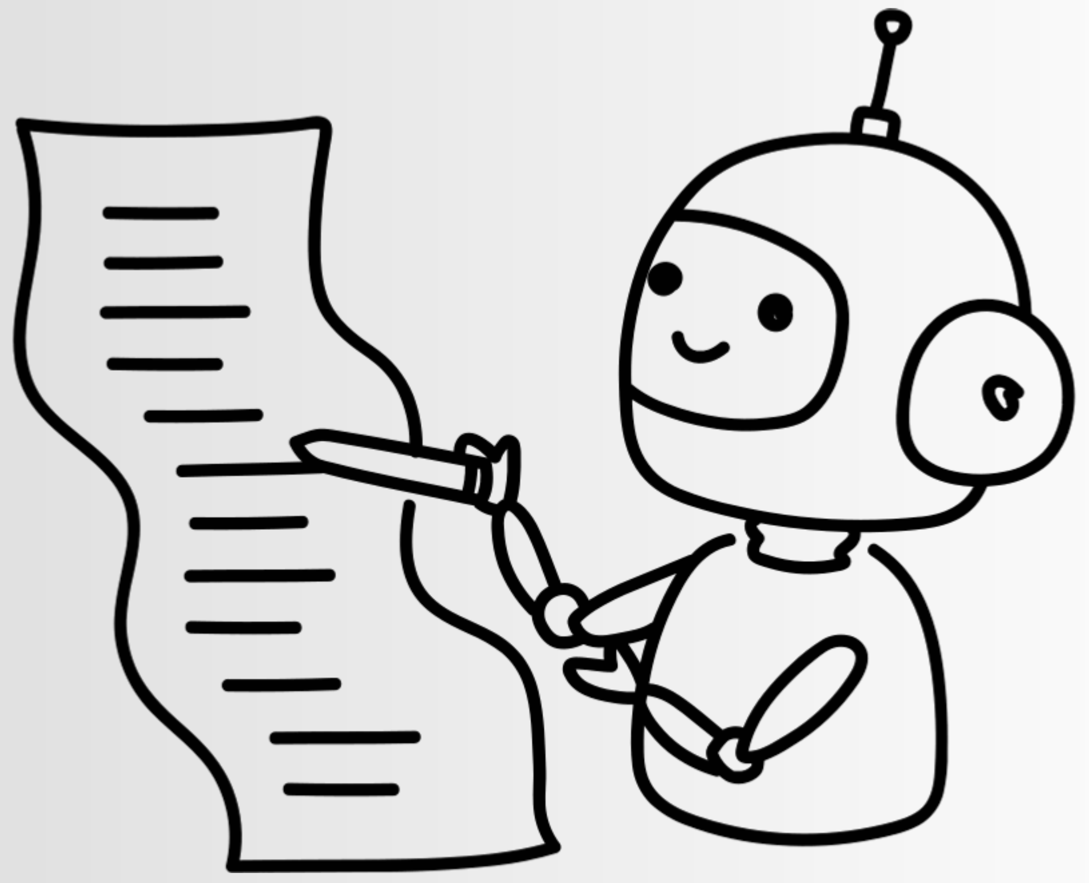
Image Captioning using Hugging Face Vision Models

BLIP



22224516 Gurung Yash
22224014 Khatri Priya
22224040 Sagar Bhusal

Table of Contents:

Image Captioning with Hugging Face Model

In this project, we explore image captioning using Hugging Face’s BLIP model. Both the dataset and model were imported from Hugging Face.

