

Politechnika Warszawska

W Y D Z I A Ł E L E K T R O N I K I
I T E C H N I K I N F O R M A C Y J N Y C H



Instytut Radioelektroniki i Technik Multimedialnych

Praca dyplomowa

na kierunku Studia Podyplomowe
w specjalności Głębokie Sieci Neuronowe - Zastosowania w Mediach Cyfrowych

Dostrajanie zapytania w problemie odpowiedzi na pytania na podstawie
informacji wizualnej.

Emilia Zawadzka-Gosk

Numer albumu 01183161

promotor
dr inż. Jacek Komorowski

WARSZAWA 2023

Dostrajanie zapytania w problemie odpowiedzi na pytania na podstawie informacji wizualnej.

Streszczenie.

Główym celem pracy jest stworzenie rozwiązania które umożliwi wskazywanie odpowiedzi na obrazie na pytanie zadane w języku naturalnym. Nacisk w pracy został położony na wykorzystanie istniejących modeli głębokich sieci neuronowych w trzyetapowym procesie rozwiązywania postawionego problemu oraz zastosowanie techniki wspierającej inferencję modelu języka jaką jest prompt tuning. W pierwszej części pracy przedstawiono istniejące rozwiązania, odpowiadające na poszczególne aspekty omawianego w pracy zagadnienia. Druga część pracy rozpoczyna się od opisu zaproponowanego w niniejszym projekcie rozwiązania, a także wyjaśnienia trzech etapów z których się ono składa, czyli detekcji obiektów, opisu wykrytych obszarów oraz klasyfikacji za pomocą modelu językowego. Eksperymenty opisane w dalszej części pracy przedstawiają wyniki klasyfikacji dokonanej przez model językowy na zbiorze walidacyjnym z zastosowaniem różnych podejść, oraz omówienie rezultatu działania całego rozwiązania. W zakończeniu pracy przedstawiono wnioski z wykonanego badania oraz dalsze działania, które mogą przynieść poprawę wyników.

Słowa kluczowe: multymodalny, detekcja obiektów, model języka, prompt-tuning



.....
miejscowość i data

.....
imię i nazwisko studenta

.....
numer albumu

.....
kierunek studiów

OŚWIADCZENIE

Świadomy/-a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie, pod opieką kierującego pracą dyplomową.

Jednocześnie oświadczam, że:

- niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym,
- niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
- niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanego z nadawaniem dyplomów lub tytułów zawodowych,
- wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
- znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.

Oświadczam, że treść pracy dyplomowej w wersji drukowanej, treść pracy dyplomowej zawartej na nośniku elektronicznym (płycie kompaktowej) oraz treść pracy dyplomowej w module APD systemu USOS są identyczne.

.....
czytelny podpis studenta

Spis treści

1. Wstęp	9
2. Cel pracy	10
3. Przegląd literatury	11
4. Opis rozwiązania	13
4.1. Opis zbioru danych	13
4.2. Ogólny opis rozwiązania	14
4.3. Przygotowanie danych do prompt-tuningu	15
5. Wyniki ewaluacji eksperimentalnej	19
6. Podsumowanie	22
Bibliografia	23
Spis rysunków	25
Spis tabel	25

1. Wstęp

Zadania które obejmują więcej niż jedną modalność są naturalne dla człowieka. Język naturalny oraz aspekty wizualne, czyli obrazy, często się przenikają i są używane przez ludzi do rozwiązywania różnych rodzajów problemów. Kombinacja języka naturalnego i wizji komputerowej staje się coraz bardziej powszechna, nie tylko w dziedzinie badań naukowych, ale także w rozwiązaniach komercyjnych. Modele, które integrują te dwie modalności, zawierają informacje o przestrzeniach semantycznych, które łączą obrazy i język. Charakteryzują się one głównie dużą skalą oraz dużą liczbą parametrów, co oznacza, że ich uczenie, a nawet dostrojenie, wymaga znacznych zasobów obliczeniowych i czasu.

Aby zmniejszyć wpływ tych ograniczeń, rozwijane są metody pozwalające na bardziej efektywne wykorzystanie dużych modeli przy ograniczonych zasobach obliczeniowych. W niniejszej pracy skoncentrowałem się na metodzie "prompt tuning" dla modelu języka opartego na architekturze Transformer. Bazując na wynikach inferencji modeli DETR i Transformer wizyjnego, pozwala ona na poprawę dokładności identyfikacji odpowiedzi na obrazie na pytanie zadane w języku naturalnym.

Inspiracją do realizacji tego projektu był konkurs Toloka Visual Question Answering Challenge (<https://toloka.ai/challenges/wsdm2023/>), którego dane zostały użyte w prezentowanym rozwiążaniu. Wykorzystując dane z konkursu oraz przy użyciu modeli DETR i ViT stworzony został zestaw danych treningowych i walidacyjnych. Tak przygotowane zbiory posłużyły do treningu "soft-promptu", oraz walidacji całego rozwiązania.

2. Cel pracy

Celem niniejszej pracy jest zaproponowanie rozwiązania, które wykorzystując istniejące duże modele językowe oraz ograniczone zasoby obliczeniowe pozwala na powiązanie opisu w języku naturalnym z obrazem oraz wspólne wnioskowanie na podstawie obu modalności, w postaci wskazania fragmentu obrazu zawierającego odpowiedź na pytanie zadane w języku naturalnym. Na przykład, jeśli zadane pytanie brzmi "What do we boil water in?"(pol. "W czym gotujemy wodę?"), należy wskazać fragment obrazu zawierający czajnik.

