

**UNIWERSYTET WARMIŃSKO-MAZURSKI W OLSZTYNIE
WYDZIAŁ MATEMATYKI I INFORMATYKI**

Kierunek: Informatyka

Adam Trentowski

**Wieloklasowa analiza sentymentu
z wykorzystaniem modeli NLP –
porównanie metod**

Praca magisterska wykonana
w Katedrze Metod Matematycznych Informatyki
pod kierunkiem
dr Agnieszki Zbrzezny

Olsztyn, 2025 rok

UNIVERSITY OF WARMIA AND MAZURY IN OLSZTYN
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Field of Study: Computer Science

Adam Trentowski

**Multiclass sentiment analysis
using NLP models –
method comparison**

Master's Thesis is performed
in the Department of Mathematical
Methods of Computer Science
under supervision of
Agnieszka Zbrzezny, PhD

Olsztyn, 2025

Spis treści

Wstęp	3
Cel pracy i pytania badawcze	3
Struktura pracy	3
Rozdział 1. Wprowadzenie	4
1.1. Krótka historia rozwoju modeli NLP	4
Rozdział 2. Przegląd literatury	6
2.1. Modele statystyczne	6
2.2. Modele sekwencyjne	7
2.3. Modele transformatorowe	8
2.4. Podsumowanie	9
Rozdział 3. Zbiory danych i przetwarzanie tekstu	10
3.1. Opis zbiorów danych do analizy sentymetu	10
3.1.1. Twitter Tweets Sentiment Dataset	10
3.1.2. Multiclass Sentiment Analysis Dataset	10
3.1.3. TweetEval	11
3.1.4. Łączenie zbiorów danych	11
3.2. Opis zbioru danych do analizy emocji	11
3.3. Czyszczenie i przygotowanie zbiorów	12
3.4. Statystyczna charakterystyka danych po czyszczeniu	13
3.4.1. Zbiór do analizy emocji	14
Rozkład klas	14
Analiza długości tekstów	14
Analiza częstości występowania słów	16
3.4.2. Zbiór do analizy sentymetu	18
Rozkład klas	18
Analiza długości tekstów	18
Analiza częstości występowania słów	20
3.5. Podział danych na zestawy treningowe, walidacyjne i testowe	21
Rozdział 4. Modele i metody treningu	22
4.1. Model SVM	22
4.1.1. Zastosowana implementacja i parametry	22
4.1.2. Proces treningu	23
4.2. Model LSTM	23
4.2.1. Zastosowana implementacja i parametry	23
4.2.2. Proces treningu	24
4.3. Model GRU	25
4.3.1. Zastosowana implementacja i parametry	25
4.3.2. Proces treningu	26

4.4.	Model BERT	26
4.4.1.	Zastosowana implementacja i parametry	27
4.4.2.	Proces treningu	27
4.5.	Reprodukcia wyników	29
4.6.	Miary skuteczności	29
Rozdział 5. Eksperymenty i wyniki		30
5.1.	Modele jednozadaniowe dla analizy sentymentu	30
5.2.	Modele jednozadaniowe dla analizy emocji	31
5.3.	Modele wielozadaniowe	31
5.3.1.	Zestawienie modeli wielozadaniowych	31
5.3.2.	Porównanie modeli jednozadaniowych i wielozadaniowych	32
5.4.	Analiza błędów modeli	33
5.4.1.	Klasyfikacja sentymentu	33
5.4.2.	Klasyfikacja emocji	36
Rozdział 6. Wnioski i podsumowanie		39
6.1.	Najskuteczniejszy model w klasyfikacji sentymentu i emocji	39
6.2.	Wpływ uczenia wielozadaniowego	39
6.3.	Czas treningu i zasoby obliczeniowe	40
6.4.	Wyzwania i ograniczenia modeli NLP w analizie sentymentu	40
6.5.	Podsumowanie	40
Bibliografia		41
Spis rysunków		43
Spis tabel		44
Streszczenie		45
Abstract		46

Wstęp

Współczesne media społecznościowe, stały się głównym źródłem komunikacji i wyrażania opinii. Każdego dnia użytkownicy publikują olbrzymią liczbę wpisów, w których dzielą się emocjami, nastrojami oraz opiniami na różne tematy – od polityki, przez kulturę, po codzienne wydarzenia. Analiza tych treści może dostarczyć cennych informacji w wielu dziedzinach, takich jak marketing, socjologia, psychologia czy nawet analiza ryzyka w biznesie. W szczególności analiza sentymentu pozwala na automatyczne określanie, jakie emocje i nastawienie wyrażają użytkownicy w swoich wypowiedziach [1]. Dynamiczny rozwój metod przetwarzania języka naturalnego (*Natural Language Processing*, NLP) umożliwia stosowanie coraz bardziej zaawansowanych technik klasyfikacji sentymentu i emocji, co znacząco zwiększa możliwości interpretacji oraz wykorzystania danych tekstowych w badaniach naukowych i biznesie.

Cel pracy i pytania badawcze

Niniejsza praca ma na celu przeprowadzenie porównania skuteczności wieloklasowej analizy sentymentu oraz identyfikacji emocji z wykorzystaniem modeli NLP, takich jak BERT (*Bidirectional Encoder Representations from Transformers*), SVM (*Support Vector Machines*), LSTM (*Long Short-Term Memory*) i GRU (*Gated Recurrent Units*). Badanie obejmuje zarówno klasyfikację sentymentu, jak i rozpoznawanie emocji w tekście. Celem jest określenie, który z tych modeli osiąga najlepsze wyniki oraz czy uczenie wielozadaniowe, czyli jednoczesne trenowanie modelu do analizy sentymentu i emocji, wpływa na dokładność predykcji.