- **Dataset**: We used the flickr8k dataset
- **Model**: BLIP

Machine Learning Model

BLIP Model Overview

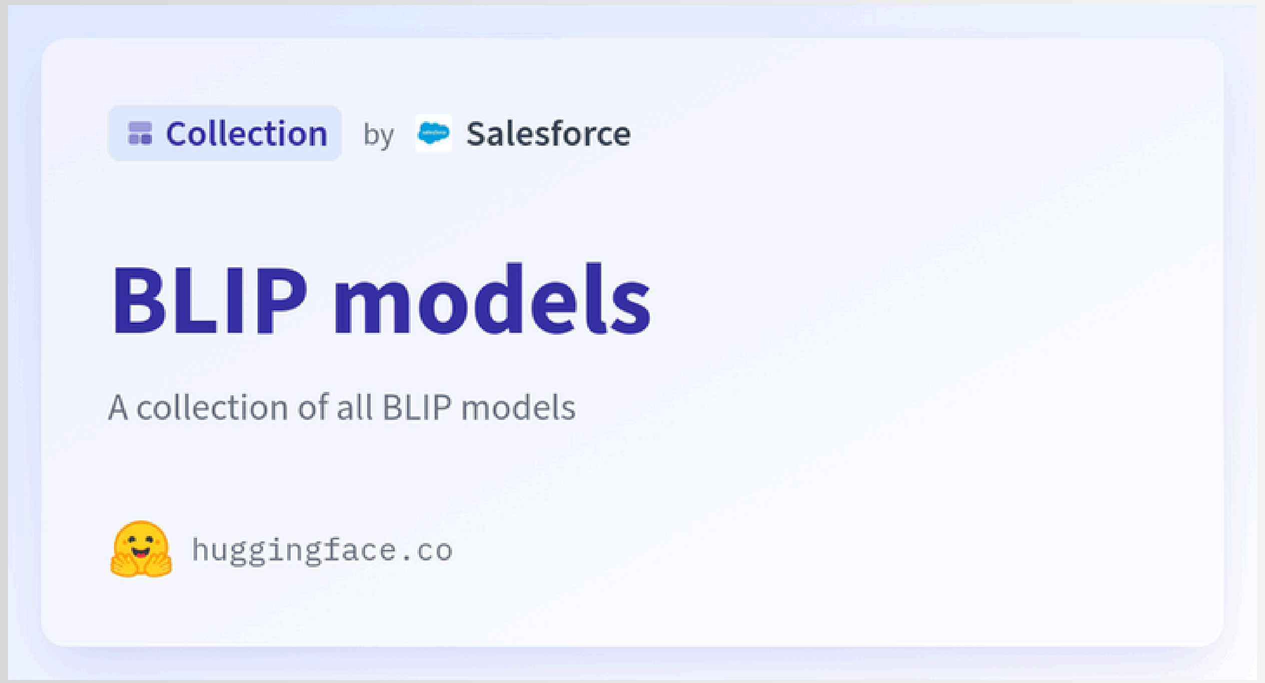
- **Creator:** The BLIP model was developed by researchers at Salesforce. Some of the members: Junnan Li, Dongxu Li, Caiming Xiong, Steven Hoi.

- **Purpose:**

BLIP (Bootstrapped Language-Image Pre-training) is a unified vision-language model designed for tasks like image captioning, image-text retrieval, and visual question answering.

- **Project Utility**:

BLIP's pre-trained framework enables us to efficiently generate captions for images by leveraging its state-of-the-art performance in vision-language tasks.



Collection by Salesforce

BLIP models

A collection of all BLIP models

huggingface.co

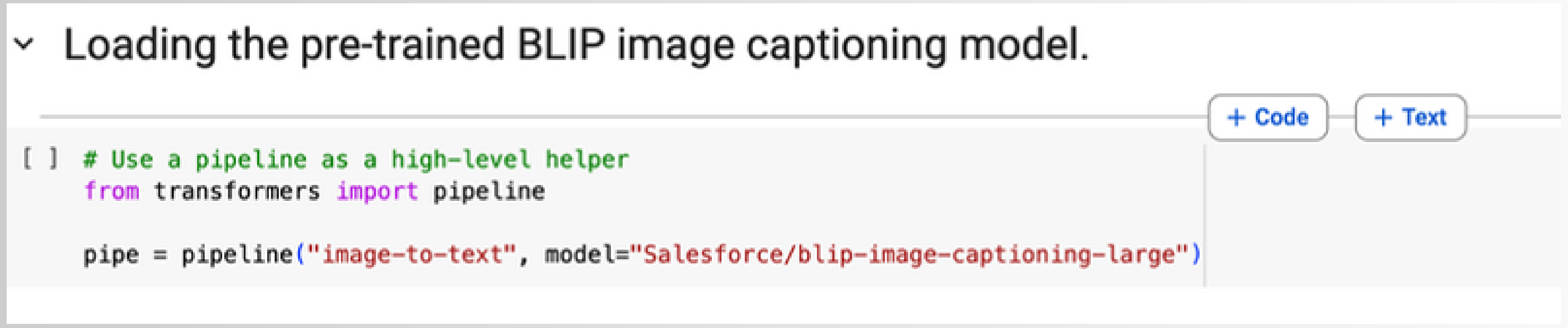
Key Features of BLIP

BLIP is a model that is able to perform various multi-modal tasks including:

- Visual Question Answering
- Image-Text retrieval (Image-text matching)
- Image Captioning

Methods and Workflow

1. **Loading the Pre-trained BLIP Model**

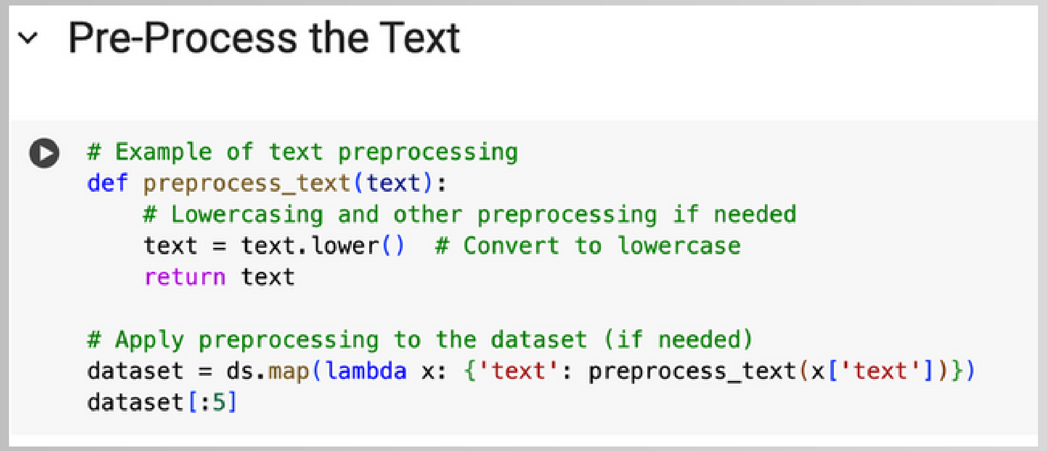


Loading the pre-trained BLIP image captioning model.

```
# Use a pipeline as a high-level helper
from transformers import pipeline

pipe = pipeline("image-to-text", model="Salesforce/blip-image-captioning-large")
```

2. **Data Preprocessing**

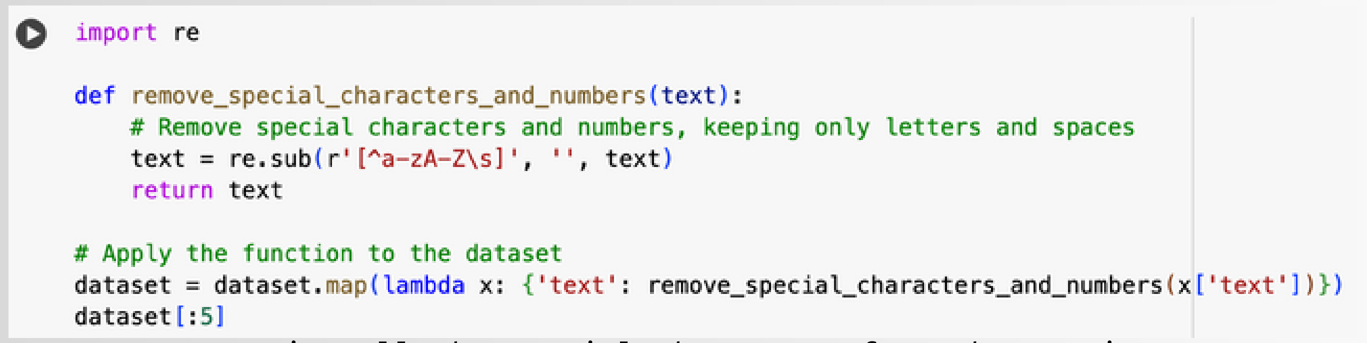


Pre-Process the Text

```
# Example of text preprocessing
def preprocess_text(text):
    # Lowercasing and other preprocessing if needed
    text = text.lower()  # Convert to lowercase
    return text

# Apply preprocessing to the dataset (if needed)
dataset = ds.map(lambda x: {'text': preprocess_text(x['text'])})
dataset[:5]
```

