POLITECHNIKA WROCŁAWSKA

Wydział Podstawowych Problemów Techniki

Praca inżynierska

Rozwój mobilnej aplikacji do rozpoznawania języka migowego z wykorzystaniem transformerów

Development of mobile application for sign language recognition using transformers

Autor: Maciej Grzesik asda asds

Opiekun: Prof. uczelni dr hab. Sebastian Kraszewski

Wrocław 2024

Moim ...

Spis treści

Rozdział 1

Wprowadzenie

1.1 Motywacja

Wykluczenie społeczne osób z niepełnosprawnościami stanowi poważny problem w społeczeństwie. Osoby z uszkodzeniem słuchu, szczególnie w życiu codziennym, doświadczają trudności w komunikacji z otoczeniem, które opiera się na konwencjonalnej mowie. Język migowy, będący podstawową formą komunikacji dla osób głuchoniemych, nadal jest słabo znany i nauczany wsród osób zdrowych, co pogłębia te trudności. Na podstawie badań Pauliny Malczewskiej wynika, że aż 90% ankietowanych osób głuchych doświadcza alienacji oraz dyskryminacji ze strony osób słyszących [1].
Bariery komunikacyjne między osobami głuchoniemymi a tymi, którzy nie znają języka migowego, stanowią znaczący problem społeczny i medyczny. Przyczynia się to do izolacji społecznej, trudności w dostępie do usług zdrowotnych i ogólnie niszej jakości życia.
Istotnym jest zaadresowanie tego problemu za pomocą aplikacji wspierającej osoby głuchonieme, nie tylko ułatwi to komunikację, ale także przyczyni się do zwiększenia inkluzywności społecznej, zapewniając równe szanse i redukując poziom dyskryminacji.

1.2 Cel

Celem pracy jest stworzenie aplikacji mobilnej, która umożliwia tłumaczenie języka migowego na tekst w czasie rzeczywistym. Funkcjonalność aplikacji oparta jest na transformerach, czyli architekturze głębokiego uczenia maszynowego. Dzięki zastosowaniu modeli uczenia maszynowego możliwe będzie klasyfikowanie, a tym samym tłumaczenie języka migowego co ułatwi komunikację osobom głuchym. Aplikacja zbudowana jest w oparciu o środowisko Flutter, co pozwala na jej funkcjonowanie na różnych platformach mobilnych (Android i iOS).

1.3 Zakres

W ramach pracy inżynierskiej zaprojektowany zostanie przyjazny interfejs użytkownika, który umożliwi proste korzystanie z aplikacji zarówno przez osoby głuche, jak i słyszące. Wykorzystanie środowiska Flutter pozwoli na implementacje interaktywnego interfejsu użytkownika przy wykorzystaniu wbudowanych funkcjonalności. Dodatkowo zostanie wyeliminowana potrzeba pisania kodu w natywnym dla danego środowiska języku, co zapewnia aplikacji możliwość działania na różnych mobilnych systemach operacyjnych.

Zostanie również przeprowadzona implementacja oraz trening modelu. W tym etapie projektu zostaną zastosowane odpowiednie algorytmy, które umożliwią skuteczne uczenie modelu na podstawie zebranych danych dotyczących gestów języka migowego. Proces uczenia będzie obejmował zarówno fazę wstępną, w której model będzie dostosowywany do charakterystyki danych, jak i fazę walidacji, w której oceni się jego dokładność i zdolność do generalizacji gestów języka migowego. Istotynm jest również przeprowadzenie testów funkcjonalności modelu w warunkach rzeczywistych aby potwierdzić jego poprawne działanie lub wprowadzanie ewentualnych poprawek.

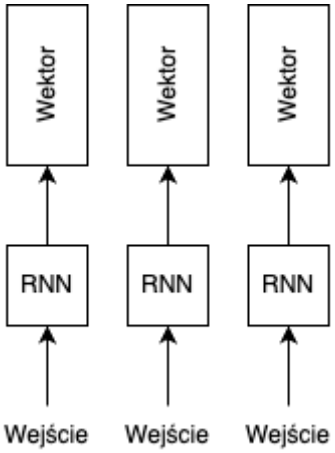
Rozdział 2

Teoria

Transformery to zaawansowane modele uczenia maszynowego, szeroko stosowane w przetwarzaniu języka naturalnego, wizji komputerowej, ale także w audio i przetwarzaniu multimedialnym.
W odróżnieniu od poprzednio stosowanych modeli uczenia maszynowego opierających się na podejściu rekurencyjnym i mechanizmie uwagi, naukowcy pracujący dla firmy Google zaproponowali rozwiązanie wykorzystujące jedynie mechanizm uwagi. Skutkiem tego podejścia jest znaczne przyspieszenie czasu uczenia modelu przy zachowaniu wysokiej precyzji [2].