W ramach badania podjęto próbę odpowiedzi na następujące pytania:

- Który model uczenia maszynowego osiąga najwyższą dokładność w klasyfikacji sentymentu?
- Który model jest najbardziej efektywny w identyfikacji emocji?
- Jak uczenie wielozadaniowe wypada w porównaniu do klasyfikacji jednozadaniowej?
- Jakie są główne wyzwania i ograniczenia związane z wykorzystaniem różnych modeli NLP do analizy sentymentu i emocji?

Struktura pracy

Kolejne rozdziały zawierają wprowadzenie i przegląd literatury obejmujący badania dotyczące analizy sentymentu oraz technik przetwarzania języka naturalnego. Następnie opisane zostały wykorzystane zbiory danych, ich źródła oraz struktura. Kolejna część pracy poświęcona została opisowi zastosowanej metodologii, z uwzględnieniem opisów implementacji modeli i strategii ich trenowania. Część eksperymentalna obejmuje testy porównawcze, mające na celu ocenę efektywności różnych podejść. Ostatni rozdział zawiera podsumowanie przeprowadzonych badań oraz wnioski wynikające z uzyskanych wyników.

Rozdział 1

Wprowadzenie

W literaturze przedmiotu analiza sentymentu i emocji definiowana jest jako dziedzina badań analizująca opinie, sentymenty, oceny, postawy i emocje ludzi wobec różnych podmiotów, takich jak osoby, organizacje, produkty, usługi czy wydarzenia [1]. Wymienione opinie często przekazywane są w formie pisanej, głównie za pośrednictwem platform internetowych, takich jak Facebook, X (dawniej Twitter), Reddit czy inne media społecznościowe, gdzie użytkownicy regularnie publikują komentarze i posty wyrażające ich zdanie na dany temat. Na przykład, według Internet Live Stats, dziennie na platformie X pojawia się 500 milionów komentarzy i wpisów [2]. Ze względu na ogólną liczbę dostępnych tekstów, ręczna identyfikacja opinii w nich zawarta, jest zarówno trudna jak i czasochłonna, co uzasadnia potrzebę stosowania systemów zautomatyzowanych [1].

W ramach szeroko rozumianej analizy sentymentu można wyróżnić dwa główne podejścia:

1. Analiza sentymentu – polega na klasyfikacji tekstu pod względem ogólnego wydźwięku na kategorie takie jak pozytywny, negatywny czy neutralny. Wyzwania w tej dziedzinie NLP wynikają głównie z problemów kontekstowych oraz wieloznaczności językowej. Sarказm lub ironia mogą prowadzić do błędnych interpretacji, gdyż wydźwięk słów pozornie pozytywnych może mieć negatywne znaczenie, np. „Świetnie, uwielbiam stać w korkach!” [3].
2. Analiza emocji – koncentruje się na wykrywaniu bardziej szczegółowych stanów emocjonalnych, takich jak radość, smutek czy gniew. W tym wypadku również często wymagane jest zrozumienie kontekstu oraz niuansów językowych. Dodatkową trudnością może być współwystępowanie różnych emocji, co utrudnia jednoznaczna klasyfikację. Na przykład tekst może jednocześnie wyrażać strach i zaskoczenie, jak w zdaniu „Nie mogłem uwierzyć własnym oczom, kiedy nagle zobaczyłem tę dziwną postać – przeszywający mnie strach sparaliżował mnie na chwilę”, lub smutek i gniew, co ilustruje wypowiedź: „Czuję ból po stracie, a jednocześnie złość, że nie mogłem nic zrobić”. Z drugiej strony, w niektórych przypadkach emocje są wyjątkowo wyraziste i jednoznaczne, zwłaszcza w tekstuach nacechowanych emocjonalnie lub w krótkich, intensywnych wypowiedziach jak wybuchi gniewu – „Nienawidzę tego!” czy wyrazy uwielbienia – „Kocham to!”. W takich sytuacjach modele klasyfikacyjne mogą osiągać wysoką skuteczność bez potrzeby skomplikowanego przetwarzania kontekstowego [4].

1.1. Krótka historia rozwoju modeli NLP

Analiza sentymentu i emocji w NLP ulega w ostatnich latach ciągłeemu rozwojowi. Rozwój ten można podzielić na kilka etapów, odzwierciedlających postęp w metodach przetwarzania języka naturalnego. Początkowo, w latach 50. i 60. XX wieku, w NLP stosowano podejścia oparte na regułach i słownikach, takie jak tłumaczenie maszynowe przy użyciu prostych metod odwzorowania wyrazów [5]. W latach 80. XX wieku rozpoczęła się stopniowa transformacja w kierunku uczenia maszynowego – dziedziny, która zastępowała ręcznie pisane reguły algo-

rytmami uczącymi się na danych [6]. Wraz z rozwojem uczenia maszynowego oraz rosnącą dostępnością mocy obliczeniowej w latach 90. XX wieku zaczęto wykorzystywać modele klasyfikacyjne, takie jak *Support Vector Machines* (SVM) oraz *Naive Bayes*, które analizowały tekst na podstawie wyodrębnionych cech [7].