Projekt został zainspirowany konkurem Toloka Visual Question Answering Challenge (<https://toloka.ai/challenges/wsdm2023/>) ogłoszonym w 2022 roku przez firmę technologiczną Toloka. Problemem przedstawionym w wyzwaniu jest wskazanie odpowiedzi na obrazie na pytanie zadane w języku naturalnym. Organizator konkursu dostarczył treningowy zbiór danych zawierający:

- pytania,
- powiązane z nimi obrazy,
- współrzędne obiektów na obrazie, które wskazują poprawną odpowiedź na pytanie.

Dane dotyczą zagadnień ogólnych, nie wymagają posiadania wiedzy z żadnej specjalistycznej dziedziny. Wskazanie odpowiedzi na obrazie na zadane pytanie w większości przykładów ze zbioru jest zadaniem łatwym dla człowieka. Rysunek 2.1 przedstawia przykłady pytań, obrazów, oraz zaznaczonych odpowiedzi.



Pytanie: Which is different from the group?

Koordinaty: 409, 172, 432, 206



Pytanie: What do we drive for personal use?

Koordinaty: 161, 181, 569, 367



Pytanie: What do we drive for personal use?

Koordinaty: 161, 181, 569, 367



Pytanie: What do we drive for personal use?

Koordinaty: 161, 181, 569, 367

Rysunek 2.1. Przykłady pytań, obrazów z zaznaczoną odpowiedzią, oraz koordynaty odpowiedzi znajdujących się w zbiorze danych konkursu Toloka Visual Question Answering Challenge.

3. Przegląd literatury

Opracowany w niniejszej pracy problem składa się z kilku aspektów, takich jak detekcja obiektów, interpretacja zawartości obrazu i przedstawienie jej w języku naturalnym oraz zagadnień przetwarzania języka naturalnego, w tym rozumienia zadawanych pytań oraz szukania na nie odpowiedzi. Tę złożoną tematykę można rozpatrywać jako ciąg następujących po sobie podzadań. W niniejszej pracy zostało to opracowane w postaci ciągu operacji, w których udział biorą pretrenowane modele głębokich sieci neuronowych realizujące każde z zagadnień oddzielnie. Drugie podejście to zastosowanie jednego modelu sieci neuronowej, który tworzy przestrzeń powiązań pomiędzy zagadnieniami z dziedziny wizji komputerowej i przetwarzania języka naturalnego.

Odwołując się do pierwszego podejścia mówimy o trzech rodzajach rozwiązań sieci neuronowych realizujących:

- detekcję obiektu na obrazie
- zrozumienie pytania postawionego w języku naturalnym
- powiązanie dwóch powyższych zadań przestrzenią łączącą tekst i obraz.

Pierwszy ze wspomnianych kroków, czyli detekcja obiektów jest zagadniением, które może być zrealizowane za pomocą rozwijanych na przestrzeni ostatnich lat rozwiązań do

3. Przegląd literatury

których należą między innymi takie modele jak YOLO [1], RetinaNet [2], FasterR-CNN [3], czy zastosowany w projekcie DETR [4]. Rozwiązania te w różnią się między sobą zaproponowaną przez ich autorów architekturą oraz głównym zastosowaniem. Faster R-CNN, RetinaNet i YOLO korzystają z konwolucyjnych sieci neuronowych do ekstrakcji cech z obrazów. Przewidują i klasyfikują obszary, na których znajdują się obiekty. Zarówno RetinaNet jak i YOLO pozwalają wykrywać obiekty w różnych skalach oraz dobrze radzą sobie z niezbalansowanymi klasami. Zupełnie innym podejściem do problemu detekcji jest zastosowany w niniejszym projekcie dyplomowym DETR, który jest architekturą Transformer, stosowaną zwykle w problemach NLP, tutaj natomiast zastosowaną do obrazów.

Zagadnienia przetwarzania języka naturalnego rozwiązywane są obecnie przez duże modele językowe (LLM), które zazwyczaj oparte są o architekturę Transformer. W szczególności w wariantie dekodera, jak GPT [5] lub enkodera, jak BERT [6].

Modele multimodalne, w szczególności mówimy tutaj o modelach łączących język naturalny i obraz mogą mieć różne architektury oraz zastosowania. Na przykład model CLIP [7], tworzy wspólną przestrzeń w której osadzone są obrazy i tekst, co umożliwia wykorzystanie go w wielu rodzajach problemów powstających na styku wspomnianych dwóch modalności. Do grupy podobnych rozwiązań możemy zaliczyć Visual Transformer (ViT) [8], który wsparty przez model językowy jest stosowany na przykład do opisu obrazów. Odrębną grupą rozwiązań multimodalnych są modele realizujące jedno konkretne zadanie, np. generowanie obrazu na podstawie opisu tekstowego, jak DALL-E [9], czy StableDiffusion [10].

Lata 2012-2023 przyniosły kompleksowe i bardziej zaawansowane metody pracy z zagadnieniami powiązanych modalności. Warto tutaj wspomnieć na przykład o modelu GLIP [11], który łączy problem lokalizacji oraz rozumienia obrazu z językiem naturalnym, co umożliwia zastosowanie go w problemie detekcji obiektów, które opiszemy w języku naturalnym. Kolejnym rozwiązaniem skupiającym się na podobnym problemie jest ContextDET [12], rozwiązanie zaproponowane w 2023 roku realizujące zagadnienie detekcji obiektów na obrazie w kontekście opisanym językiem naturalnym. Zbliżonym do wymienionych wcześniej modelem jest GroundingDino [13], który potrafi dokonać detekcji obiektu którego nazwa została zawarta w podpowiedzi słownej (tzw. prompt). Rozwiązaniem najbliższej odpowiadającym potrzebie przedstawionej w projekcie dyplomowych jest DetGPT [14] opracowany w 2023 roku przez naukowców z The Hong Kong University of Science and Technology, który realizuje detekcję obiektów na podstawie pewnego kontekstu tekstowego, który niekoniecznie jest bezpośrednim opisem interesującego nas obiektu, ale opisuje jego własności w języku naturalnym.