make the text in the captions all in lower case



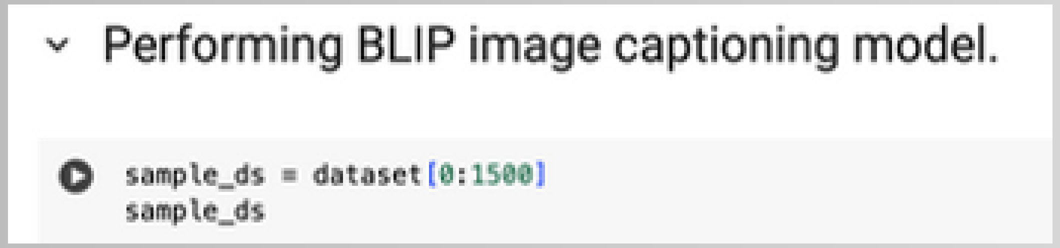
```
import re

def remove_special_characters_and_numbers(text):
    # Remove special characters and numbers, keeping only letters and spaces
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    return text

# Apply the function to the dataset
dataset = dataset.map(lambda x: {'text': remove_special_characters_and_numbers(x['text'])})
dataset[:5]
```

Removing all the special characters from the captions

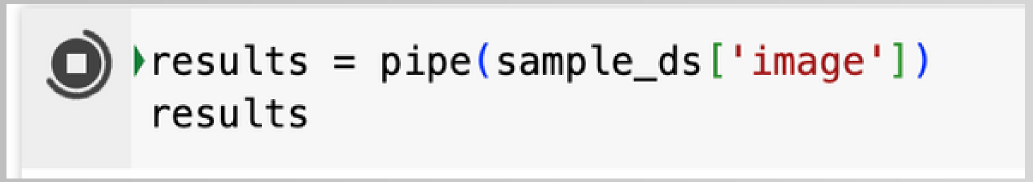
3. **Dataset Sampling**



Performing BLIP image captioning model.

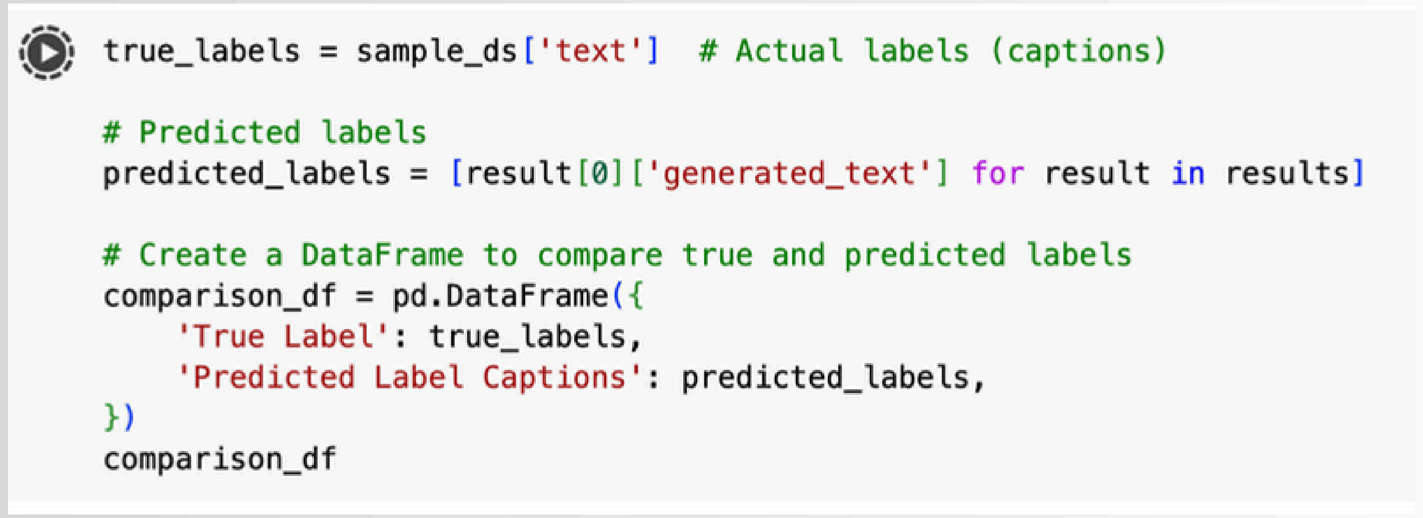
```
sample_ds = dataset[0:1500]
sample_ds
```