Lata 2010. przyniosły kolejny postęp, kiedy to modele oparte na sieciach neuronowych, takie jak *Long Short-Term Memory* (LSTM) oraz *Gated Recurrent Units* (GRU), zaczęły być szeroko stosowane. LSTM, zaprezentowane po raz pierwszy w 1997 roku, zyskało na popularności właśnie na początku lat 2010. GRU, zaprezentowane w 2014 roku, stanowiło uproszczoną wersję LSTM, zapewniając podobne efekty, ale z mniejszymi wymaganiami obliczeniowymi. Dzięki zdolności do modelowania długoterminowych zależności w danych tekstowych, oba modele poprawiły jakość analizy sentymentu i emocji. Okazały się także skuteczne w przypadku dłuższych i bardziej złożonych tekstów [8].

Kolejnym kamieniem milowym było wprowadzenie modelu BERT (*Bidirectional Encoder Representations from Transformers*) przez firmę Google w 2018 roku. Model ten wykorzystuje mechanizm samouwagi (*self-attention*), umożliwiający równoczesne uwzględnienie kontekstu zarówno poprzedzających, jak i następujących słów w zdaniu. Dzięki temu model potrafi uchwycić dwukierunkowe zależności między słowami, co znacząco poprawiło dokładność w zadaniach klasyfikacyjnych [9].

Warto również wspomnieć o rozwinięciach modelu BERT, takich jak RoBERTa, ALBERT czy DistilBERT, które oferują różne kompromisy pomiędzy szybkością działania, rozmiarem modelu oraz efektywnością. RoBERTa poprawia jakość poprzez zmodyfikowany proces treningowy i większe zbiory danych [10], ALBERT redukuje rozmiar modelu poprzez współdzielenie parametrów [11], a DistilBERT stanowi wersję uproszczoną, lżejszą i szybszą kosztem nieznacznej utraty dokładności [12]. W niniejszej pracy zdecydowano się jednak na wykorzystanie oryginalnego modelu BERT ze względu na jego szeroką dostępność, ugruntowaną pozycję w literaturze oraz umiarkowane wymagania zasobowe, które okazały się wystarczające dla badanego problemu. Ponadto, zastosowanie BERT-a umożliwia łatwe porównanie wyników z innymi badaniami.

W niniejszej pracy porównane zostaną wybrane modele uczenia maszynowego, które odzwierciedlają ewolucję podejść stosowanych w NLP na przestrzeni ostatnich dekad. Analiza obejmie zarówno klasyczne metody, jak i nowoczesne architektury oparte na sieciach neuronowych i transformatorach, umożliwiając ocenę ich skuteczności oraz efektywności w kontekście klasyfikacji sentymentu i emocji w tekstach.

Rozdział 2

Przegląd literatury

Rozdział ten stanowi zestawienie współczesnych badań dotyczących analizy sentymentu i emocji w kontekście przetwarzania języka naturalnego. Omówione zostaną zarówno porównania tradycyjnych podejść statystycznych, jak i nowoczesnych metod opartych na głębokim uczeniu.

2.1. Modele statystyczne

Wśród klasycznych metod analizy sentymentu, tj. metod stosowanych przed powszechnym wykorzystaniem sieci neuronowych, najczęściej wykorzystywanymi podejściami były Maszyna Wektorów Nośnych (SVM) oraz *Naive Bayes*. *Naive Bayes* cechuje się jednak niższą skutecznością, to z powodu podatności na problemy związane z niezależnością cech. Model ten zakłada bowiem warunkową niezależność między słowami, co w praktyce często nie jest prawdą, zwłaszcza w przypadku tekstu naturalnych o złożonej strukturze [13].

Badania przeprowadzone w ostatnich latach potwierdzają, że *Naive Bayes* generalnie radzi sobie gorzej w porównaniu do nowszych metod. Przykładowo, w pracy z 2021 roku [14], autorzy porównali skuteczność SVM i *Naive Bayes*, z uwzględnieniem różnych technik ekstrakcji cech, w kontekście analizy komentarzy z blogów. W badaniu zastosowano zarówno tradycyjne metody reprezentacji tekstu, takie jak TF-IDF, jak i podejścia hybrydowe, które łączyły klasyczne modele z elementami głębokiego uczenia. Trening modeli i ich ewaluacja odbywały się przy użyciu standardowych miar klasyfikacyjnych, takich jak dokładność (ang. *accuracy*). W porównaniu do powyższego badania niniejsza praca poszerza zakres modeli o architektury oparte na głębokich sieciach neuronowych oraz modele transformatorowe. Ponadto, podejście zastosowane w omawianym badaniu różni się od podejścia niniejszej pracy, gdzie skoncentrowano się przede wszystkim na porównaniu wydajności algorytmów uczenia maszynowego bez stosowania technik przetwarzania i ekstrakcji cech. Omawiane badanie stanowi jednak punkt odniesienia dla niniejszego projektu, zwłaszcza w kontekście klasycznych metod klasyfikacji tekstu. Wyniki przeprowadzonych eksperymentów potwierdziły, że pomimo swojej prostoty i niskich wymagań obliczeniowych, algorytm *Naive Bayes* cechuje się niższą skutecznością w porównaniu do Maszyny Wektorów Nośnych (SVM).

