Wprowadzenie do Przetwarzania Języka Naturalnego

Sprawozdanie z etapu pierwszego

Temat projektu: Przewidywanie oceny na podstawie opinii tekstowej

Mateusz Krakowski

Bartosz Latosek

Mikołaj Bańkowski

24 kwietnia 2024

Spis treści

1 Opis Projektu

1.1 Cel projektu

Celem projektu jest stworzeniu modelu klasyfikacji, który będzie w stanie automatycznie przypisywać liczbę gwiazdek hotelowi, na podstawie analizy recenzji, jaką hotel otrzymał. W tym celu wykorzystamy techniki przetwarzania języka naturalnego (Natural Language Processing) oraz uczenia maszynowego (Machine Learning).

1.2 Założenia projektowe

- Procesowi przetwarzania języka naturalnego, będą poddawane recenzje napisane w języku angielskim,
- Dane mogą pochodzić z różnych platform recenzenckich i zasobów online,
- Testowaniu zostaną poddane trzy różne modele oraz model naiwny, uzyskane wyniki zostaną porównane,
- Projekt realizowany w języku Python

1.3 Kamienie milowe

Na potrzebę projektu niezbędne jest, aby wykonać następujące zadania:

- Zgromadzenie danych zawierających recenzje hotelowe wraz z przypisanymi do nich ocenami w postaci liczby gwiazdek,
- Przygotowanie zbioru danych poprzez poddanie ich procesowi oczyszczania i przetwarzania wstępnego,
- Wybór i implementacja narzędzi lub algorytmów do przetwarzania języka naturalnego,
- Budowa modelu klasyfikacji, który zostanie wytrenowany na podstawie przetworzonych danych,
- Testowanie i ocena modelu za pomocą metryk takich jak dokładność (accuracy), precyzja (precision), czułość (recall) oraz F1-score,
- Transfer wiedzy - trening i ewaluacja modelu uczonego na recenzjach hoteli na zbiorze danych recenzji filmowych.

1.4 Zbiory Danych

Na potrzeby projektu przeszukaliśmy publicznie dostępne zbiory danych zawierające recenzje hoteli. Zbiór danych - Hotel Reviews [1] pochodzi z zasobów online dostępnych na platformie Kaggle. W zbiorze danym znajduję się około 50 milionów recenzji, natomiast na potrzeby projektu, będziemy bazować na 125 tys. recenzji. Pojedyncza recenzja składa się z takich pól jak:

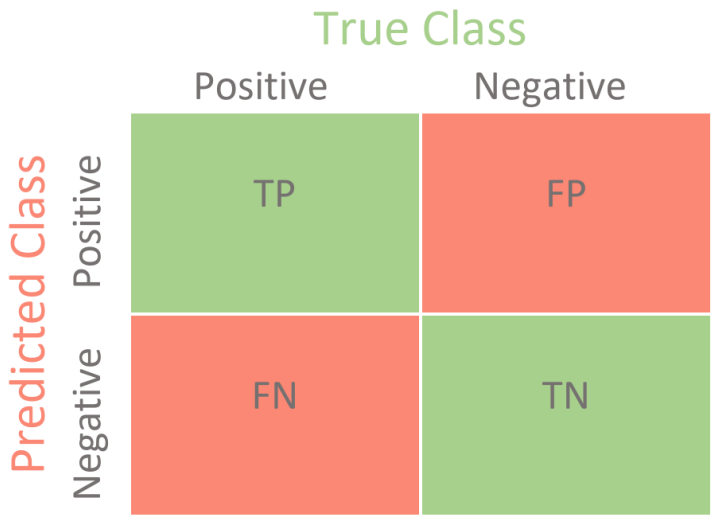
- hotel_url - adres URL recenzji hotelowej, który wskazuje na konkretną stronę internetową lub platformę, na której opublikowana została recenzja,
- author - nazwa autora recenzji, czyli osoby, która ją napisała,
- date - data publikacji recenzji,
- rating - ocena przyznana hotelowi przez autora recenzji w skali 1-5,
- title - tytuł recenzji, który zawiera krótkie podsumowanie lub opinię na temat hotelu,
- text - treść recenzji, która zawiera szczegółowe opinie i doświadczenia autora związane z pobytem w hotelu,
- property_dict - słownik zawierający oceny poszczególnych właściwości hotelu,

Kluczowymi polami dla nas będzie pole 'rating' zawierające ocenę numeryczną oraz pole 'text' zawierające ocenę pisemną hotelu.

1.5 Miary jakości modelu

1.5.1 Macierz pomyłek

Macierz zawierająca 4 wartości mówiące o tym jak model poradził sobie z klasyfikacją danych



Rysunek 1: Macierz Pomyłek

Na podstawie macierzy pomyłek obliczane poniższe miary jakości.

1.5.2 Accuracy

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

Accuracy niesie informację o tym, jaki procent próbek testowych został poprawnie sklasyfikowany przez model.

1.5.3 Recall(Sensitivity)

$$Recall = Sensitivity = \frac{TP}{TP + FN}$$

Recall mówi nam, jak model radzi sobie z klasyfikowaniem przypadków pozytywnych danej klasy.

1.5.4 Specificity

$$Specificity = \frac{TN}{TN + FP}$$

Specificity mówi nam, jak model radzi sobie z klasyfikowaniem przypadków negatywnych danej klasy.

1.5.5 Precision

$$Precision = \frac{TP}{TP + FP}$$

Precision mówi nam, w jakich proporcjach model klasyfikuje próbki jako pozytywne w zależności od faktycznej klasy próbki.