4. Opis rozwiązania

4.1. Opis zbioru danych

Dane użyte w projekcie pochodzą z konkursu Toloka, ogłoszonego pod koniec 2022 roku. W ich skład wchodzi plik w formacie .csv oraz 38990 obrazów w formacie .jpg. Plik .csv zawiera takie informacje jak:

- pytanie w języku angielskim,
- link do powiązanego z pytaniem obrazu,
- informacje o szerokości i długości obrazu w pikselach,
- współrzędne obszaru na obrazie, w którym znajduje się odpowiedź na zadane pytanie.

Każdemu obrazowi ze zbioru danych odpowiada jeden rekord pliku .csv. Obrazy znajdujące się w zbiorze danych mają różne rozdzielczości, jednak nie przekraczają 640x640 pikseli. Są w formacie .jpg. Pytania, średnio składają się z 7,58 słowa, przy medianie 7 słów. Najdłuższe pytanie zawiera 34 słowa i brzmi ono "*What is the thing in vehicle which is designed to make sure that you are seen and stay noticed when it's most needed when turning or changing lanes in all light in weather conditions.*". Najkrótsze to dwuwyrazowe "*What meows?*". Słowo, w powyższym opisie, zostało zdefiniowane jako ciąg znaków znajdujący się pomiędzy białymi znakami. Przykład obrazu, powiązanego z nim pytania oraz zaznaczonej czerwonym kolorem odpowiedzi znajduje się na rysunku 4.1. Udostępniony w konkursie zbiór danych treningowych zawiera niecałe 40 tysięcy tego typu przykładów. Celem tak przygotowanego zestawu danych jest przedstawienie problemu wskazania odpowiedzi na obrazie na pytanie zadane w języku naturalnym. Odpowiedzią na pytanie jest obiekt znajdujący się na obrazie. W prezentowanym zbiorze danych wskazany jest w postaci współprzędnych skrajnego lewego górnego i prawego dolnego piksela.

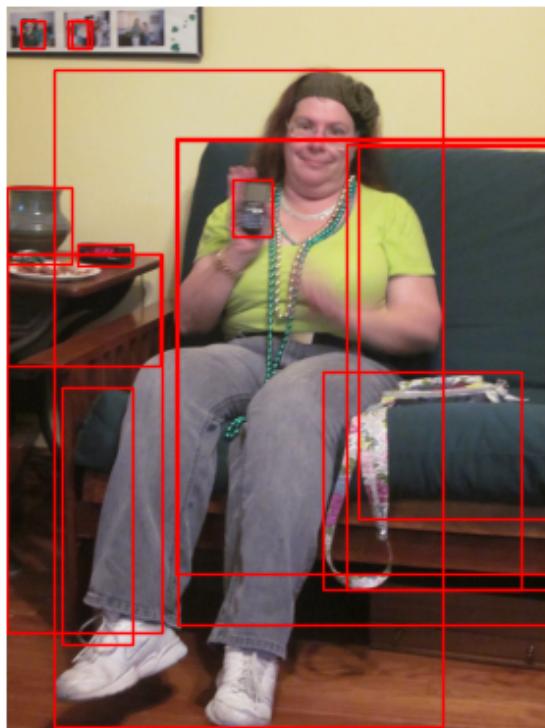


Rysunek 4.1. Przykład obrazu ze zbioru danych z zaznaczoną odpowiedzią na pytanie: "*What animal can bark?*"

4.2. Ogólny opis rozwiązania

Opisany we wstępie problem może być rozwiązywany na wiele sposobów z zastosowaniem głębokich sieci neuronowych. W niniejszym projekcie zdecydowano się na podejście kilkuetapowe:

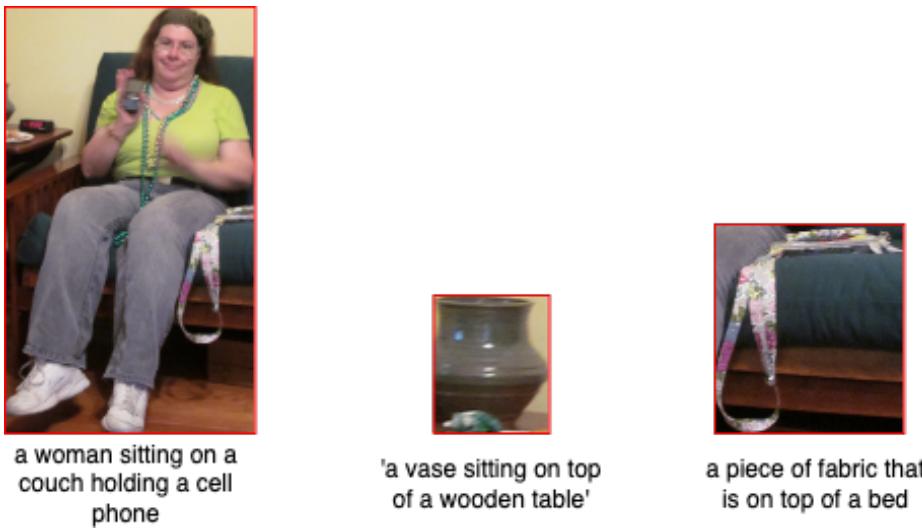
1. **Detekcja obiektów zrealizowana za pomocą modelu DETR [4].** W tym celu zastosowany został pretrenowany model DEtection TRansformer będący modelem wizyjnym, dostępnym w bibliotece Hugging Face. Model ten w procesie inferencji pozwala na wykrycie oraz sklasyfikowanie obiektów znajdujących się na obrazie. W pierwszym kroku dokonywana jest detekcja jak największej ilości znajdujących się na obrazie obiektów (w idealnym przypadku są to wszystkie obiekty znajdujące się na obrazie). W tym celu ustawiono wartość parametru "*threshold*" na 0.3. Próg ten informuje w skali od 0 do 1 ile model jest pewny zwróconego przez siebie wyniku. Dokonana zmiana wartości zwiększa ilość wykrytych obiektów, jednak może wpływać na trafność otrzymanego wyniku. Rysunek 4.2 przedstawia przykładowy wynik detekcji obiektów dokonanych na obrazie.