Podobne wnioski płyną z pracy z roku 2022 [15], w której autorzy porównali algorytmy SVM i *Naive Bayes* pod kątem klasyfikacji mowy nienawiści w mediach społecznościowych. W badaniu zastosowano standardowe techniki przygotowania danych, co jest zgodne z podejściem wykorzystanym w niniejszej pracy, a następnie przeprowadzono trening i ewaluację modeli przy użyciu standardowych miar skuteczności, które również zostaną użyte w niniejszym badaniu. W odróżnieniu od omawianej pracy, która koncentruje się na prostych klasyfikatorach oraz specyficznym zadaniu detekcji mowy nienawiści, niniejsze badanie rozszerza zakres analizy o klasyfikację sentymentu oraz emocji, porównując jednocześnie także modele nowsze i bardziej zaawansowane. Wyniki omawianej pracy wskazały, że algorytm SVM osiągnął bardzo wysoko-

ką dokładność klasyfikacji, sięgającą około 99%, podczas gdy algorytm *Naive Bayes* uzyskał znacznie niższy wynik, wynoszący około 50%.

Dlatego w niniejszej pracy skupiono się na modelu SVM, który mimo swojej prostoty nadal okazuje się konkurencyjny w niektórych zadaniach klasyfikacyjnych. Przykładem jest badanie [16], w którym autorzy porównali skuteczność Maszyny Wektorów Nośnych z wcześniej wytrenowanymi modelami językowymi (ang. *Pretrained Language Model*, PLM), takimi jak BERT, na czterech zbiorach danych. Wyniki wykazały, że nawet po dostrojeniu, PLM-y nie przewyższyły znacząco prostego klasyfikatora SVM. Autorzy sugerują, że w zadaniach klasyfikacji tekstu tradycyjne podejście oparte na SVM, wspierane staranną inżynierią cech, może być bardziej efektywne kosztowo i wydajnościowo niż stosowanie zaawansowanych modeli językowych. Podczas gdy opisane badanie kładzie nacisk na inżynierię cech oraz wykorzystanie gotowych modeli PLM, w niniejszej pracy zastosowano podejście oparte na samodzielnym trenowaniu modeli (za wyjątkiem BERT-a, który jest wstępnie wytrenowany) bez ingerencji w przetwarzanie tekstu czy dodatkową inżynierię cech, co pozwoli na analizę ich zachowania i skuteczności w kontekście różnych zadań NLP. Niemniej, gdy teksty zawierają niejasności, takie jak ironia, model SVM może mieć trudności w uchwyceniu pełnego kontekstu, co ogranicza jego skuteczność.

2.2. Modele sekwencyjne

Modele sekwencyjne, takie jak LSTM i GRU, wykazują wysoką skuteczność w analizie sentymentu i emocji, a to dzięki zdolności do modelowania długoterminowych zależności w danych tekstowych. W szczególności są one bardziej efektywne w przypadku dłuższych i bardziej złożonych tekstów. W badaniu [17] przeprowadzono porównanie trzech podejść do analizy sentymentu na danych pochodzących z różnych platform społecznościowych. Autorzy wykorzystali metody leksykalne oparte na słownikach (*TextBlob*), klasyczny model probabilistyczny *Naive Bayes* oraz model sieci neuronowej LSTM, zdolny do wychwytywania wzorców i długoterminowych zależności w tekście. Dane zostały poddane standardowym procesom przygotowania, obejmującym tokenizację i normalizację tekstu, a każdy model trenowano oddziennie na zbiorach pochodzących z poszczególnych platform. Ewaluacja skuteczności odbywała się przy użyciu standardowych miar. Wyniki wykazały, że model LSTM systematycznie przewyższał zarówno podejście leksykalne jak i probabilistyczne. W porównaniu do omawianego badania niniejsza praca rozszerza zakres analizowanych modeli. Podobnie jak w opisywanym badaniu, opiera się na danych tekstowych z mediów społecznościowych, jednak różni się podejściem metodologicznym, koncentrując się na minimalnej ingerencji w preprocessing danych. Ponadto analiza obejmuje zarówno klasyfikację sentymentu, jak i emocji oraz dodatkowo rozszerza ocenę modeli o aspekt efektywności obliczeniowej i czasu treningu.

W publikacji [18] porównano ponad 100 podejść opartych na głębokim uczeniu do klasyfikacji sentymentu na 21 publicznie dostępnych zbiorach danych z recenzjami klientów. Analiza ta wykazała, że modele sekwencyjne, takie jak LSTM i GRU, osiągają generalnie wyższą dokładność niż tradycyjne metody klasyfikacji, na przykład SVM. Autorzy wspomnianej pracy zauważali ponadto, że modele trenowane na zbalansowanych zbiorach danych uzyskiwały wyższe wyniki niż te uczone na zbiorach niebalansowanych, co stanowi istotną przesłankę do stosowania technik równoważenia klas. Proces przetwarzania danych ograniczał się natomiast do standardowych procedur, bez zastosowania zaawansowanej inżynierii cech, co jest zbieżne z podejściem przyjętym w niniejszym badaniu. W odróżnieniu od pracy [18], niniejsze badanie nie tylko porównuje skuteczność wybranych modeli uczenia maszynowego przy minimalnej in-

gerencji w dane wejściowe, ale również analizuje wpływ podejścia wielozadaniowego na jakość predykcji. Głównym celem jest stworzenie neutralnego, ujednoliconego środowiska eksperymentalnego, które umożliwia obiektywną ocenę skuteczności oraz efektywności wybranych modeli w zadaniach jedno- i wielozadaniowych.