1.5.6 Miara F1

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

F1 Score mówi nam, jak dobrze model radzi sobie z klasyfikacją przypadków TP, przy tym minimalizując liczbę przypadków FP i FN.

1.5.7 Support

$$Support = TP + FN$$

Support mówi nam ile w danych ewaluacyjnych jest przypadków danej klasy.

1.6 Założenia odnośnie metryk oceny

Accuracy mimo że mówi nam o celności algorytmu, nie niesie za sobą wystarczającej informacji o jakości modelu. Ważne będzie, aby przy ocenie wziąć pod uwagę Recall oraz Specificity. To, aby ustalić która z tych dwóch metryk jest ważniejsza, musi odpowiedzieć pytanie czy jesteśmy bardziej skłonni dopuścić do klasyfikacji przypadków fałszywie negatywnych, czy fałszywie pozytywnych. Dodatkowo dobrą miarą do porównania modeli będzie miara F1.

2 Implementacja

2.1 Preprocessing - Przetwarzanie Danych

Z uwagi na to że oryginalny zbiór[1] złożony jest z 50 milionów próbek, należało go pomniejszyć. z 50 milionów wybrano 1 milion w losowy sposób wybierając z całego zbioru. Powinno to zapewnić, że zbiór danych może w dobry sposób reprezentować całą populację recenzji. Dostarczony skrypt Python `src/data/make_dataset.py` wykonuje następujące kroki preprocessingu na surowych danych:

1. **Odczyt Danych**: Kod odczytuje surowe dane z pliku JSON określonego jako ścieżka wejściowa.

2. **Wyciąganie Tekstu i Oceny**: Definiuje funkcję `extract_text_rating()`, aby wyciągnąć tekst i ocenę z danych JSON. Wewnątrz tej funkcji dane JSON są wczytywane, a następnie wyciągane są pola 'text' i 'rating'.

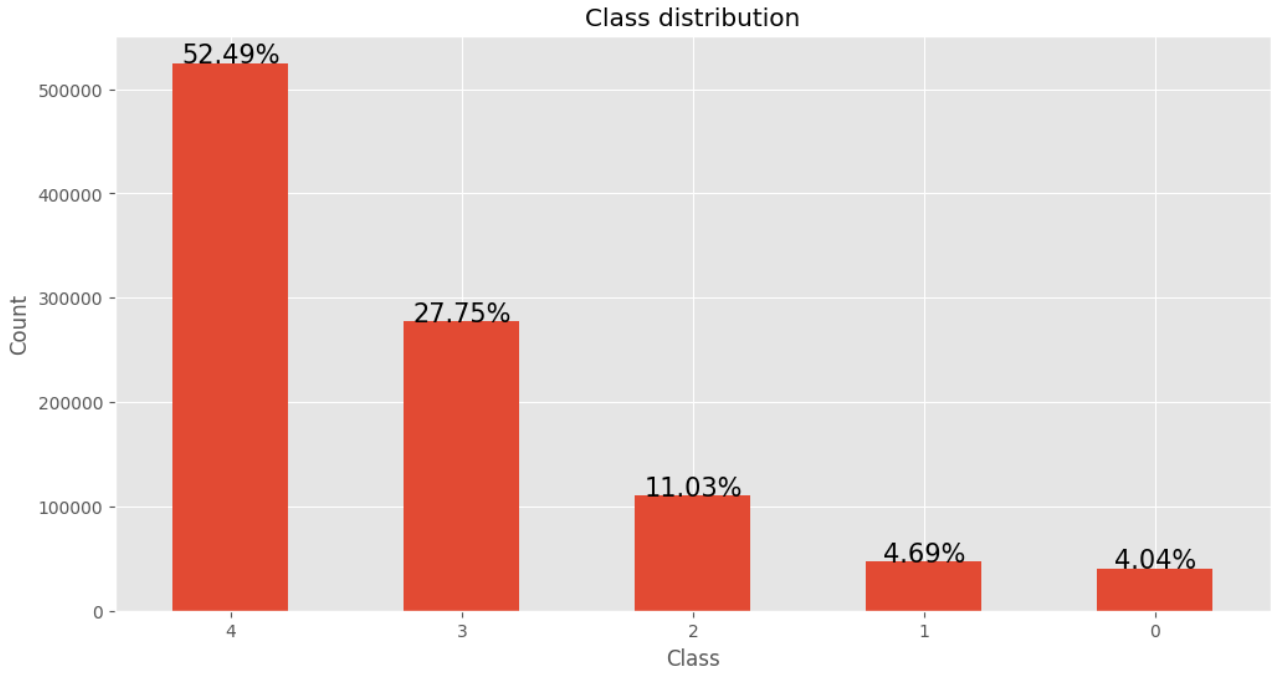
3. **Dostosowanie Oceny**: Definiowana jest kolejna funkcja `adjust_rating()`, aby dostosować ocenę przez odjęcie 1. Funkcja ta zapewnia, że oceny zaczynają się od 0 zamiast od 1.

4. **Zapis Danych do CSV**: Po wyciągnięciu tekstu i dostosowaniu oceny, kod zapisuje pary tekst-ocena do pliku CSV określonego jako ścieżka wyjściowa. Wykorzystuje moduł `csv.DictWriter`, aby zapisać dane z określonymi nazwami pól ('text' i 'rating').

