

Generowanie tekstu z obrazków przy użyciu transformatorów 'image-to-text'

Maciej Biegan
Kamil Dziedzic
Jan Bobak
Jan Adamowicz

Abstrakt

Celem projektu było stworzenie narzędzia do generowania opisów tekstowych na podstawie obrazków z wykorzystaniem modeli 'image-to-text' dostępnych w portalu Hugging Face. Narzędzie opiera się na transformatorach, które umożliwiają zaawansowane przetwarzanie obrazów i generowanie opisów w różnych trybach. Projekt obejmował implementację, testowanie i ocenę wydajności narzędzia.

Wstęp

2.1 Cel

Celem projektu było stworzenie narzędzia, które generuje tekstowe opisy obrazów za pomocą transformatorów 'image-to-text' dostępnych na portalu HuggingFace [1]. Transformatory stanowią podstawę zaawansowanych modeli głębokiego uczenia wykorzystując modele BLIP (Bootstrapped Language-Image Pretraining). Aplikacja umożliwia wybór trybu generowania (prosty lub zaawansowany). Dzięki takiemu podejściu użytkownicy mogą dostosować szczegółowość generowanych opisów do swoich potrzeb, co zwiększa uniwersalność narzędzia. Ponadto projekt zakładał, że użytkownik będzie mógł przysyłać obrazy zarówno z lokalnych zasobów, jak i za pomocą odnośników URL, co zapewnia elastyczność w obsłudze różnych źródeł danych.

2.2 Zakres

Projekt obejmował:

- Implementację narzędzia do generowania opisów obrazów w języku python.
- Testowanie działania modelu na różnych obrazach oraz parametrach
- Analizę czasu działania i jakości generowanych opisów w zależności do zastosowanego modelu.

2.3 Metodyka

Do realizacji projektu zastosowano podejście oparte na najlepszych praktykach w dziedzinie przetwarzania języka naturalnego oraz wizji komputerowej, wykorzystując technologie open-source, takie jak Hugging Face. Wykorzystane zostały modele Salesforce/blip-image-captioning-base, Salesforce/blip-image-captioning-large oraz bibliotekę Pillow do przetwarzania obrazów. Kod został napisany w języku Python, a testowanie przeprowadzono na obrazach wprowadzonych zarówno lokalnie, jak i przez URL. Za pomocą biblioteki **transformers**, obrazy przekształcano na wektorowe reprezentacje przy użyciu modułu kodera, a następnie generowano opisy za pomocą modułu dekodera. Generowanie odbywało się w dwóch trybach prostym i zaawansowanym.

I. Część Teoretyczna

3.1 Mechanizm działania transformatorów w zadaniu Image-to-Text

Transformery to zaawansowana architektura sieci neuronowych, która zdobyła popularność dzięki swojej skuteczności w przetwarzaniu sekwencji danych, takich jak tekst i obrazy. W przypadku zadania Image-to-Text transformery wykorzystują mechanizm uwagi (ang. *attention*), który pozwala modelowi na kontekstowe przetwarzanie informacji.

Model BLIP (Bootstrapped Language-Image Pretraining), zastosowany w projekcie, składa się z dwóch głównych komponentów:

1. **Koder obrazu** - Przekształca dane wizualne na wektory reprezentacji. Koder ten opiera się na architekturze ViT (Vision Transformer), która dzieli obraz na mniejsze fragmenty (*patches*) i przetwarza je jako sekwencję danych wejściowych.
2. **Dekoder tekstu** - Generuje opisy w formie tekstowej, przekształcając wektory reprezentacji na sekwencję słów. Dekoder opiera się na warstwach transformera, które uwzględniają zarówno kontekst wizualny, jak i zależności między słowami w zdaniu.

3.2 Proces przetwarzania obrazu

Działanie modelu BLIP można opisać w trzech krokach:

1. **Wstępne przetwarzanie obrazu:** Model przyjmuje obraz w formacie RGB, który jest przeskalowany i normalizowany, aby spełniać wymagania kodera. Proces ten zapewnia, że dane wejściowe są spójne pod względem rozmiaru i jakości.
2. **Kodowanie obrazu:** Koder ViT przekształca obraz na reprezentację wektorową, która zawiera istotne cechy wizualne.
3. **Generowanie tekstu:** Na podstawie reprezentacji wektorowej dekodek tworzy tekst, który najlepiej opisuje zawartość obrazu.