2.3. Modele transformatorowe

Modele transformatorowe, takie jak BERT, stanowią obecnie jedno z najskuteczniejszych podejść w analizie sentymentu. Ich przewaga nad wcześniejszymi technikami, wynika przede wszystkim z zastosowania mechanizmu samouwagi (ang. *self-attention*), który pozwala modełowi analizować kontekst danego słowa zarówno z lewej, jak i z prawej strony. W pracy autorów modelu BERT [9] przedstawiono wyniki eksperymentów, które wykazały, że model ten przewyższa wcześniejsze podejścia, zarówno tradycyjne, jak i oparte na sieciach neuronowych, w sześciu standardowych zadań NLP, takich jak m.in. klasyfikacja tekstu czy analiza sentymentu. Na tej podstawie można wysunąć hipotezę, że w ramach przeprowadzonych w niniejszej pracy eksperymentów, model BERT osiągnie najlepsze wyniki spośród analizowanych architektur.

Również w nowszych badaniach dotyczących analizy emocji, modele transformatorowe potwierdzają swoją przewagę. Przykładem jest praca [19], w której przeprowadzono porównanie kilku modeli, w tym DistilBERT, ELECTRA, Twitter-RoBERTa oraz LSTM z osadzaniem GloVe, w zadaniu klasyfikacji emocji na zbiorze GoEmotions. Dane przetworzono, poprzez zmniejszenie liczby etykiet (z 6 do 4) oraz przez usunięcie m.in. znaczników HTML, adresów e-mail, znaków interpunkcyjnych, specjalnych oraz stopwordów. Wyniki eksperymentów wykazały, że wariant modelu transformatorowego BERT osiągnął najwyższą dokładność, mimo ograniczonej liczby danych treningowych. Co istotne, autorzy zauważyl, że intensywne przetwarzanie tekstu może pogarszać wyniki modeli transformatorowych, które do skutecznego działania wymagają dostępu do pełnego kontekstu językowego. W niniejszej pracy skupiono się na minimalnej ingerencji w dane wejściowe, tak aby zapewnić optymalne warunki porównania modeli bez dodatkowej inżynierii cech, co jest podejściem zbieżnym z obserwacjami autorów. Dodatkowo, w przeciwieństwie do badania [19], zastosowano zbalansowane zbioru danych oraz rozszerzono zakres eksperymentów o analizę wpływu uczenia wielozadaniowego.

Skuteczność modelu BERT w zadaniu klasyfikacji sentymentu została również potwierdzona w badaniu [20], w którym autorzy porównali m.in. model LSTM oraz model BERT. Eksperymenty przeprowadzono na zbiorze danych z Twittera, klasyfikując wypowiedzi na pięć poziomów sentymentu: od skrajnie negatywnego do skrajnie pozytywnego. Modele oceniano przy użyciu standardowych miar skuteczności, takich jak dokładność, precyzyja, czułość oraz F1-score. W badaniu wykorzystano klasyczne techniki przetwarzania tekstu, w tym usuwanie znaków specjalnych, tokenizację oraz eliminację stopwordów. Wyniki wykazały, że model BERT osiągnął najwyższe wskaźniki skuteczności, przewyższając zarówno klasyczny model oparty na częstotliwości słów, jak i model LSTM. Choć LSTM radził sobie zauważalnie lepiej niż podejście częstotliwościowe, to nadal ustępował BERTowi pod względem dokładności. Niniejsza praca rozszerza analizę o modele SVM oraz uwzględnia wpływ uczenia wielozadaniowego.

2.4. Podsumowanie

Badania jednoznacznie potwierdzają wyższą skuteczność modeli nowszej generacji, zwłaszcza w zadaniach wymagających rozpoznawania kontekstu. Wśród nich BERT często wykazuje się najwyższą skutecznością, jednak kosztem większego zapotrzebowania na zasoby obliczeniowe i czas treningu.

Podczas analizy dostępnych badań zauważono jednak kilka luk badawczych. Po pierwsze, nie zidentyfikowano pracy, która w sposób systematyczny i kolektywny porównywałaby modele NLP od klasycznych metod, przez sieci rekurencyjne, aż po architektury transformatorowe – a jednocześnie uwzględniała wpływ uczenia wielozadaniowego na wydajność modeli.

Wiele omawianych badań zakładało uprzednią inżynierię cech, rozbudowany preprocessing danych lub optymalizację hiperparametrów. Choć takie zabiegi poprawiają dokładność, mogą jednocześnie utrudniać porównanie natywnej skuteczności algorytmów. W niniejszej pracy celowo ograniczono ingerencję w dane wejściowe, aby stworzyć możliwie najbardziej neutralne środowisko testowe, pozwalające na ocenę modeli w ich podstawowej formie.

Dodatkowo przeprowadzona zostanie analiza czasu trenowania poszczególnych modeli. Choć skuteczność predykcyjna jest zwykle głównym kryterium porównań, to właśnie koszt obliczeniowy decyduje często o praktycznej przydatności danego rozwiązania, zwłaszcza w systemach produkcyjnych o ograniczonych zasobach. Aspekt ten bywa pomijany lub traktowany marginalnie w wielu pracach.

