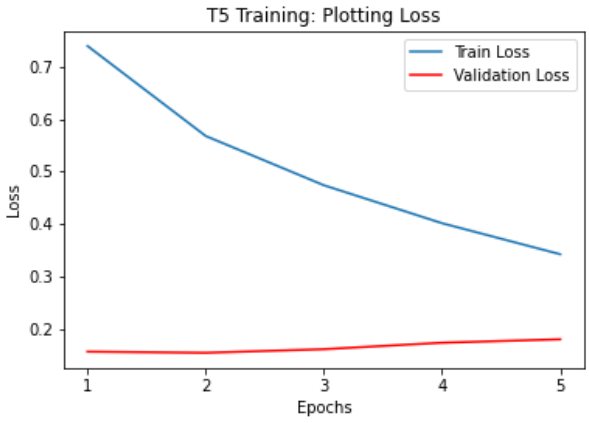
Google's T5 [4], a Transformer model in which all NLP tasks conform to the requirement that both the model's input and output are text strings, has a wide variety of possible use cases. Loaded from the transformers library [5], we were

(a) ”harness”, ”waiting”, ”come”

(b) ”yawn”, ”queen”

Fig. 3: Image to Keyword Examples

Fig. 4: T5 Modelling Loss



able to begin with the pre-trained weights of the basic T5 and immediately experiment with caption generation. Without any training, the T5 would simply return the keywords given as inputs as the generated caption, which was obviously not satisfactory. However, after creating a training and validation cycle, we were able to see immediate results that are emblematic of the promise that transformer models provide. After either one or two epochs of training on our combined keyword and caption training set, we would already see the signs of overfitting: the validation loss would begin to steadily climb, while the training loss would continue its downward descent. See Figure 4 for a plot of our training and validation losses against the number of epochs. In addition, we have included some sample outputs from our keyword to caption model, passing in a select random subset of keyword inputs from our testing set as in Table 1 that represent the overall trends of our model. As can be observed from these outputs, our keyword to sentence model does struggle somewhat with specific words that are rarely seen, such as the ’tibby’ in the true caption ’my tibby tiger’. However, the captions generated by the T5 model, while not an exact replica, are satisfactory in their similarity to the true captions from the dataset. A great example of a satisfactory can be seen at the bottom of Table 1, as our model is able to create multiple legible sentences from a large amount of inputs, including the rare ’zara’. Even though the keyword to caption portion of our model is by no means perfect, it is more than capable of generating satisfactory captions that a human can understand. As such, we decided that our final model was ready to be created.

VI. Evaluation

While a subjective examination of our results to this point has significant merit when it comes to understanding our model and its performance, using objective metrics to examine trends in our model also has merit. To start, we evaluated the performance of the model on the full test set. To do this, we needed a wide variety of evaluation metrics. First, we studied the metrics that lent themselves well to machine translation and/or text summarization [8]. This includes the metrics such as $BLEU$, $METEOR$, and $ROUGE$. $BLEU$, or BiLingual Evaluation Understudy, is a metric that was created for evaluating machine translations, and relies on n-gram precision scores and a brevity penalty. $METEOR$, or

Image	Generated Keywords	User Keywords	Generated Caption
	”laundry”	”cutest”	”cutest laundry machine ever”
	”tree”, ”christmas”, ”ready”	”excited”	”excited about the christmas tree”

TABLE II: Predicted captions

Metric for Evaluation of Translation with Explicit ORdering, is a metric that improves on $BLEU$ in its ability to go beyond explicit matches in words by utilizing synonym matching [8]. $ROUGE$, or Recall-Oriented Understudy for Gisting Evaluation, is a similar metric to $BLEU$, but is more dependent on recall and is therefore kinder to longer sentences. In addition to these machine translation-based metrics, we utilized the newer $CIDEr$ metric that is designed more for image caption evaluations [8]. Because our dataset did not come with multiple captions, $CIDEr$ effectively functions as a cosine similarity measurement between n-grams of both the reference and candidate solutions.