Rysunek 4.2. Przykład obrazu z obiektami wykrytymi za pomocą modelu DETR.

2. **Opis obszarów wykrytych w poprzednim kroku za pomocą modelu VisionEncoder-DecoderModel.** W kroku drugim również zastosowany został pretrenowany model z biblioteki Hugging Face. VisionEncoderDecoderModel jest rozwiązaniem w którym enkoderem jest model Vision Transformer (ViT) [8] a dekoderem GPT-2 [5]. Jest to model multimodalny, pozwalający na wykorzystanie go w zadaniu krótkiego opisu

obrazu w języku naturalnym. Każdy z wykrytych w pierwszym kroku obiektów, został opisany przez VisionEncoderDecoderModel. Wybrane przykłady opisów wykrytych obiektów przedstawione zostały na rysunku 4.3.



Rysunek 4.3. Przykłady opisów obiektów wygenerowane przez VisionEncoderDecoderModel.

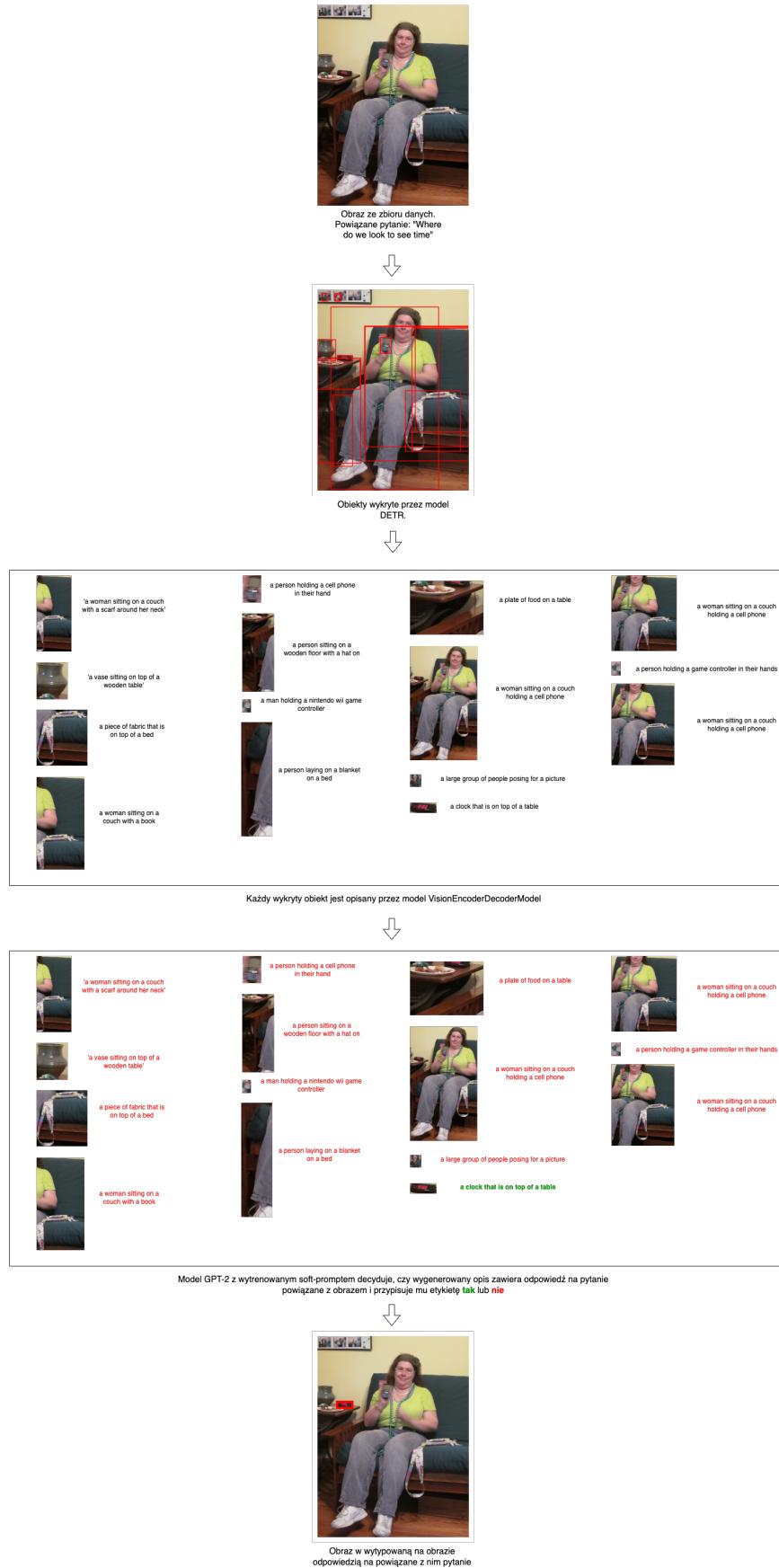
3. **Ocena przez model językowy GPT-2 [5]**, czy opis wygenerowany w kroku drugim zawiera odpowiedź na pytanie ze zbioru danych. W kroku trzecim użyty został pretrenowany model GPT-2 z biblioteki Hugging Face. Dodatkowo, w celu poprawy dokładności odpowiedzi modelu GPT-2 w kroku trzecim rozwiązania wykonany został prompt-tuning [15].