Rozdział 3

Zbiory danych i przetwarzanie tekstu

W niniejszej pracy do analizy sentymentu oraz emocji wykorzystano kilka ogólnodostępnych zbiorów danych pochodzących z różnych źródeł. Każdy zbiór został wybrany pod kątem liczebności, jakości danych oraz dostępności licencyjnej umożliwiającej przetwarzanie i analizę. Wszystkie dane dotyczą wypowiedzi użytkowników opublikowanych na różnych platformach społecznościowych, głównie na X (dawniej Twitterze). Cechują się one często emocjonalnym charakterem, typowym dla komentarzy internetowych. Zbiory miały docelowo liczyć około 100 tys. wpisów, co stanowiło kompromis pomiędzy wielkością zbioru a dostępnością zasobów obliczeniowych.

3.1. Opis zbiorów danych do analizy sentymentu

Celem przygotowania danych do zadania klasyfikacji sentymentu było uzyskanie korpusu zawierającego około 100 tys. wpisów. Aby osiągnąć ten cel, połączono kilka dostępnych publicznie zbiorów danych w jeden spójny zestaw. Po integracji dane zostały znormalizowane w zbiór trójklasowy o następujących etykietach:

- 0 – negatywny,
- 1 – neutralny,
- 2 – pozytywny.

W ramach tego procesu wykorzystano trzy zbiory:

3.1.1. Twitter Tweets Sentiment Dataset

Pierwszy wykorzystany zbiór to „*Twitter Tweets Sentiment Dataset*”, dostępny na platformie Kaggle [21]. Zbiór ten składa się z 27 480 tweetów, z których każdy został oznaczony etykietą: positive, neutral lub negative. Etykiety te zostały przemapowane na wartości liczbowe 2, 1 oraz 0 odpowiednio.

Licencja tego zbioru, to CC0 (ang. *Public Domain*), co umożliwia jego swobodne wykorzystanie, również do celów komercyjnych. Wstępna analiza wykazała obecność jednej pustej wartości, która została usunięta. Poza tym zbiór nie zawierał duplikatów.

3.1.2. Multiclass Sentiment Analysis Dataset

Drugim zbiorem był „*Multiclass Sentiment Analysis Dataset*”, opublikowany na platformie Hugging Face [22]. Zawiera on 41 643 przykłady tekstowe, oznaczone sentymentem jako positive, negative lub neutral, z dodatkowym polem label wskazującym już odpowiednie etykiety numeryczne 2, 0 i 1.

Zbiór udostępniony jest na licencji Apache 2.0, umożliwiającej zarówno modyfikacje, jak i komercyjne wykorzystanie pod warunkiem zachowania atrybucji. Zbiór nie zawierał duplikatów,

lecz podobnie jak poprzedni, zawierał jedną pustą wartość, którą usunięto podczas czyszczenia danych.

3.1.3. TweetEval

Trzecim i zarazem największym wykorzystanym zbiorem danych był „*TweetEval*”, również dostępny na platformie Hugging Face [23]. Spośród siedmiu dostępnych podzbiorów wybrano zbiór „*sentiment analysis*”, obejmujący 59 899 tweetów. Według źródła oznakowanie danych odpowiada: 0 (negatywne), 1 (neutralne) i 2 (pozytywne), co umożliwiło bezpośrednie włączenie danych do docelowego zbioru.

TweetEval dostarczany jest na licencji CC BY 3.0, co oznacza konieczność podania źródła i autorstwa. Wersja zbioru dotycząca sentymentu bazuje na danych opublikowanych przez Sarę Rosenthal, Nourę Farrę i Preslavę Nakovę w pracy „*Proceedings of the 11th international workshop on semantic evaluation (2017)*”. Podczas wstępного przetwarzania danych nie wykryto brakujących wartości, natomiast wykryto 26 duplikatów, które zostały usunięte.

3.1.4. Łączenie zbiorów danych

Po wstępny przeglądzie danych i ujednoliceniu etykiet klasowych, trzy opisane powyżej zbiory danych zostały połączone w jeden korpus. Aby zapewnić losowe rozmieszczenie przykładów pochodzących z różnych źródeł oraz uniknąć potencjalnych zależności wynikających z kolejności danych, cały korpus został dodatkowo przemieszany z ustalonym ziarnem.

W wyniku połączenia powstał zestaw zawierający niespełna 129 tys. rekordów tekstowych. Dane nie zawierały brakujących wartości, natomiast dalsza analiza wykazała obecność 27 tys. duplikatów, które zostały usunięte. Ostateczny rozmiar zbioru danych po połączeniu wyniósł 101 515 unikalnych wpisów.

3.2. Opis zbioru danych do analizy emocji

Zbiór danych wykorzystany do zadania klasyfikacji emocji to „*Emotion Dataset*”, opublikowany na platformie Kaggle [24]. Zbiór ten zawiera około 400 tys. tweetów, z których każdy został przyporządkowany do jednej z sześciu klas emocji. Rozkład danych pomiędzy klasami jest jednak silnie niezrównoważony – najmniej liczna klasa zawiera w przybliżeniu 15 tys. wpisów.

