Optymalną architekturę dla moduły LSTM wybraliśmy porównując wartości MSELoss dla zbioru walidacyjnego. Najniższa wartość (uzyskana w dowolnym epochu) wskazywała na najlepszy model. Zmian dokonywaliśmy w parametrach hidden\_size i num\_layers LSTM z biblioteki pytorch oraz długość ciągu wejściowego (ilość liter/długość tokenu).

Ostatecznym wynikiem była poniższa klasa:Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

Gdzie:  
hidden\_size = 128  
num\_layers = 1  
a długość tokenu wynosiła = 100

Model trenowany był wg algorytmu:

Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

Opis wygenerowany automatycznie Zapobiegliśmy przeuczeniu poprzez zachowanie jedynie modelu o najniższym mean square error (MSE) i zatrzymanie uczenie jeśli przez 10 kolejnych epok MSE nie spadło.

Intuicyjnie wiemy, że model jest bardzo słaby. Użycie MSE pozwala przedstawić wysoki wynik, ponieważ nie traktuje wyniku binarnie a pozwala zaliczyć wyniki zbliżone jako sukcesy i pozwala na swobodę w obsłudze wartości ujemnych i niecałkowitych. Najniższy osiągnięty MSE to 11,2902.  
Dla przykładu fragment z „Pana Tadeusza”, gdzie należy dokończyć imię Napoleona we fragmencie „wojna niechybna! kiedy z poselstwem tajemnem tu biegłem, wojsk forpoczty już stały nad niemnem; nap” otrzymujemy „skkskssksssksssksssksskss”. Dla wejścia z poza „Pana Tadeusza” "mała mi miała mały młyn i małą miskę i małego misia i małą myszkę i małą makietę i małą markuję i ma" gdzie można spodziewać się kolejnej małej rzeczy otrzymujemy „sskssksskssksskssksskssks”

Dokładność modelu na zbiorze testowym to 0,461021747.

Optymalna architektura TransformerDecoder została wybrana porównując wartości MSELoss dla zbioru walidacyjnego. Najniższa wartość (uzyskana w dowolnym epochu) wskazywała na najlepszy model. Zmienialiśmy sugerowany parametr hidden\_size, num\_layers i num\_heads.

Ostatecznym wynikiem była poniższa klasa: Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Gdzie:  
hidden\_size = 128  
num\_layers = 1  
heads = 1

Model trenowany był zgodnie z algorytmem: Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

Opis wygenerowany automatycznie

Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

Opis wygenerowany automatycznie

Zapobiegliśmy przeuczeniu poprzez zachowanie jedynie modelu o najniższym mean square error (MSE).

Dla przykładu fragment z „Pana Tadeusza”, gdzie należy dokończyć imię Napoleona we fragmencie „tu biegłem, wojsk forpoczty już stały nad niemnem; nap” daje wynik „wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww”. Dla wejścia z poza „Pana Tadeusza” „mała mi miała mały młyn i małego misia i małą miskę i „ gdzie można spodziewać się kolejnej małej rzeczy otrzymujemy „dddddddddddddddddddddddddddddddddddddddddddddddddddddd”.

Dokładność na zbiorze testowym to 0,060354256465517245

Żaden z utworzonych przez nas modeli nie jest użyteczny.