3.3 Parametry i optymalizacja

Jakość generowanego tekstu zależy od parametrów takich jak:

- **Max_length:** Maksymalna długość generowanego opisu.
- **Num_beams:** Liczba ścieżek w algorytmie wyszukiwania wiązkowego (*beam search*).
- **No_repeat_ngram_size:** Minimalna liczba powtórzeń n-gramów w generowanym tekście.

Parametry te pozwalają kontrolować złożoność i szczegółowość opisów.

II. Część Praktyczna

4.1 Implementacja narzędzia

Zaprojektowano i zaimplementowano skrypt w Pythonie, który wykorzystuje model BLIP do generowania tekstowych opisów obrazów. Kluczowe etapy implementacji:

1. **Inicjalizacja modelu:** Skrypt łąduje model i procesor z portalu Hugging Face, umożliwiając lokalne przechowywanie danych w celu zoptymalizowania czasu wczytywania.
2. **Przetwarzanie obrazu:** Użytkownik może wprowadzić ścieżkę do obrazu lokalnego lub URL. Obraz jest przekształcany do formatu wymaganego przez model za pomocą biblioteki Pillow.
3. **Generowanie opisu:** W zależności od wybranego trybu (prosty lub zaawansowany) model generuje opis o różnym stopniu szczegółowości. Oba tryby różnią się zastosowanymi parametrami modelu, co wpływa na długość, szczegółowość oraz czas generowania opisów.

Parametry użyte dla modelu:

- **max_length:** Maksymalna długość generowanego opisu wynosi 100 znaków. Ogranicza to opis do krótkich, zwięzłych informacji.
- **min_length:** Minimalna długość opisu wynosi 10 znaków, co zapewnia wygenerowanie przynajmniej podstawowego opisu.
- **num_beams:** Algorytm wyszukiwania wiązkowego (*beam search*) analizuje 5 potencjalnych ścieżek w celu wygenerowania najlepszego opisu. To kompromis między szybkością działania a jakością wyników.
- **early_stopping:** Generowanie kończy się natychmiast, gdy model znajdzie satysfakcjonujący wynik. Dzięki temu czas generowania jest krótszy.
- **no_repeat_ngram_size:** Model unika powtarzania tych samych n-gramów (sekwencji trzech kolejnych słów), co poprawia spójność tekstu.

4.2 Testowanie i wyniki

Przeprowadzono testy na zbiorze obrazów reprezentujących różne scenariusze, takie jak:

- **Obrazy z prostymi elementami:** Obraz z psem na trawniku
- **Obrazy złożone:** Sceny zawierające wiele elementów, takiej jak Rejtan – Upadek Polski – obraz Jana Matejki z 1866.

Wyniki:

Porównanie Modeli

- Rozmiar i wydajność: Model Base jest lżejszy i szybszy, co sprawia, że jest bardziej praktyczny w zastosowaniach wymagających przetwarzania dużej liczby obrazów. Model Large oferuje wyższą jakość, ale wymaga większej mocy obliczeniowej.
- Jakość opisów: Model Large generuje bardziej szczegółowe i bogate opisy, co jest istotne w zastosowaniach wymagających szczegółowej analizy obrazów.
- Zastosowanie:
 - BLIP Base: Nadaje się do codziennych zastosowań, takich jak tworzenie opisów dla galerii obrazów czy prostych analiz wizualnych.
 - BLIP Large: Idealny do bardziej wymagających zadań, takich jak analizy naukowe, tworzenie szczegółowych raportów wizualnych czy analiza złożonych scen.

Testowe zdjęcie:



Salesforce/blip-image-captioning-base

```
● (myenv) kamildziedzic@MacBook-Pro-Kamil pjn % python3 projekt.py

Menu:
1. Generuj opis z URL obrazu (simple)
2. Generuj opis z URL obrazu (advanced)
3. Wyjdź
Wybierz opcję: 1
Wprowadź URL obrazu: /Users/kamildziedzic/Downloads/467186778_2013511619077349_1452096888790434597_n.png

Opis (simple): a dog sitting on top of a tree branch
Czas generowania: 2.17 sekund

Menu:
1. Generuj opis z URL obrazu (simple)
2. Generuj opis z URL obrazu (advanced)
3. Wyjdź
Wybierz opcję: 2
Wprowadź URL obrazu: /Users/kamildziedzic/Downloads/467186778_2013511619077349_1452096888790434597_n.png

Opis (advanced): a dog sitting on top of a tree branch in the middle of the image is a golden retriever sitting on a log in the dog is looking up at the camera
Czas generowania: 16.53 sekund
```