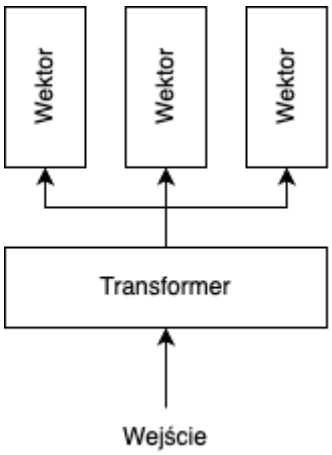
2.1 Zasada działania

Działanie transformera opiera się na warstwie kodującej oraz na warstwie dekodującej. Zadaniem warstwy kodującej jest zakodowanie argumentów wejściowych (np. tekstu, obrazów) do postaci numerycznej. Warstwa dekodująca, wykorzystuje przekazane z warstwy kodującej zakodowane informacje do przetworzenia ich na wartości wyjściowe (np. do tekstu czy wideo).
Istotnym elementem, który różnicuje transformery od rekurencyjnych modeli uczenia maszynowego jest mechanizm uwagi. W tradycyjnych modelach rekurencyjnych sieci neuronowych (RNN) dane przetwarzane są sekwencyjnie tzn., że są analizowane krok po kroku (patrz 2.1) [3] co skutkuje ograniczeniami m.in. w równoległości przetwarzaniu danych.



Rysunek 2.1: Przepływ informacji wejściowych w rekurencyjnych sieciach neuronowych

Rozwiązania inżynieryjne wykorzystane w transformerach pozwalają modelom na przetwarzanie wszystkich elementów sekwencji jednocześnie (patrz 2.2).



Rysunek 2.2: Przepływ informacji wejściowych w transformerach

W praktyce oznacza to, że uczenie transformerów przebiega w określonej stałej ilości $O(1)$ sekwencyjnie wykonywanych operacji, modele RNN wymagają natomiast $O(n)$ sekwencyjnie wykonywanych operacji co sprawia, że są mniej wydajne przy dłuższych sekwencjach.

2.2 Architektura

Zanim dane przetwarzane przez transformer zostaną przekazane do kodera muszą zostać wstępnie przetworzone. Proces ten można podzielić na dwa kroki:

- Input Embedding polega na zamienieniu danych wejściowych na przystępne dla obliczeń komputerowych wektory.
- Positional Encoding składa się z wektorów, których zadaniem jest nadanie kontekstu na podstawie pozycji danych w sekwencji. Jest to istotne, gdyż z punktu widzenia maszyny nie jest wstanie odróżnić ona informacji użytych w różnym znaczeniu.

Sam koder składa się z dwóch elementów:

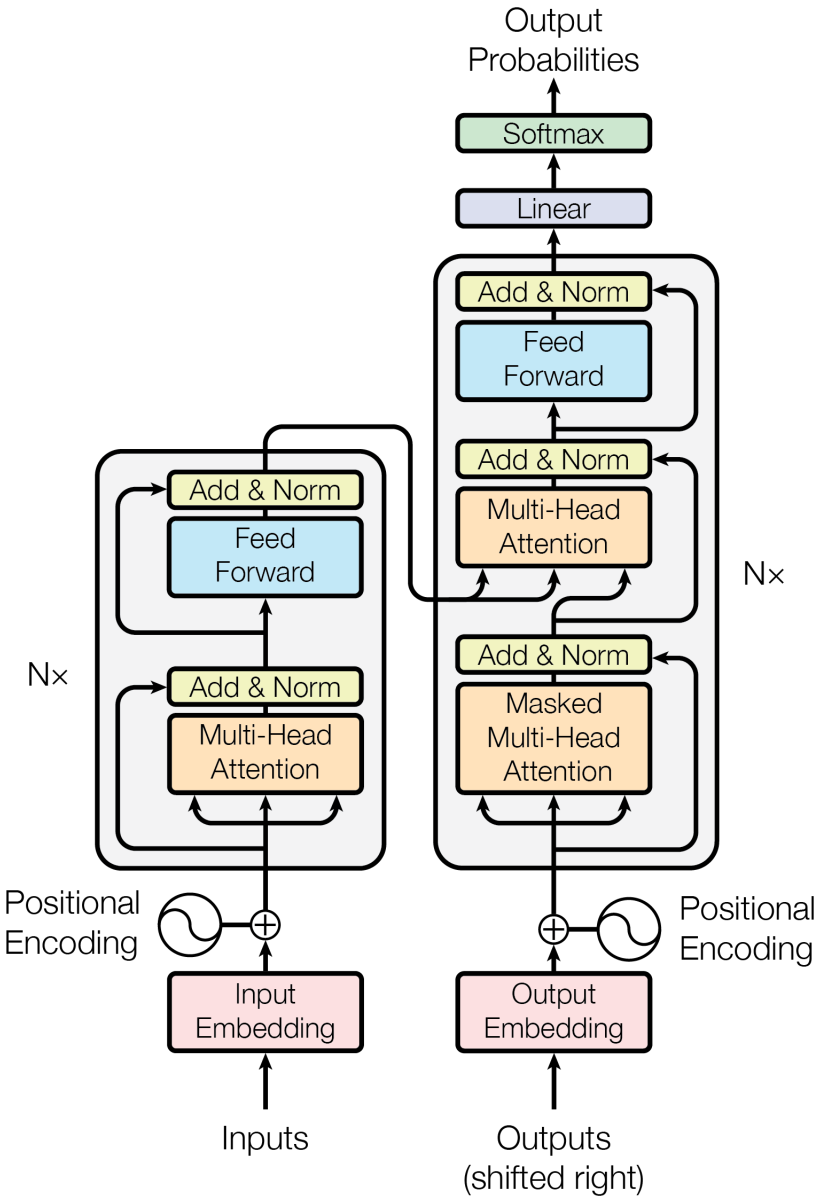
- Multi-Head Attention polega na obliczeniu wektora uwagi dla poszczególnych informacji przekazanych do tej warstwy. W tym etapie obliczany jest tzw. wektor uwagi za pomocą którego określa się wagę poszczególnych informacji. W przypadku przetwarzania wideo identyfikowane są kluczowe momenty takie jak m.in. gesty poprzez nadanie wag poszczególnym klatkom wideo w zależności od ich znaczenia. Ignorowane są mniej istotne informacje lub szum.
- Feed Forward, w tym etapie obliczone wektory przekazywane są do sieci MLP (Multi Layer Perceptron), która używana jest dla każdego wektora uwagi.

Tak przygotowane dane zostają przekazane do dekodera, którego zadaniem jest przewidywanie kolejnych słów czy obrazów. W tym procesie udział biorą poszczególne bloki:

- Embedding proces ten przebiega dokładnie tak samo jak w przypadku warstwy kodującej. Dane zamieniane są na wartości numeryczne w postaci macierzy.
- Positional Encoding również odbywa się w taki sam sposób jak w przypadku warstwy kodującej. Obliczane są wektory nadające kontekst informacjom.
- Masked Multi-Head Attention polega na zamaskowaniu, czyli przemnożeniu przez macierz zer.
- Multi-Head Attention with encoder łączy wektor wyjściowy z warstwy enkodującej z wektorem z poprzedniego kroku ze sobą. Podczas tego etapu sprawdzane jest w jakim stopniu każdy wektor jest ze sobą powiązany.

- Feed Forward jest to sieć, której zadaniem jest uproszczenie tłumaczenia wektora, aby łatwiej można było przerobić transformerowi wyniki parowań. Następnie w etapie linear layer przekształcane są wyniki mające na ten sam wymiar co dane wejściowe. Następnie przy użyciu funkcji softmax zmieniane są wyniki prawdopodobieństwa.

Poszczególne bloki architektury wraz z przepływem informacji w modelu zostały zaprezentowane na schemacie 2.3 [2].



Rysunek 2.3: Schemat blokowy architektury transformera

Bibliografia

[1] P. Malczewska. Izolacja społeczna osób z uszkodzonym słuchem jako wspólny obszar badań pedagogiki i antropologii. *Pedagogika a etnologia i antropologia kulturowa. Wspólne obszary badań*, page 128, 2011. 7

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 9, 12

[3] M. Mamczur. Czym jest i jak działa transformer sieć neuronowa?, March 2020. Dostęp: 23 maj 2024. 9