Zgodnie z dokumentacją zbioru, przyporządkowanie etykiet klas prezentuje się następująco:

- 0 – smutek,
- 1 – radość,
- 2 – miłość,
- 3 – gniew,
- 4 – strach,
- 5 – zaskoczenie,

Dane zostały udostępnione na licencji MIT, która umożliwia ich dowolne wykorzystywanie, modyfikowanie oraz rozpowszechnianie, również w celach komercyjnych, przy zachowaniu informacji o autorach. Zbiór nie zawierał pustych rekordów, natomiast zostało usuniętych 686 duplikatów.

3.3. Czyszczenie i przygotowanie zbiorów

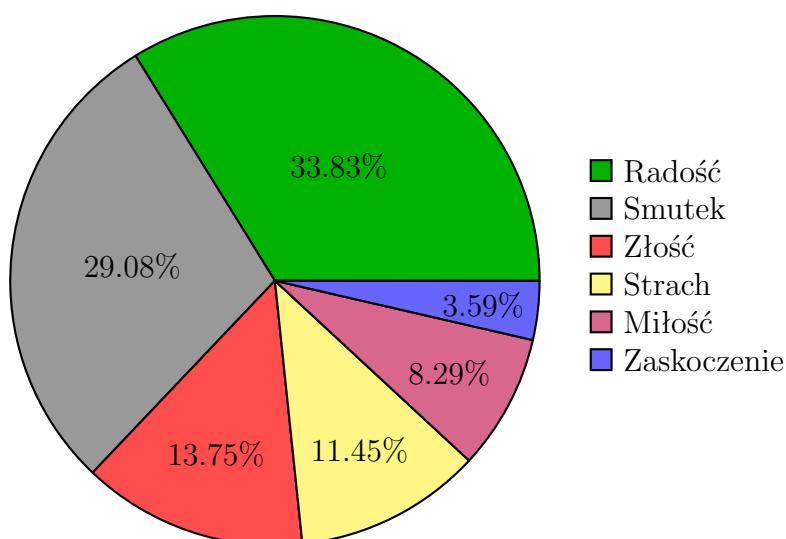
W celu zapewnienia sprawiedliwego porównania skuteczności różnych modeli klasyfikacyjnych, zarówno dane dotyczące sentymetru, jak i emocji zostały poddane jednakowemu, umiarkowanemu czyszczeniu. Celem było zminimalizowanie wpływu intensywnego przetwarzania wstępnego na jakość predykcji, koncentrując się na zdolnościach klasyfikacyjnych poszczególnych architektur. W szczególności zrezygnowano z lematyzacji, czyli procesu sprowadzania wyrazów do ich form podstawowych (np. „*miałem*” → „*mieć*”), która może mieć znaczenie w modelach opartych na tradycyjnych reprezentacjach tekstu. W przypadku modelu BERT takie zabiegi nie są konieczne, ponieważ zastosowany w nim tokenizator oparty na podwyrazach (ang. *WordPiece*) samodzielnie radzi sobie z różnorodnymi formami, ucząc się reprezentacji słów w ich rzeczywistym kontekście.

Zgodnie z ustaleniami zaprezentowanymi w literaturze, współczesne modele NLP potrafią efektywnie wykorzystywać elementy tekstu, takie jak interpunkcja czy niestandardowa pisownia, dlatego nadmierna ingerencja w strukturę tekstu może prowadzić do utraty cennych informacji kontekstowych. Badania Kurniasih i Manika [25] wskazują, że intensywne czyszczenie tekstu nie przynosi istotnych korzyści przy wykorzystaniu modeli takich jak BERT czy sieci LSTM. Ponadto Tan i współautorzy [26] dowodzą, że modele potrafią efektywnie wykorzystywać kontekst zawarty w nieformalnym i częściowo nieuporządkowanym tekście.

Proces czyszczenia ograniczał się więc do kilku prostych operacji:

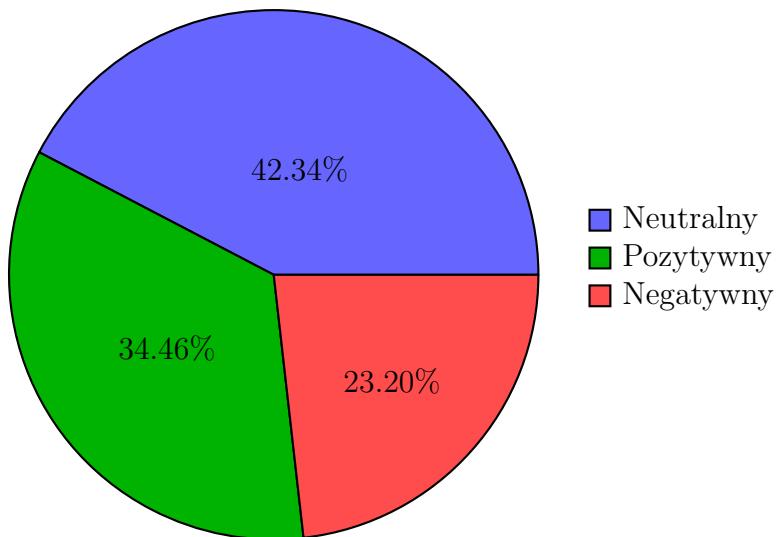
- usunięcia linków (np. <http://...>, <www...>),
- usunięcia oznaczeń użytkowników (np. @username),
- zachowania znaków interpunkcyjnych (., !?) jako potencjalnych nośników emocji,
- usunięcia pozostałych znaków specjalnych i nadmiarowych białych znaków,
- usunięcia bardzo krótkich wpisów (≤ 3 znaki), które uznano za szum informacyjny.