Rysunek 4.4 przedstawia przebieg całego opisanego powyżej procesu na przykładzie jednego obrazu.

4.3. Przygotowanie danych do prompt-tuningu

Modele językowe ogólnego przeznaczenia zwykle otrzymują od użytkownika zadanie w postaci odpowiednio sformułowanego promptu, czyli ciągu tekstowego, który opisuje zadanie, jakie model ma rozwiązać. Prompty formułowane są w języku naturalnym, zawierają polecenie i dane, które model ma przetworzyć. Aby uzyskać lepszą odpowiedź modelu do promptu można dodać przykład poprawnie rozwiązane zadania (one-shot prompting) lub kilka przykładów, jeśli zagadnienie jest bardziej skomplikowane (few-shot prompting). Ze względu na sposób działania sieci neuronowych, aby mogły one przetworzyć tekst w języku naturalnym, musi zostać on zamieniony na wartości liczbowe. Tekst zamieniany jest na ciąg tokenów za pomocą tokenizera. Następnie ciąg ten jest zamieniany na tensor osadzenia (embedding). Jest on następnie podawany na wejściu modelu językowego. Wykorzystując tę informację możliwe jest dołączenie do właściwego promptu dodatkowego tensora, tak zwanego soft-promptu. Aby soft-prompt poprawiał wyniki otrzymywane na wyjściu modelu językowego należy go wytrenować. W tym celu wykorzystujemy model ję-

4. Opis rozwiązania



Rysunek 4.4. Wizualizacja kroków detekcji odpowiedzi na obrazie na pytanie "*Where do we look to see time?*".

zykowy, którego parametry zamrażamy. W pętli trenowania na wejściu modelu językowego podawany jest właściwy prompt z dołączonym soft-promptem, a otrzymany na wyjściu modelu rezultat wykorzystywany jest do optymalizacji wartości soft-promptu metodą spadku wzdłuż gradientu. Parametry modelu językowego pozostają niezmienione.

Aby możliwe było przeprowadzenie trenowania soft-promptu w niniejszym projekcie niezbędne było trzyetapowe przygotowanie danych. W pierwszym etapie przygotowania danych dokonana została detekcja obiektów na każdym z obrazów ze zbioru danych. W tym celu użyty został model DETR. Próg (threshold) wartości predykcji detekcji został ustalony na 0.3, aby jak najwięcej potencjalnych obiektów zostało wskazane przez model. Dla każdego z wykrytych obszarów policzone zostało IoU porównujące go do współrzędnych obiektu w danych źródłowych. W przypadku, gdy wynik wyniósł ponad 70% rekord taki był oznaczany jako prawidłowa detekcja (label = 1). Prawidłowa detekcja wystąpiła tylko w około połowie obrazów ze zbioru danych. Wykonanie tej klasyfikacji jest istotne w kontekście późniejszego trenowania soft-promptu w trzecim etapie projektu.

W drugim etapie wykorzystany został model VisionEncoderDecoderModel, a konkretnie jego wersja vit-gpt2-image-captioning dostępna w bibliotece Huggingface. Ze względu na dużą ilość wykrytych obiektów w całym zbiorze danych oraz uwzględniając ograniczenia środowiska Google Colab, na którym wykonany został cały projekt, w niniejszej pracy przyjęto następującą strategię:

- W przypadku gdy w kroku liczenia IoU detekcja uzyskiwała wartość >70%, ten wycinek obrazu był opisywany przez model VisionEncoderDecoderModel
- Jeśli na danym obrazie nie dokonano żadnej prawidłowej detekcji, losowano jedną z detekcji nieprawidłowych (<70%) dla tego obrazu i opisywano ją za pomocą VisionEncoderDecoderModel.

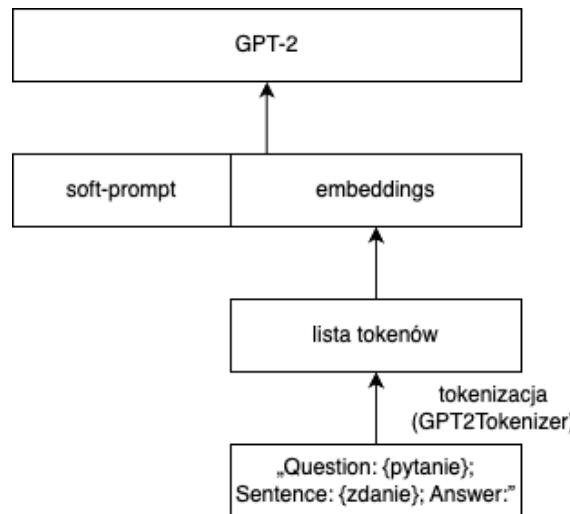
W ten sposób zbiór danych pozostał podobnej wielkości do danych surowych oraz zbalansowana była ilość przykładów dla obu klas (label = 1 i label = 0).