2.2 Analiza zbiorów

2.2.1 Zbiór niezbalansowany

Sekcja ta została stworzona na podstawie pliku data_analysis.ipynb, który znajduje się w katalogu \data\raw. Funkcje odpowiedzialne za przygotowanie danych zostały zaimplementowane w pliku src\data\make_dataset.py.



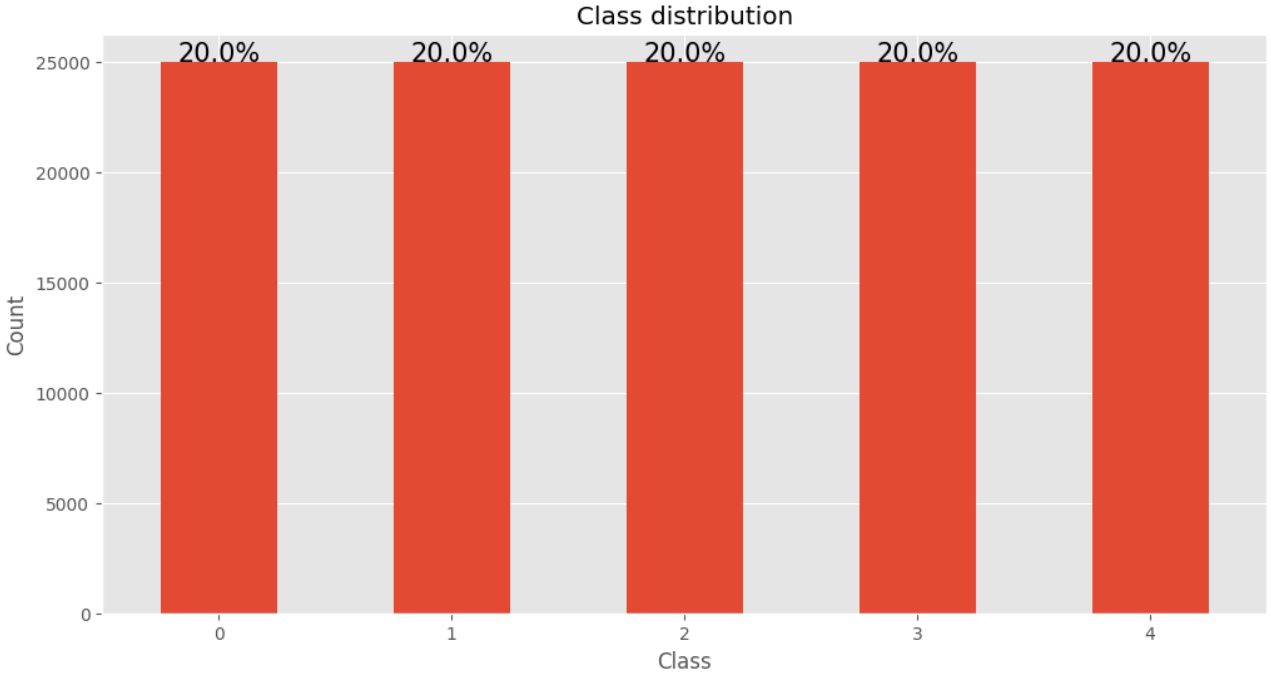
Rysunek 2: Dystrybucja klas dla zbioru niezbalansowanego

Mamy do czynienia ze zbiorem 1 miliona rekordów, rekordy składają się z opinii tekstowej oraz oceny dyskretnej w skali 0-4. Każda z klas oceny oznacza liczbę gwiazdek którą otrzymał hotel. Wyliczenie liczby gwiazdek połączonej do danej opinii tekstowej liczymy ze wzoru wartość + 1. Czyli na przykład opinia o klasie z wartością 4 oznacza opinię pięciogwiazdkową.

W zbiorze doszukaliśmy się 38 rekordów, które były duplikatami, pozbyliśmy się ich. Nie odnaleźliśmy żadnych wartości, które oznaczałyby brak opinii tekstowej.

Występują znaczne dysproporcje w kontekście dystrybucji atrybutu klasy. Klasa większościową, którą jest klasa reprezentująca opinię pięciogwiazdkową stanowi ponad 50% całego zbioru. Jest to zarazem plus jak i minus. Z uwagi na tą dysproporcję sensownym pomysłem będzie jako base-line modeli dla zbioru niezbalansowanego przyjąć model, który jako predykcję za każdym razem zwraca klasę większościową. Możemy się spodziewać, że statystycznie taki model będzie posiadał celność około 52.49% dla zbioru testowego. Dla modeli bardziej skomplikowanych, takich jak transformer RoBERTa, skutki tej dysproporcji muszą zostać zniwelowane poprzez zastosowanie warzenia klas w trakcie nauki. Wprowadzenie tego mechanizmu powinny zmniejszyć przeuczanie się modelu.