4. **Caption Generation**



```
results = pipe(sample_ds['image'])
results
```

- Generated captions using BLIP were treated as predicted_labels.
- Original captions were retained as true_labels for comparison.

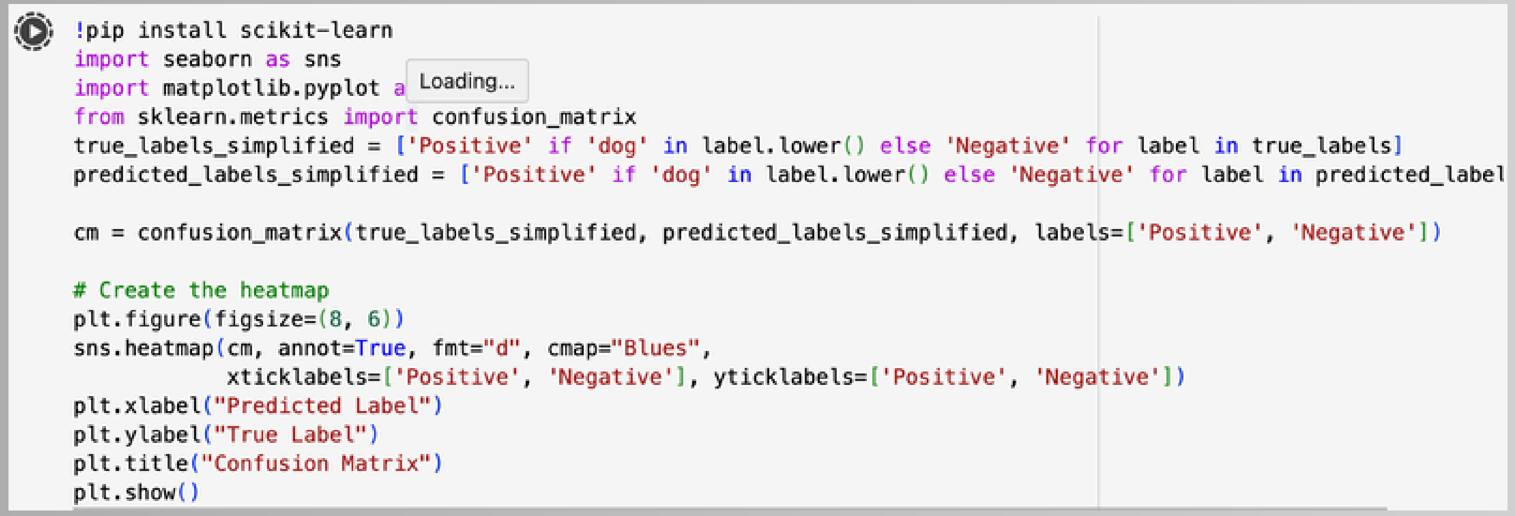


```
true_labels = sample_ds['text']  # Actual labels (captions)

# Predicted labels
predicted_labels = [result[0]['generated_text'] for result in results]

# Create a DataFrame to compare true and predicted labels
comparison_df = pd.DataFrame({
    'True Label': true_labels,
    'Predicted Label Captions': predicted_labels,
})
comparison_df
```