W trzecim etapie model językowy typu GPT2 miał za zadanie określić, czy w podanej mu sekwencji składającej się z pytania oraz zdania, w zdaniu znajduje się odpowiedź na pytanie, przy czym zdanie nie było bezpośrednią odpowiedzią na pytanie. Prawidłową odpowiedzią (ground truth) stała się klasa nadana na podstawie wartości IoU, czyli wspomniany wcześniej „label”, którego wartości na potrzeby zastosowanego modelu języka zostały zamienione odpowiednio: 1 na „yes”, oraz 0 na „no”. Prompt został skonstruowany w następujący sposób (Rys. 4.5):

1. Każdy element zbioru treningowego przedstawiony został w postaci: „*Question: pytanie; Sentence: zdanie; Answer:*”. pytanie pochodziło ze zbioru danych. zdanie było opisem fragmentu obrazu wygenerowanym przez VisionEncoderDecoderModel.
2. Stworzone w powyżej przedstawiony sposób prompt zostało zamieniony na listę tokenów za pomocą tokenizera GPT2Tokenizer, który stosuje algorytm Byte-Pair Encoding (BPE).

4. Opis rozwiązania

3. Na podstawie listy tokenów powstał wektor osadzenia (embedding).
4. Tensor embeddingu poprzedzony został soft-promptem w postaci zerowego tensora o rozmiarze 10x1024.



Rysunek 4.5. Schemat konstruowania promptu z użyciem soft-promptu.

W celu wytrenowania soft-promptu zastosowany został model GPT-2 w wersji LMHead, czyli z ostatnią warstwą przygotowaną do generowania kolejnego tokenu. W omawianym zadaniu obserwowany był jedynie ostatni wygenerowany token. Na czas treningu zamrożone zostały parametry modelu języka GPT-2, a wsteczna propagacja gradientu dotyczyła jedynie wartości tensora soft-prompt. Tak wytrenowany tensor wykorzystany został do walidacji rozwiązania.

5. Wyniki ewaluacji eksperymentalnej

Przygotowany wcześniej zbiór danych został podzielony na dwie części: dane treningowe oraz dane walidacyjne. Ze względu na trudności z utrzymaniem długiej sesji w środowisku Google Colab zbiór walidacyjny liczył 1000 próbek. Za pomocą danych treningowych dokonany został trening soft-promptu o rozmiarze 10x1024 (co jest odpowiednikiem 10 tokenów, każdy o długości 1024). W tym celu użyto modelu GPT-2 z biblioteki Huggingface. Wszystkie parametry modelu zostały zamrożone, natomiast trenowane były parametry soft-promptu. Cały trening trwał 3 epoki, stopa uczenia natomiast była stała i wynosiła 0.001. Zmiany straty oraz dokładności dla danych treningowych i walidacyjnych przedstawione zostały na rysunku 5.1. Długość treningu to około 3 godziny (każda epoka trwała około 1 godzinę).

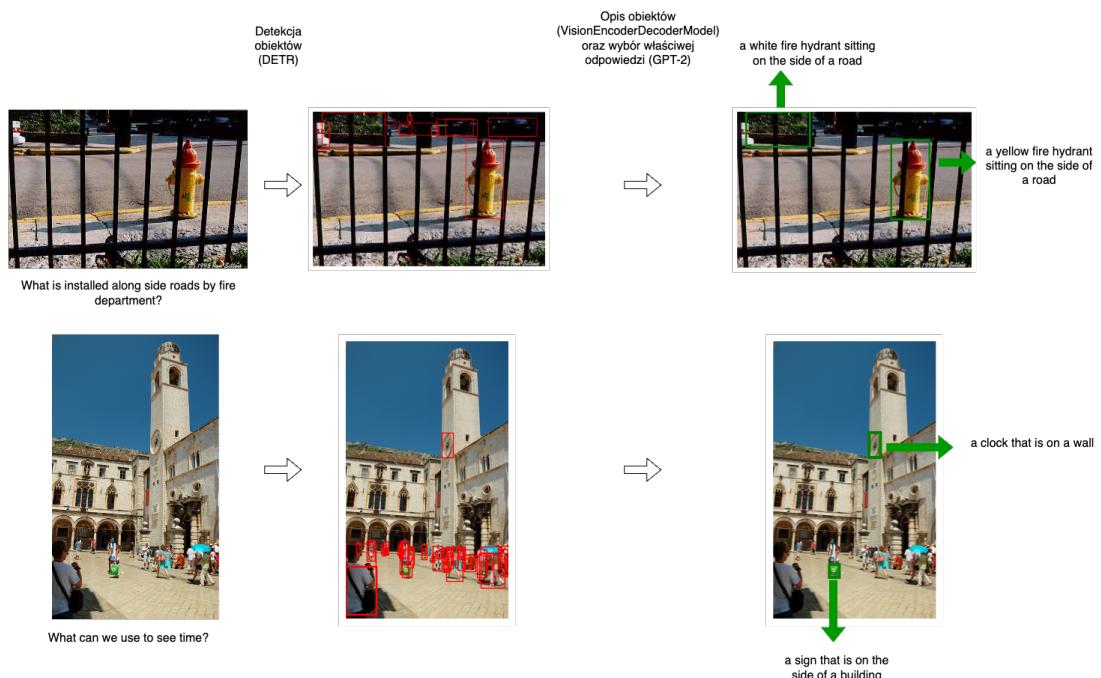


Rysunek 5.1. Wartości straty oraz dokładności w trakcie treningu soft-promptu.