2.2.2 Zbiór zbalansowany



Rysunek 3: Dystrybucja klas dla zbioru zbalansowanego

Zbiór danych, o którym mowa, to zbiór zrównoważony z pięcioma klasami, z każdą klasą zawierającą 25 tysięcy egzemplarzy. Łącznie zbiór ten składa się z 125 tysięcy rekordów. Każdy rekord zawiera opinię tekstową oraz ocenę dyskretną w skali od 0 do 4. Wartość oceny odpowiada liczbie gwiazdek, jaką otrzymał hotel. Na przykład, opinia o klasie z wartością 4 oznacza opinie pięciogwiazdkową, ponieważ do oceny dodajemy 1, aby uzyskać liczbę gwiazdek. Zbiór danych jest zbalansowany, co oznacza, że każda klasa ma tę samą liczbę egzemplarzy. Takie podejście do tworzenia zbioru danych jest istotne, ponieważ zapewnia równomierny rozkład informacji pomiędzy różnymi klasami ocen, co może pomóc w uniknięciu stronniczości modelu w kierunku dominującej klasy. Jednak zbalansowany zbiór danych może wpłynąć na wyniki modelów predykcyjnych. Na przykład, niezbalansowane zbiory danych mogą prowadzić do modeli, które są dokładne dla dominującej klasy, ale słabo radzą sobie z klasyfikacją mniej licznych klas. W zbalansowanym zbiorze danych model może być bardziej zrównoważony w swoich predykcjach dla wszystkich klas. Z uwagi na to że w tym zbiorze nie istnieje klasa większościowa, jako baseline możemy przyjąć wyniki modelu losowego, który za każdym razem zwraca wylosowaną przez siebie klasę. Prawdopodobieństwo wylosowania poszczególnych klas jest takie samo, więc model powinien uzyskiwać wyniki na poziomie 20% celnych klasyfikacji.

2.3 Wykorzystane modele

2.3.1 Model nawiwny dla zbioru niezbalansowanego

Model naiwny zwraca zawsze klasę której liczba była największa przy zadanym zbiorze uczącym. Z uwagi na to, że grupa większościowa w przyjętym zbiorze danych stanowi 45% zbioru, spodziewamy się 100% miary Accuracy dla klasy oznaczającej opinię 5 gwiazdkową i 0% Accuracy dla pozostałych. Można było również przyjąć model zwracający zawsze wartość losową, ale zdecydowaliśmy się użyć model zwracający klasę większościową po to aby postawić poprzeczkę wyżej dla pozostałych modeli.

Zalety:

- Dobry baseline do porównywania jakości innych modeli.

Wady:

- Oczywista, niska jakość klasyfikatora.

2.3.2 Model losowy dla zbioru zbalansowanego

Z uwagi na to, że w zbiorze zbalanoswanym nie istnieje klasa większościowa, jako baseline możemy przyjąć wyniki modelu losowego, który za każdym razem zwraca wylosowaną przez siebie klasę. Prawdopodobieństwo wylosowania poszczególnych klas jest takie samo, więc model powinien uzyskiwać wyniki na poziomie 20% celnych klasyfikacji, co wynika z równego rozkładu prawdopodobieństwa dla każdej klasy. **Zalety**:

- Stanowi dobry punkt odniesienia do porównywania jakości innych modeli dla zbioru zbalansowanego.

Wady:

- Jakość klasyfikatora jest oczywiście niska.

2.3.3 Model transformer NLP BERT

BERT[2], czyli "Bidirectional Encoder Representations from Transformers", to zaawansowany model przetwarzania języka naturalnego (NLP), który opiera się na architekturze Transformer. Jest to model klasyfikacji, który analizuje nie tylko pojedyncze słowa, lecz także ich wzajemne relacje i kontekst w zdaniu.

Działanie BERTa opiera się na dwukierunkowym przetwarzaniu sekwencji słów, co oznacza, że uwzględnia zarówno lewostronne, jak i prawostronne konteksty każdego słowa w zdaniu. Dzięki temu model może lepiej zrozumieć znaczenie zdania i skuteczniej radzić sobie z różnymi zadaniami NLP.

BERT korzysta z mechanizmu transformerów, który umożliwia mu analizowanie długich sekwencji danych przy minimalnym utracie kontekstu. Transformer składa się z warstw kodujących i dekodujących, przy czym BERT wykorzystuje jedynie warstwy kodujące. Każda z tych warstw ma wiele mechanizmów samoatencji, które pozwalają modelowi na uwzględnianie zależności między różnymi słowami w zdaniu.

Model BERT jest trenowany na ogromnych zbiorach danych tekstowych, co pozwala mu na efektywne uczenie się różnych cech językowych. Podczas treningu model stara się przewidzieć następne słowo w zdaniu, mając na uwadze wszystkie słowa w jego kontekście. Ten proces uczenia pozwala modelowi na wyuczenie się reprezentacji słów, które są bogate w informacje semantyczne.

Zalety:

- Dwukierunkowe przetwarzanie - BERT uwzględnia przetwarzanie zdań zarówno z lewej, jak i prawej, co pozwala mu na lepsze zrozumienie ogólnego kontekstu.
- Model BERT osiąga świetne wyniki w wielu złożonych zadaniach NLP, takich jak klasyfikacja, rozpoznawanie nazw własnych, analiza sentymentu i wiele innych.
- BERT jest uniwersalnym modelem, który można dostosować do różnych zastosowań poprzez dodanie lub modyfikację wyjściowych warstw sieci.

Wady:

- Trenowanie modelu BERT wymaga dużych zasobów obliczeniowych, co może być trudne do osiągnięcia w przypadku braku odpowiedniej infrastruktury.
- Ze względu na swoją złożoność, trening modelu BERT może być czasochłonny, co może prowadzić do długiego czasu oczekiwania na wyniki.
- Aby uzyskać wysoką wydajność, model BERT wymaga treningu na dużych zbiorach danych, co może być problematyczne w przypadku braku odpowiednio dużych zbiorów danych treningowych.

2.3.4 Model transformer NLP RoBERTa

RoBERTa[4], czyli "Robustly Optimized BERT Approach", to zaawansowany model przetwarzania języka naturalnego (NLP) oparty na architekturze Transformer. RoBERTa jest modelem klasyfikacji, który wykorzystuje zaawansowane metody analizy cech do przypisania nowych danych tekstowych do odpowiednich kategorii lub zadań.