In Table III, we see the scores from each of our evaluation metrics. Of note is the significant decrease in score with longer n-grams for $BLEU$, which signifies that long-term trends in the true captions are harder to capture than just including the words themselves is. This makes sense, as we have a large number of keywords given from running KeyBERT on the reference solutions for the predictions that are incredibly likely to be included in the candidate solutions. In addition, social media captions do not have to follow clear grammar rules, which means that longer phrases are less likely to follow patterns on a large scale.

In addition, we were very interested in the effect that the number of keywords has on the performance of the caption generation. In Table IV, we see the scores from each of our metrics using different numbers of keywords as specified in the top column. The scores themselves are all from the same set of test images that all have at least 5 KeyBERT output keywords from KeyBERT on their respective reference solutions. This allowed us to change the number of given and utilized keywords in the predictions from 0 to 5, inclusive, while maintaining the same data to ensure proper comparability across these independent scorings. Of note with the captions is that the requirement of at least 5 keywords being available requires very long captions, which does decrease the accuracy of many of the metrics. Because of this, the $ROUGE$ metric does relatively well compared to the precision-based metrics, as its recall-based nature means that longer sentences are preferred. Of note within the scores themselves is how quickly and consistently $CIDEr$ increases, even while all of the other metrics suffer a confusing decrease in scoring with 4 keywords when compared to either 3 or 5 keywords used. In addition, another interesting point is the differences between different $BLEU$ scores, indicating the ease at which the model matches individual words with increases in keywords but without seriously accurate output indicative of the overall structure of the caption itself. These metrics do indicate the importance of including all of the keywords possible, as the accuracy does increase with more keywords given.

TABLE III: Metrics for Caption Matching, Full Test Set

Evaluation Metric	Performance
$BLEU_1$	0.400
$BLEU_2$	0.258
$BLEU_3$	0.203
$BLEU_4$	0.156
$METEOR$	0.282
$ROUGE-L$	0.350
$CIDEr$	1.392

TABLE IV: Metrics for Caption Matching, Limiting Given Keywords

Evaluation Metric	0	1	2	3	4	5
$BLEU_1$	0.053	0.111	0.176	0.222	0.182	0.280
$BLEU_2$	0.000	0.000	0.000	0.118	0.095	0.221
$BLEU_3$	0.000	0.000	0.000	0.000	0.000	0.167
$BLEU_4$	0.000	0.000	0.000	0.000	0.000	0.125
$METEOR$	0.011	0.043	0.078	0.119	0.116	0.231
$ROUGE-L$	0.045	0.090	0.093	0.180	0.121	0.266
$CIDEr$	0.064	0.128	0.168	0.327	0.401	1.321

VII. Conclusion

Through the development of this social media captioning model we have come across some promising results as well as a number of pitfalls that have taught us about the value and difficulty of the social media image captioning task. There are over 2 billion images shared on social media every day and the majority of them are captioned images. Social media captions are a source of social, cultural, and even monetary capital. This is a task that most humans have engaged in and a program that can caption these images is immensely valuable. However, unlike the task of ordinary image captioning, there is no true "right answer" for captioning a social media image. One's social media image caption does not just depend on the actual image but also the user's audience, other content, and even the social media platform they are using. We found that our model and evaluation metrics struggled with this lack of uniformity among social media captions. Even within the curated Redcaps dataset the captions varied drastically in length, grammatical correctness, and relevance to the image. Our model was largely able to comment on the visual and semantic elements in the image and understood some of the nuanced social connotations of the images but there was a lot of variability in the input dataset that created noise for the model.

References

[1] RedCaps: web-curated image-text data created by the people, for the people
https://doi.org/10.48550/arXiv.2111.11431

[2] Show and tell: A neural image caption generator
Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan
http://arxiv.org/abs/1411.4555

[3] Keytotext: Keywords to Sentences. Gagan Bhatia (2022).
https://github.com/gagan3012/keytotext

[4] Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu (2019).
https://arxiv.org/abs/1910.10683

[5] T5. HuggingFace. https://huggingface.co/docs/transformers/model_doc/t5

[6] Deep Residual Learning for Image Recognition.
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun
https://arxiv.org/abs/1512.03385

[7] Minimal keyword extraction with BERT
https://maartengr.github.io/KeyBERT/

[8] Re-evaluating Automatic Metrics for Image Captioning.
Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem (2016). https://arxiv.org/pdf/1612.07600.pdf