5. **Evaluation Using Confusion Matrix**



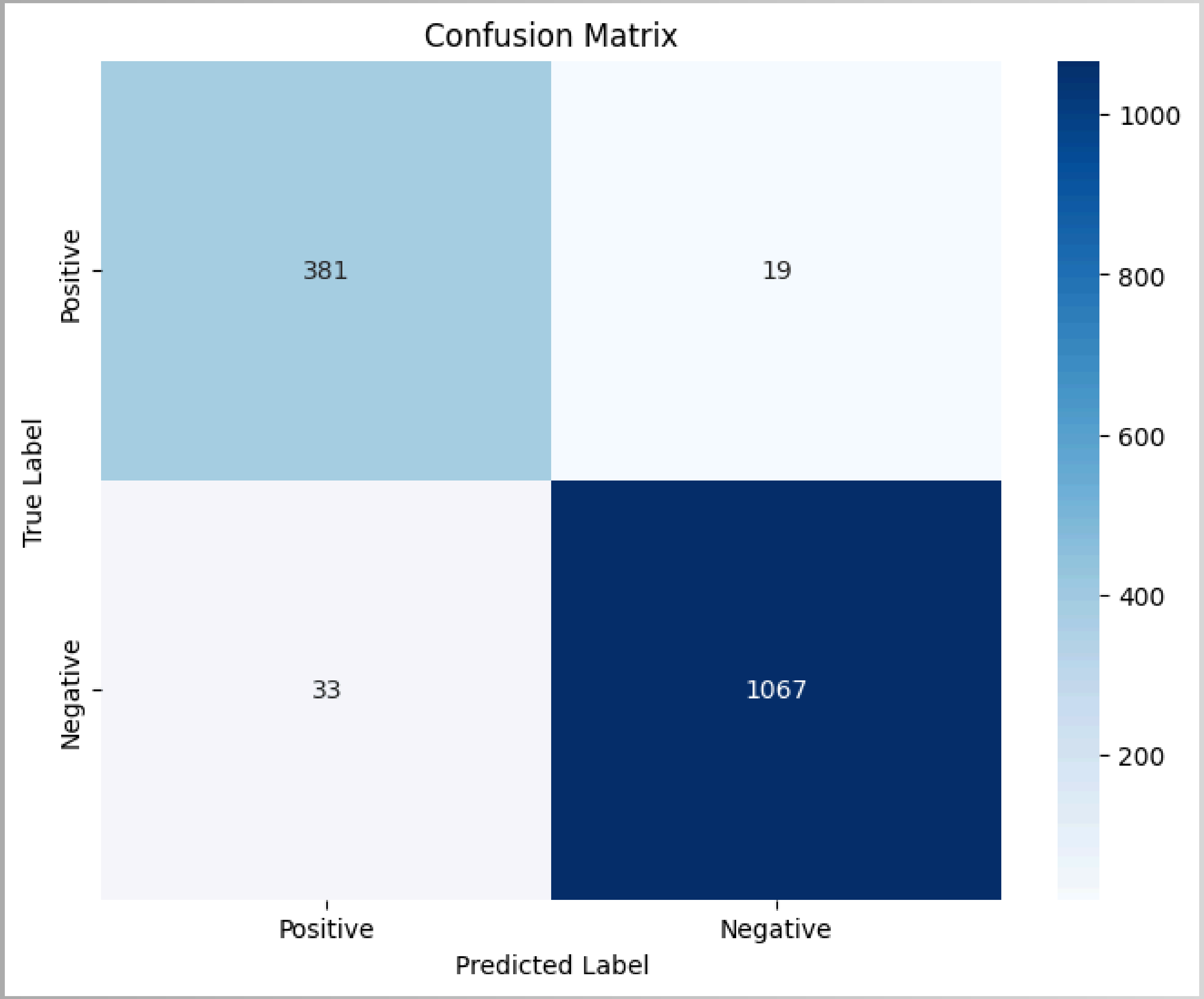
```
!pip install scikit-learn
import seaborn as sns
import matplotlib.pyplot a
from sklearn.metrics import confusion_matrix
true_labels_simplified = ['Positive' if 'dog' in label.lower() else 'Negative' for label in true_labels]
predicted_labels_simplified = ['Positive' if 'dog' in label.lower() else 'Negative' for label in predicted_label

cm = confusion_matrix(true_labels_simplified, predicted_labels_simplified, labels=['Positive', 'Negative'])

# Create the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
            xticklabels=['Positive', 'Negative'], yticklabels=['Positive', 'Negative'])
plt.xlabel("Predicted Label")
plt.ylabel("True Label")
plt.title("Confusion Matrix")
plt.show()
```