Działanie RoBERTa opiera się na przetwarzaniu sekwencji słów w celu zrozumienia kontekstu i znaczenia tekstu. W praktyce, RoBERTa dzięki mechanizmowi atencji[5] analizuje kontekst globalny zdania, uwzględniając relacje między słowami.

RoBERTa korzysta z zaawansowanych technik embeddingów BERT (Bidirectional Encoder Representations from Transformers), które przekształcają słowa na wektory numeryczne, reprezentujące ich semantykę i znaczenie. Te embeddingi są wykorzystywane przez model do uczenia się cech charakterystycznych dla różnych klas lub zadań NLP.

Aby przypisać etykietę lub wykonać inne zadanie NLP, RoBERTa korzysta z ogromnej ilości danych treningowych, w których słowa są powiązane z odpowiednimi etykietami lub informacjami. Podczas procesu uczenia RoBERTa analizuje te dane treningowe, aby nauczyć się, jakie cechy są charakterystyczne dla danej klasy lub zadania, oraz jak przypisać nowe dane tekstowe do tych klas.

Aby zaadaptować model RoBERTa do problemu klasyfikacji, można wykorzystać jego zdolność do przetwarzania sekwencji tekstowych i przypisywania ich do odpowiednich kategorii.

W naszym przypadku polegało to na dostosowaniu warstw wyjściowych, zamiast przewidywać kolejnego tokenu, wyjściem modelu RoBERTa w naszym przypadku jest liczba od 0 do 4, która oznacza klasę wiadomości wejściowej.

Zalety:

- RoBERTa oferuje znacznie lepsze zrozumienie kontekstu i semantyki tekstu dzięki wykorzystaniu architektury Transformer, co przekłada się na wyższą jakość predykcji w złożonych zadaniach NLP.
- Dzięki swojej głębokiej architekturze i trenowaniu na dużych zbiorach danych, RoBERTa może osiągać lepsze wyniki w zadaniach przetwarzania języka naturalnego w porównaniu do tradycyjnych modeli NLP.
- RoBERTa jest skalowalna i może być trenowana na bardzo dużych zbiorach danych, co pozwala na jej adaptację do różnorodnych zastosowań i dziedzin.

Wady:

- RoBERTa wymaga dużej ilości zasobów obliczeniowych i czasu trenowania ze względu na swoją złożoną architekturę i dużą liczbę parametrów, co może być ograniczeniem w przypadku braku procesorów z rdzeniami CUDA.
- Ze względu na swoją skomplikowaną naturę, RoBERTa może być trudna do interpretacji, co może stanowić problem w przypadku zastosowań, gdzie ważna jest transparentność działania modelu.
- Istnieje ryzyko nadmiernego dopasowania (overfitting) do danych treningowych ze względu na dużą pojemność modelu, co może prowadzić do obniżenia wydajności w przypadku nowych, nieznanych danych.

2.4 Porównanie modelu BERTa i RoBERTa

BERT i RoBERTa to zaawansowane modele przetwarzania języka naturalnego (NLP), oparte na architekturze Transformer. Oto główne różnice między nimi:

Architektura

BERT wykorzystuje tylko warstwy kodujące w architekturze Transformer. Każda z tych warstw zawiera mechanizmy samoatencji, które pozwalają modelowi na uwzględnienie relacji między słowami w zdaniu.

RoBERTa opiera się na architekturze BERT, ale jest bardziej zoptymalizowana pod względem hiperparametrów i procesu trenowania. RoBERTa również korzysta wyłącznie z warstw kodujących, co przekłada się na lepszą wydajność w porównaniu do dekodujących.

Trenowanie

BERT jest trenowany na dużych zbiorach danych tekstowych poprzez rozwiązanie dwóch zadań: Masked Language Model (MLM) oraz Next Sentence Prediction (NSP). Podczas treningu próbuje przewidzieć zmaskowane słowa w zdaniach oraz określić, czy drugie zdanie w parach zdaniowych faktycznie następuje po pierwszym zdaniu w naturalnym kontekście. Trening BERTa wymaga znacznych zasobów obliczeniowych i czasu.

RoBERTa korzysta z podobnej metody treningu, ale wymaga większej ilości danych oraz dłuższego czasu trenowania. RoBERTa osiąga wyższą wydajność niż BERT dzięki bardziej rozbudowanym i precyzyjnym reprezentacjom słów.

Wydajność

BERT mimo bardzo dobrych wyników w różnych zadaniach NLP, BERT może być ograniczony przez zasoby obliczeniowe i czas trenowania.

RoBERTa osiąga jeszcze lepsze wyniki dzięki bardziej rozbudowanemu procesowi trenowania. Dzięki zwiększonej ilości danych treningowych i dłuższemu czasowi trenowania, RoBERTa może przewyższać BERTa w złożonych zadaniach NLP.

Dwukierunkowość i globalny kontekst

BERT analizuje zarówno lewostronne, jak i prawostronne konteksty każdego słowa w zdaniu, co pozwala mu na lepsze zrozumienie znaczenia całego zdania.

RoBERTa bada globalny kontekst zdania, uwzględniając relacje między słowami. Dzięki mechanizmowi atencji RoBERTa może lepiej zrozumieć globalny kontekst i związki między słowami.