Wybrane przykłady ilustrujące działanie prezentowanego rozwiązania znajdują się na rysunku 5.2

Stosując wytrenowany soft-prompt oraz ten sam podstawowy model GPT-2 osiągnięta została dokładność poprawnych odpowiedzi wygenerowanych przez model na poziomie 77% na zbiorze walidacyjnym. Jest to wartość wyraźnie wyższa, niż wykonując analogiczne inferencje na modelu, tylko z niezmiennym prefiksem (bez przykładów): „*Does the sentence contain the answer for the question?*”, w wyniku której otrzymujemy 0%. Odrobinę lepszy wynik, ale nadal niższy niż z wykorzystaniem zaproponowanej w pracy metody uzyskany został stosując metodę few-shot learning, która polega na dołączeniu

5. Wyniki ewaluacji eksperymentalnej



Rysunek 5.2. Wybrane przykłady detekcji odpowiedzi na zadane pytanie.

do promptu niezmiennej sekwencji, zawierającej przykłady, na podstawie których model może wnioskować. W przypadku tego eksperymentu, był to następujący tekst: „*Does the sentence contain the answer for the question? Question: What animal is fluffy and furry?; Sentence: the cat sleeps on a windowsill; Answer: yes Does the sentence contain the answer for the question? Question: What can be used to cut bread?; Sentence: a person on a bike; Answer: no Does the sentence contain the answer for the question? ..*”. Dokładność w tym przypadku to 14%. Gdy few-shot learning prompt został wydłużony do czterech przykładów, dokładność wzrosła do 18,7%. Najlepszy wynik został osiągnięty po dostrojeniu całego modelu (fine-tuning), wyniósł 81,9% na zbiorze walidacyjnym. W przypadku operacji fine-tuningu wystarczyła jedna epoka aby osiągnąć pożądany efekt, jednak ze względu na ilość parametrów dla których liczony był gradient, trwała ona około 3 godziny. Stopa uczenia była stała i wynosiła 0,0001. Opisane powyżej wyniki przedstawione zostały w tabeli 5.1. Dokładność została policzona z zastosowaniem metryki accuracy_score z biblioteki sklearn dla zbioru etykiet przewidzianych przez model oraz prawdziwych (wynikających z wartości IoU). Oznacza to, że:

1. w przypadku gdy obie etykiety miały tę samą wartość predykcja taka otrzymywała wartość 1, w przeciwnym wypadku wartość 0,
2. wartości z poprzedniego kroku zostały zsumowane
3. w celu normalizacji suma była podzielona przez ilość próbek.

Wyliczona została również uśredniona wartość IoU dla całego rozwiązania. Wyniosło ono 7,92, przy czym jeśli dla jednego obrazu dokonane zostało kilka detekcji, które zostały uznane za poprawne w trakcie walidacji, a w dane źródłowe wskazywały na tylko jedną

z nich, to wyliczane było ich średnie IoU, co wpływa negatywnie na ostateczny wynik ostateczny wynik.

Zastosowane rozwiązańe	Dokładność na zbiorze walidacyjnym
Prefix bez przykładów	0%
Prefix (2 przykłady)	14%
Prefix (4 przykłady)	18,7%
Dostrajanie zapytania (prompt-tuning)	77%
Dostrajanie modelu (fine-tuning)	81,9%

Tabela 5.1. Dokładność predykcji przy zastosowaniu różnych technik wspomagających model języka.

Powyższe zestawienie wskazuje, że najlepszy wynik został osiągnięty po dostrojeniu całego modelu (fine-tuning). Zaproponowany w projekcie prompt-tuning uzyskał dokładność na zbiorze walidacyjnym o 5% niższą, jednak liczba parametrów w przypadku trenowania soft-promptu jest o cztery rzędy wielkości niższa, co ma istotny wpływ na zapotrzebowanie obliczeniowe eksperymentu. Tabela 5.2 zawiera porównanie podstawowych informacji dotyczących wydajności obu procesów. Obydwa treningi odbywały się w środowisku Google Colab.

	soft-prompt tuning	fine-tuning
Liczba parametrów	10240	345M
Liczba epok	3	1
Czas trwania 1 epoki	1 godz.	3 godz.
Całkowity czas treningu	3 godz.	3 godz.

Tabela 5.2. Porównanie procesów dostrajania soft-promptu i całego modelu językowego GPT-2.

6. Podsumowanie

W niniejszej pracy udało się osiągnąć założony cel. Powstało rozwiążanie, które wykorzystując istniejące, łatwo dostępne pretrenowane modele głębokich sieci neuronowych oraz środowisko Google Colab pozwala na rozwiązywanie zagadnienia wskazywania odpowiedzi na obrazku na pytanie zadane w języku naturalnym. W trakcie pracy nad prezentowanym rozwiążaniem pojawiły się różnego rodzaju wnioski i wątpliwości, które byłyby warte eksploracji w dalszych etapach rozwoju projektu. Pierwszy z nich dotyczy wpływu poszczególnych komponentów na całość. Ze względu na fakt, że w zaproponowanym rozwiążaniu stosowane są trzy modele sieci neuronowych, każdy z nich ma wpływ na ostateczny wynik. Jeśli komponent, odpowiedzialny za wykonanie detekcji obiektów, nie wykryje interesującego nas przedmiotu na obrazie, pozostałe komponenty nie są w stanie skompensować tej straty. Zmiana sieci, którą obecnie jest DETR, na inną, również dokonującą detekcji obiektów, być może mogłaby odrobinę poprawić wynik końcowy. Odpowiednio, jeśli opis fragmentu obrazu wykonany przez komponent drugi, którym jest model VisionEncoderDecoderModel, jest niedokładny lub występują w nim halucynacje wpływa to na efekt oceny przez komponent trzeci. Kolejną kwestią, nad którą należałyby się dłużej pochylić, jest wpływ długości soft promptu na wynik klasyfikacji modelu języka. Wnioskując na podstawie wyników zawartych w tabeli 5.1, a szczególnie przyglądając się prefiksom w przykładach typu „few-shot”, możemy spodziewać się, że podobny efekt może być dostrzeżony w przypadku promptów trenowań. Następnym aspektem wartym rozważenia jest ocena wyniku ostatecznego rozwiązania, szczególnie należąby się tutaj przyjrzeć przypadkom, w których na obrazie znajduje się więcej niż jeden przedmiot, który mógłby być odpowiedzią na zadane pytanie, natomiast dane źródłowe wskazują na tylko jeden z nich. Na końcu warto wspomnieć również o ciekawostce. W opisanym w pracy eksperymencie soft-prompt był tensorem o określonej długości, jednak, ponieważ stanowił on część zapytania, które otrzymuje model języka, spodziewane jest, że ma interpretację w języku naturalnym. Interpretacja ta jednak okazała się daleka od oczekiwania (spodziewano się ciągu tokenów zblizonego do prefiksu opisanego w rozdziale 5.). Po znalezieniu najbliższych tensorów w przestrzeni wszystkich tokenów tokenizera i zamianie ich na postać tekstową otrzymano: „Newman obadeersadeaterart theater”.