Salesforce/blip-image-captioning-large

```
Menu:
1. Generuj opis z URL obrazu (simple)
2. Generuj opis z URL obrazu (advanced)
3. Wyjdź
Wybierz opcję: 1
Wprowadź URL obrazu: /Users/kamildziedzic/Downloads/467186778_2013511619077349_1452096888790434597_n.png

Opis (simple): there is a dog that is sitting on a branch in the grass
Czas generowania: 3.35 sekund

Menu:
1. Generuj opis z URL obrazu (simple)
2. Generuj opis z URL obrazu (advanced)
3. Wyjdź
Wybierz opcję: 2
Wprowadź URL obrazu: /Users/kamildziedzic/Downloads/467186778_2013511619077349_1452096888790434597_n.png

Opis (advanced): there is a dog sitting on a branch in the middle of a field with trees in the background and a dog sitting on the branch in the foreground of the foreground
Czas generowania: 21.18 sekund
```

Porównanie modelu Salesforce/blip-image-captioning-base z różnymi parametrami:

Zdjęcie, na którym przeprowadzane zostały testy



Opis dla trybu zaawansowanego jest dokładniejszy jednak zajmuje dużo więcej czasu na przetworzenie i wygenerowanie tekstu. Generowanie opisu w trybie prostym zajęło 1.82 sekundy dając jednozdaniowy krótki opis a tryb zaawansowany zajął 24.09 sekundy i stworzył złożony opis obrazu.

```
(myenv) kamildziedzic@MacBook-Pro-Kamil pjn % python3 projekt.py
Menu:
1. Generuj opis z URL obrazu (simple)
2. Generuj opis z URL obrazu (advanced)
3. Wyjdź
Wybierz opcję: 1
Wprowadź URL obrazu: /Users/kamildziedzic/Downloads/JanMatejkoRejtan.jpg
Opis (simple): a painting of a group of people in a room
Czas generowania: 1.82 sekund

Menu:
1. Generuj opis z URL obrazu (simple)
2. Generuj opis z URL obrazu (advanced)
3. Wyjdź
Wybierz opcję: 2
Wprowadź URL obrazu: /Users/kamildziedzic/Downloads/JanMatejkoRejtan.jpg
Opis (advanced): a painting of a group of people sitting in front of a painting of a man in a white suit and a woman in a red dress, a woman in a white dress, and a man in a man in a black suit and a white dress
Czas generowania: 24.09 sekund
```

4.3 Analiza wydajności

Czas generowania opisów:

- Tryb prosty: Średni czas generowania wynosił 2.4 sekundy.
- Tryb zaawansowany: Średni czas generowania wynosił 22.4 sekundy.

4.4 Wyzwania i ograniczenia

Podczas testów zidentyfikowano kilka wyzwań:

- **Niska jakość obrazu:** Model miał trudności z generowaniem opisów dla obrazów o słabej jakości.
- **Niejednoznaczne sceny:** W przypadku obrazów zawierających wiele elementów generowane opisy czasami były zbyt ogólne.

- **Czas generowania w trybie zaawansowanym:** Wysoka liczba parametrów w trybie zaawansowanym zwiększa czas przetwarzania. Wymagana jest duża moc obliczeniowa by w racjonalnym czasie uzyskać wyniki.

Podsumowanie

Projekt umożliwił stworzenie narzędzia do automatycznego generowania opisów obrazów. Cel został osiągnięty, a narzędzie działa poprawnie w różnych scenariuszach. W przyszłości można rozważyć optymalizację modelu w celu skrócenia czasu generowania opisów lub poprawy ich jakości na bardziej wymagających danych.

Bibliografia

1. Hugging Face: Image-to-Text Task.

<https://huggingface.co/tasks/image-to-text>

2. Transformers Documentation.

<https://huggingface.co/docs/transformers>

3. Salesforce BLIP Model.

<https://huggingface.co/Salesforce/blip-image-captioning-base>

Każdy element raportu jest zgodny z wymaganiami i może być dostosowany do konkretnych potrzeb. Jeśli chcesz, mogę pomóc uzupełnić lub poprawić dowolny fragment!