3 Realizacja Projektu

3.1 Wykorzystane technologie programistyczne

Technologie użyte w projekcie zostały wybrane ze względu na ich popularność, wszechstronność oraz zdolność do efektywnego przetwarzania języka naturalnego (NLP). Oto krótkie charakterystyki każdej z tych technologii:

- Python - powszechnie używanym język programowania w dziedzinie analizy danych, w tym również w przetwarzaniu języka naturalnego. Jego czytelna składnia, bogata biblioteka standardowa oraz wiele specjalistycznych bibliotek, takich jak NLTK, spaCy czy TensorFlow, czynią go popularnym wyborem w projekcie NLP.

- Jupyter - interaktywne środowisko programistyczne, które umożliwia tworzenie i udostępnianie interaktywnych notebooków zawierających kod, wykresy, tekst oraz równania. Jest idealnym narzędziem do eksploracyjnej analizy danych oraz dokumentowania procesu analizy w sposób interaktywny i zrozumiały dla innych.

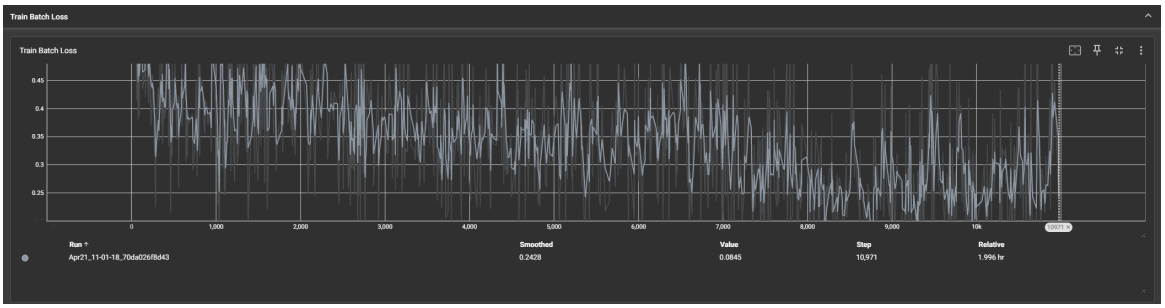
- Google Colab - platforma do pracy z notebookami Jupyter dostępna w chmurze. Zapewnia darmowy dostęp do zasobów obliczeniowych GPU i TPU, co pozwala na szybsze trenowanie modeli, szczególnie tych opartych na głębokim uczeniu. Jest to szczególnie przydatne w przypadku modeli NLP, które mogą wymagać dużych zasobów obliczeniowych.

- Cookiecutter - arzędzie do generowania projektów opartych na szablonach. Ułatwia ono tworzenie spójnych struktur projektowych, co jest przydatne w dużych projektach, takich jak ten związany z przetwarzaniem języka naturalnego. Dzięki Cookiecutter można łatwo zacząć nowy projekt, korzystając z dobrze przemyślanych struktur i najlepszych praktyk.

3.2 Wykorzystanie Tensorboard

TensorBoard to narzędzie wizualizacyjne opracowane przez TensorFlow, które umożliwia monitorowanie i analizę procesu uczenia się modelu maszynowego. Pozwala ono na śledzenie metryk wydajności modelu w czasie, takich jak funkcja straty, dokładność czy precyzja, w formie wykresów i tabel. TensorBoard oferuje także możliwość wizualizacji architektury modelu, dzięki czemu można zrozumieć jego strukturę i przepływ danych.

W projekcie, biblioteka ta została wykorzystana do wizualizacji zmiany wartości funkcji straty w czasie, w procesie uczenia i walidacji modelu w poszczególnych epokach. Przykładowe wykresy widoczne są poniżej:



Rysunek 4: Wykres wartości funkcji straty dla serii treningowej.

Rysunek 5: Wykres wartości funkcji straty w kolejnych epokach treningowych.

Wizualizacja efektów uczenia modeli pozwoliła na lepszy dobór hiperparametrów i ocenę jakości procesu trenowania modeli.

3.3 Dalszy rozwój

W dalszej części projektu, planuje się przeprowadzenie porównania modeli, w celu wyłonienia najlepszego z trzech modeli: Model naiwny, Model RoBERTa, Model BERTa oraz przeprowadzenie próby transferu wiedzy.

Porównanie modeli będzie polegało na ocenie ich wydajności na podstawie metryk Recall, Specificity oraz miary F1. Metryki te są istotne w przypadku zadań klasyfikacyjnych, szczególnie w kontekście przetwarzania języka naturalnego, gdzie istnieje potrzeba zminimalizowania liczby fałszywie negatywnych lub fałszywie pozytywnych klasyfikacji.

Transfer wiedzy zostanie przeprowadzony poprzez wykorzystanie modelu wytrenowanego na zbiorze danych zawierających recenzje hotelowe do oceny jego wydajności na zbiorze danych zawierającym recenzje filmów. Idea tego podejścia polega na sprawdzeniu, jak dobrze model poradzi sobie z zadaniami klasyfikacyjnymi w nowej dziedzinie, którą są recenzje filmów, pomimo że został wytrenowany na danych z dziedziny recenzji hotelowych.

Przeprowadzenie transferu wiedzy na takiej próbie umożliwi ocenę ogólnej zdolności modelu do generalizacji, czyli umiejętności stosowania nabytej wiedzy na nowych danych spoza zbioru treningowego. W przypadku udanego transferu wiedzy, można oczekiwać, że model będzie radził

sobie z klasyfikacją recenzji filmów na podobnym poziomie jak w przypadku recenzji hotelowych, co potwierdzi jego zdolność do wykrywania ogólnych wzorców i cech językowych, niezależnie od konkretnej dziedziny.