1. Simplification of Labels:
   - true_labels_simplified and predicted_labels_simplified categorize labels as "Positive" if they contain the substring "dog" (case-insensitive); otherwise, they are categorized as "Negative."
2. Confusion Matrix Calculation:
   - confusion_matrix computes the confusion matrix using the simplified labels and specifies the order of labels as ['Positive', 'Negative'].

HeatMap



True Positive (TP): 381

- This represents the number of cases where the model correctly identified a positive instance.

True Negative (TN): 1067

- This is the number of cases where the model correctly identified a negative instance.

False Positive (FP): 19

- These are cases where the model mistakenly predicted a positive label for a negative instance.

False Negative (FN): 33

- These are cases where the model failed to predict a positive label and instead predicted it as negative.

Evaluation Metrics

1. **BLEU:**

BLEU (Bilingual Evaluation Understudy) is an NLP metric for tasks like machine translation. It Shows how much the generated text matches the reference text word-for-word.

Scores range from 0 to 1, with higher scores showing better similarity. However,it prioritizes precision over recall, so a high score doesn’t always mean perfect quality.

2.CIDEr:

CIDEr (Consensus-based Image Description Evaluation) is a metric for image captioning that measures how similar a generated caption is to human-written references.It focuses on matching words that are rare and important, comparing them to human-written captions.

Scores range from 0 to 1,with higher scores indicating better alignment and descriptiveness compared to human references.

3.SPICE:

SPICE (Semantic Propositional Image Caption Evaluation) measures the quality of image captions by comparing their semantic content to human references.Shows how well the meaning or relationships in the generated caption match human-written captions.

Scores range from 0 to 1, with higher scores indicating better semantic alignment with human-written captions.

Evaluation Metrics (Scores)

1. **CIDEr: 0.6263**
   - **Similarity:** The generated captions align well with the overall intent and relevance of the true captions. This score indicates that the AI captures the core ideas, themes, or messages present in the reference captions.
   - **Implication:** Even if the specific words or phrasing differ, the AI generally conveys the same meaning or purpose.
2. **BLEU: 0.0211**
   - **Similarity:** Minimal overlap in exact word choices and n-gram sequences between AI-generated and true captions.
   - **Implication:** While the intent is captured (as seen in CIDEr), the AI often rephrases or uses synonyms, resulting in low exact match scores. This suggests a difference in surface-level structure.
3. **SPICE: 0.2234**
   - **Similarity:** The AI captures broad semantic elements, such as objects, attributes, and their relationships, though not with high detail.
   - **Implication:** The AI is able to describe the content reasonably well, but it may struggle with more fine-grained details or complex relationships that are present in true captions.

Conclusion

This project shows how effective Hugging Face’s BLIP model is for image captioning, with CIDEr being the most useful metric. The CIDEr score of 0.6263 indicates that the model captures the main ideas and messages of true captions well. Unlike BLEU, which penalizes differences in word choice, CIDEr focuses on overall meaning, making it a better measure of caption quality.

In summary, the BLIP model offers a powerful framework for unified vision-language tasks, achieving state-of-the-art performance in generating captions. Through this project, we have established a robust pipeline for image captioning and laid the groundwork for future enhancements in this exciting field of AI.

References

BLIP. (n.d.). https://huggingface.co/docs/transformers/en/model_doc/blip

Bleu score (n.d.). Stack Overflow. https://stackoverflow.com/questions/69178581/my-bleu-score-is-different-from-nltk-bleu-score

Tylin. (n.d.). GitHub - tylin/coco-caption. GitHub. https://github.com/tylin/coco-caption

Naveengo/flickr8k · Datasets at Hugging Face. (n.d.). https://huggingface.co/datasets/Naveengo/flickr8k