Bibliografia

- [1] J. Redmon, S. Divvala, R. Girshick i A. Farhadi, „You Only Look Once: Unified, Real-Time Object Detection”, w *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, czer. 2016. DOI: 10.1109/cvpr.2016.91. adr.: <https://doi.org/10.1109%2Fcvpr.2016.91>.
- [2] T.-Y. Lin, P. Goyal, R. Girshick, K. He i P. Dollar, „Focal Loss for Dense Object Detection”, w *2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, paź. 2017. DOI: 10.1109/iccv.2017.324. adr.: <https://doi.org/10.1109%2Ficcv.2017.324>.
- [3] S. Ren, K. He, R. Girshick i J. Sun, „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, t. 39, nr. 6, s. 1137–1149, czer. 2017. DOI: 10.1109/tpami.2016.2577031. adr.: <https://doi.org/10.1109%2Ftpami.2016.2577031>.
- [4] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov i S. Zagoruyko, „End-to-end object detection with transformers”, w *European conference on computer vision*, Springer, 2020, s. 213–229.
- [5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever i in., „Language models are unsupervised multitask learners”, *OpenAI blog*, t. 1, nr. 8, s. 9, 2019.
- [6] J. Devlin, M.-W. Chang, K. Lee i K. Toutanova, „Bert: Pre-training of deep bidirectional transformers for language understanding”, *arXiv preprint arXiv:1810.04805*, 2018.
- [7] A. Radford, J. W. Kim, C. Hallacy i in., „Learning transferable visual models from natural language supervision”, w *International conference on machine learning*, PMLR, 2021, s. 8748–8763.
- [8] A. Dosovitskiy, L. Beyer, A. Kolesnikov i in., „An image is worth 16x16 words: Transformers for image recognition at scale”, *arXiv preprint arXiv:2010.11929*, 2020.
- [9] A. Ramesh, M. Pavlov, G. Goh i in., „Zero-shot text-to-image generation”, w *International Conference on Machine Learning*, PMLR, 2021, s. 8821–8831.
- [10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser i B. Ommer, „High-resolution image synthesis with latent diffusion models”, w *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, s. 10 684–10 695.
- [11] L. H. Li, P. Zhang, H. Zhang i in., „Grounded language-image pre-training”, w *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, s. 10 965–10 975.
- [12] Y. Zang, W. Li, J. Han, K. Zhou i C. C. Loy, „Contextual Object Detection with Multi-modal Large Language Models”, *arXiv preprint arXiv:2305.18279*, 2023.
- [13] S. Liu, Z. Zeng, T. Ren i in., „Grounding dino: Marrying dino with grounded pre-training for open-set object detection”, *arXiv preprint arXiv:2303.05499*, 2023.
- [14] R. Pi, J. Gao, S. Diao i in., „DetGPT: Detect What You Need via Reasoning”, *arXiv preprint arXiv:2305.14167*, 2023.

6. Bibliografia

- [15] B. Lester, R. Al-Rfou i N. Constant, „The power of scale for parameter-efficient prompt tuning”, *arXiv preprint arXiv:2104.08691*, 2021.

Spis rysunków

2.1 Przykłady pytań, obrazów z zaznaczoną odpowiedzią, oraz koordynaty odpowiedzi znajdujących się w zbiorze danych konkursu Toloka Visual Question Answering Challenge.	11
4.1 Przykład obrazu ze zbioru danych z zaznaczoną odpowiedzią na pytanie: "What animal can bark?"	13
4.2 Przykład obrazu z obiektami wykrytymi za pomocą modelu DETR.	14
4.3 Przykłady opisów obiektów wygenerowane przez VisionEncoderDecoderModel.	15
4.4 Wizualizacja kroków detekcji odpowiedzi na obrazie na pytanie "Where do we look to see time".	16
4.5 Schemat konstruowania promptu z użyciem soft-promptu.	18
5.1 Wartości straty oraz dokładności w trakcie treningu soft-promptu.	19
5.2 Wybrane przykłady detekcji odpowiedzi na zadane pytanie.	20

Spis tabel

5.1 Dokładność predykcji przy zastosowaniu różnych technik wspomagających model języka.	21
5.2 Porównanie procesów dostrajania soft-promptu i całego modelu językowego GPT-2.	21