4 Wyniki działania modeli

4.1 Wyniki dla niezbalansowanego zbioru danych

Tabela 1: Metryki modeli dla zbioru niezbalansowanego

Model	Accuracy	Precision	Recall	F1-score	Support
Random	0.200	0.200	0.200	0.400	12500
BERT	0.691	0.697	0.691	0.691	12500
RoBERTa	0.692	0.698	0.692	0.692	12500

Modele BERT i RoBERTa osiągnęły podobne wyniki dla niezbalansowanego zbioru danych. Ich dokładność (Accuracy) wyniosła odpowiednio około 69.1% dla BERT i 69.2% dla RoBERTa. Zarówno precyzja (Precision), czułość (Recall), jak i F1-score dla obu modeli wyniosły około 69%, co wskazuje na równowagę między poprawnością klasyfikacji a zdolnością do wykrywania pozytywnych przypadków. Warto zauważyć, że oba modele osiągnęły te wyniki na tej samej liczbie przypadków, wynoszącej 12 500. Te rezultaty potwierdzają skuteczność i wydajność zarówno modelu BERT, jak i RoBERTa w klasyfikacji danych, szczególnie w przypadku niezbalansowanych zbiorów.

4.2 Wyniki dla zbalansowanego zbioru danych

Tabela 2: Metryki modeli dla zbioru zbalansowanego

Model	Accuracy	Precision	Recall	F1-score	Support
Random	0.200	0.200	0.200	0.400	12500
BERT	0.621	0.620	0.621	0.621	12500
RoBERTa	0.641	0.640	0.641	0.641	12500

Modele BERT i RoBERTa prezentują podobne wyniki w przypadku zbalansowanego zbioru danych. Ich dokładność (Accuracy) wynosiła odpowiednio około 62.1% dla BERT i 64.1% dla RoBERTa. W przypadku precyzji (Precision), czułości (Recall) oraz F1-score, oba modele osiągnęły wyniki na poziomie około 62.0% dla BERT i 64.0% dla RoBERTa. Te wyniki sugerują, że zarówno BERT, jak i RoBERTa, utrzymują równowagę między poprawnością klasyfikacji a zdolnością do wykrywania pozytywnych przypadków. Warto również zauważyć, że oba modele uzyskały te wyniki dla tej samej liczby przypadków, tj. 12 500. To potwierdza wysoką skuteczność i wydajność modeli BERT i RoBERTa w klasyfikacji danych w przypadku zbalansowanych zbiorów.

4.3 Wyniki dla poszczególnych klas dla zbioru zbalansowanego

Tabela 3: Metryki dla poszczególnych klas dla modelu RoBERTa i danych zbalansowanych

Class	Precision	Recall	F1-score	Support
0	0.69	0.75	0.72	2000
1	0.54	0.53	0.53	2000
2	0.60	0.55	0.57	2000
3	0.60	0.60	0.60	2000
4	0.76	0.78	0.77	2000

Wyniki dla poszczególnych klas dla modelu BERT na danych zbalansowanych są przedstawione powyżej. Dla każdej klasy podano precyzję, czułość, F1-score oraz wsparcie (liczbę przypadków) w zbiorze danych. Wyniki wskazują na różnice w wydajności modelu w klasyfikacji różnych klas, przy czym niektóre klasy uzyskują lepsze wyniki niż inne. Zauważalnie lepiej model głęboki radzi sobie z klasyfikacją opinii skrajnych niż np. tych z klasy opinii dwugwiazdkowych.

	Klasa 0	Klasa 1	Klasa 2	Klasa 3	Klasa 4
Klasa 0	1491	450	50	3	6
Klasa 1	550	1059	348	30	13
Klasa 2	105	422	1093	349	31
Klasa 3	10	25	322	1193	450
Klasa 4	7	3	14	426	1550

Na górze macierzy opisane są faktyczne klasy, a po lewej stronie przewidziane klasy przez model. Macierz pomyłek wyraźnie przedstawia, że zaskakująco dobre wyniki klasyfikacji skrajnych opinii wynikają z tego, że model często przy klasyfikacji myli się i przypisuję klasę sąsiednią. Z uwagi na to, że klasy skraje mają tylko jednego sąsiada, to i wyniki klasyfikacji są lepsze.

4.4 Wyniki dla problemu uproszczonego: opinia pozytywna lub negatywna

Podzielono opinie na zbiór pozytywnych i negatywnych. Opinie jedno, dwu i trzy-gwiazdkowe zaliczyliśmy do opinii negatywnych, natomiast opinie cztero i pięcio gwiazdkowe do pozytywnych.

Tabela 4: Metryki modeli dla zbioru o dwóch klasach

Model	Accuracy	Precision	Recall	F1-score	Support
Random	0.500	0.500	0.500	0.500	13000
BERT	0.824	0.833	0.824	0.823	13000
RoBERTa	0.818	0.818	0.818	0.818	13000

Wyniki przedstawione w powyższej tabeli obejmują dokładność (Accuracy), precyzję (Precision), czułość (Recall) oraz F1-score dla modeli Random, BERT i RoBERTa w zadaniu kla-

syfikacji opinii na dwie kategorie: pozytywne i negatywne. Model BERT uzyskał najwyższą dokładność, precyzję, czułość i F1-score w porównaniu do modelu losowego i RoBERTa. Natomiast modele BERT i RoBERTa osiągnęły zbliżone wyniki, sugerując ich podobną skuteczność w klasyfikacji tego rodzaju danych. Bert tym razem poradził sobie trochę lepiej. W porównaniu do wyników dla problemu 5 klasowego metryki, tak jak można było się spodziewać, poprawiły się.

5 Transfer wiedzy

W kontekście uczenia maszynowego, transfer wiedzy odnosi się do wykorzystania wiedzy lub doświadczenia zdobytego przez model w jednym zadaniu do poprawy wyników w innym zadaniu lub dziedzinie. Może to obejmować wykorzystanie wstępnie wytrenowanych modeli jako punkt wyjścia do nauki nowych zadań, przenoszenie wag (parametrów) z jednego modelu do innego w celu przyspieszenia procesu uczenia się lub poprawy generalizacji, oraz wykorzystanie wiedzy lub cech wyodrębnionych z jednego zbioru danych do poprawy wydajności w innej dziedzinie lub zbiorze danych. Transfer wiedzy w uczeniu maszynowym ma na celu wykorzystanie istniejącej wiedzy w jak najbardziej efektywny sposób, aby poprawić wyniki uczenia się modelu w nowych zadaniach lub kontekstach.

Do analizy transferu wiedzy z wytrenowanego modelu, wykorzystany został zbiór danych z recenzjami filmów [3]. Na początku zbadana została jakość modelu wytrenowanego na zbiorze danych hotelowych w starciu z nowym zbiorem - recenzji filmowych. Wyniki modelu przedstawia poniższa tabelka:

Tabela 5: Metryki modelu dla zbioru recenzji filmowych

Accuracy	Precision	Recall	F1-score	Support
0.7632	0.7886	0.7632	0.7578	2500

Z powyższych danych wynika, że model bardzo dobrze poradził sobie z klasyfikacją pokrewnych w dziedzinie danych. Może to mieć związek z charakterem zadania - słowa kluczowe, nacechowane emocjonalnie mają to samo znaczenie zarówno w kontekście recenzji hotelowych jak i filmowych. Wyrażenia takie jak "beznadziejne"lub "niezłe"niosą tę samą informację, przez co model dobrze radzi sobie w klasyfikacji nowego zbioru danych.

Wytrenowane na danych recenzji hotelowych model Bert wraz z tokenizerem (w wersji detekcji klas pozytywna / negatywna) zostały następnie poddane treningowi na wyżej wymienionym zbiorze danych z recenzjami filmowymi. Po zaledwie jednej epoce, uzyskane wyniki są bardzo obiecujące:

Model uzyskał b. dobre wyniki na nowym zbiorze danych, na co wpływ niezaprzeczalnie miało wcześniej opisane podobieństwo dziedzin problemów.

Tabela 6: Metryki dotrenowanego modelu dla zbioru recenzji filmowych

Accuracy	Precision	Recall	F1-score	Support
0.8916	0.8935	0.500	0.8914	2500

6 Podsumowanie

Przeprowadzono porównanie modeli BERTa i RoBERTa w zadaniu klasyfikacji recenzji, wykorzystując zbalansowane i niezbalansowane zbiory danych. Wyniki pokazały, że oba modele osiągnęły wysoką skuteczność w klasyfikacji zarówno na zrównoważonych, jak i niezrównoważonych danych. Szczególnie interesujące było odkrycie, że model BERT, wytrenowany na recenzjach hotelowych, skutecznie przeniósł swoją wiedzę na zadanie klasyfikacji recenzji filmowych, co potwierdza możliwość transferu wiedzy między dziedzinami.

6.1 Ciekawe odkrycie

Jednym z najbardziej interesujących odkryć było potwierdzenie skuteczności transferu wiedzy między dwiema różnymi dziedzinami - recenzjami hotelowymi i filmowymi. Model BERT, wytrenowan na zbiorze danych z recenzjami hotelowymi, osiągnął wysoką dokładność i skuteczność w klasyfikacji recenzji filmowych. Jest to interesujące odkrycie, sugerujące, że podobieństwo tematyczne oraz strukturalne cechy obu zbiorów danych pozwoliły modelowi efektywnie wykorzystać swoją wiedzę do nowego zadania. Recenzje z obu dziedzin mogą zawierać podobne wyrażenia, słowa kluczowe i nacechowanie emocjonalne, co ułatwiło adaptację modelu do nowego kontekstu. To z kolei sugeruje, że modele NLP mogą być skuteczne w transferze wiedzy między różnymi dziedzinami, jeśli te dziedziny mają podobną strukturę i charakterystykę tekstu.

6.2 Dalszy rowzój projektu

Istnieje wiele możliwości dalszego rozwoju w tej dziedzinie. Przede wszystkim warto kontynuować badania nad transferem wiedzy, eksplorując inne dziedziny, w których można wykorzystać wstępnie wytrenowane modele do poprawy wyników na nowych zadaniach. Ponadto, optymalizacja hiperparametrów modeli oraz trening na większych zbiorach danych mogłaby przynieść dalsze polepszenie wyników. Dodatkowo, eksperymenty z innymi architekturami modeli, jak również zastosowanie modeli głębokich.

7 Literatura

[1] Diego Antognini and Boi Faltings. Hotelrec: a novel very large-scale hotel recommendation dataset. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4917–4923, Marseille, France, May 2020. European Language Resources Association.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[3] V. Lakshmi and S. Npathi. Imdb dataset of 50k movie reviews. `https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews`, 2020.

[4] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

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