W wyniku zastosowanego czyszczenia oba zbiory danych – dotyczące sentymetru oraz emocji – uzyskały następujące rozkłady klas:



Rysunek 3.1: Rozkład procentowy klas w zbiorze do klasyfikacji emocji. Dane własne.

Zbiór danych przeznaczony do klasyfikacji emocji zawierał końcowo 416 120 wpisów, nie zawierając przy tym duplikatów czy wartości pustych.



Rysunek 3.2: Rozkład procentowy klas w zbiorze do klasyfikacji sentymentu. Dane własne.

Zbiór danych przeznaczony do klasyfikacji sentymentu zawierał 101 419 wpisów. Po czyszczeniu zaistniała potrzeba usunięcia 150 zduplikowanych rekordów. Finalnie pozostało 101 269 unikalnych przykładów.

Na tym etapie przetwarzanie danych zostało zakończone, a wszelkie dalsze operacje, takie jak tokenizacja czy dodanie specjalnych tokenów (np. *[CLS]*, *[SEP]*) wymaganych przez konkretne modele, zostaną przeprowadzone już podczas etapu treningu.

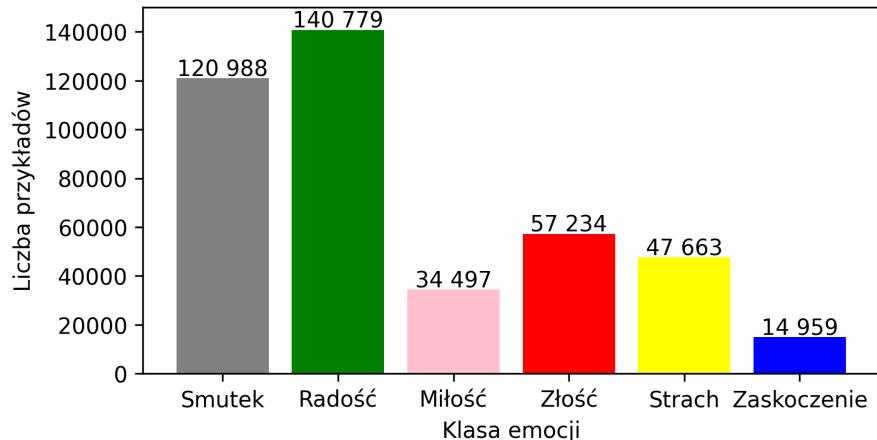
3.4. Statystyczna charakterystyka danych po czyszczeniu

W celu dokładniejszej charakterystyki danych użytych do trenowania modeli przeprowadzono dalszą analizę statystyczną. Analiza skupiła się na rozkładzie klas (w celu oceny nierówności rozkładu w zbiorze i podjęcia działań wyrównujących), długości tekstu oraz częstości występowania słów. Pozwoliło to na identyfikację ekstremalnych wartości, które mogłyby zaburzyć proces uczenia modeli. Jak piszą autorzy pracy [27], niezrównoważone zbiory mogą prowadzić do przetrenowania modeli względem klas dominujących. W takich przypadkach zaleca się zastosowanie technik przeciwdziałających temu zjawisku, takich jak ważenie klas (ang. *class weighting*), nadpróbkowanie lub podpróbkowanie.

W niniejszej pracy zdecydowano się na zastosowanie mechanizmu ważenia klas, co umożliwia algorytmom lepsze dopasowanie się do mniej licznych klas. Dodatkowo ustalono maksymalną długość tekstu na poziomie 256 znaków, co wynikało z ograniczeń w zasobach obliczeniowych.

3.4.1. Zbiór do analizy emocji

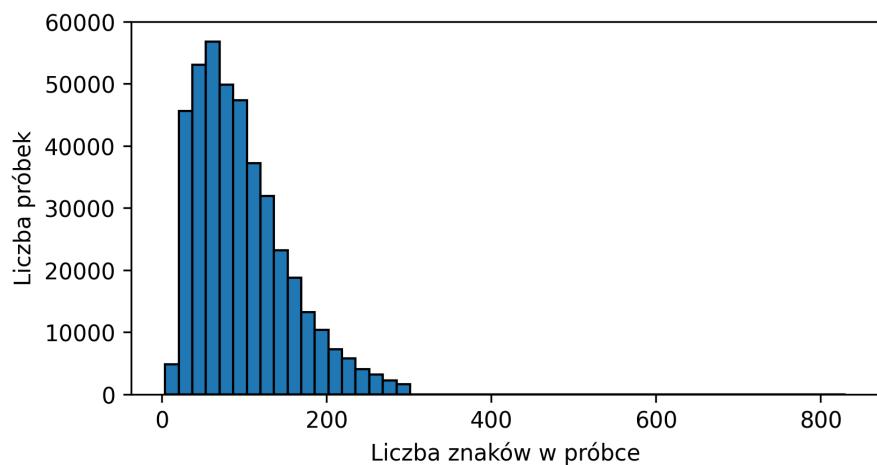
Rozkład klas



Rysunek 3.3: Rozkład liczebności poszczególnych klas w zbiorze danych do analizy emocji.
Dane własne.

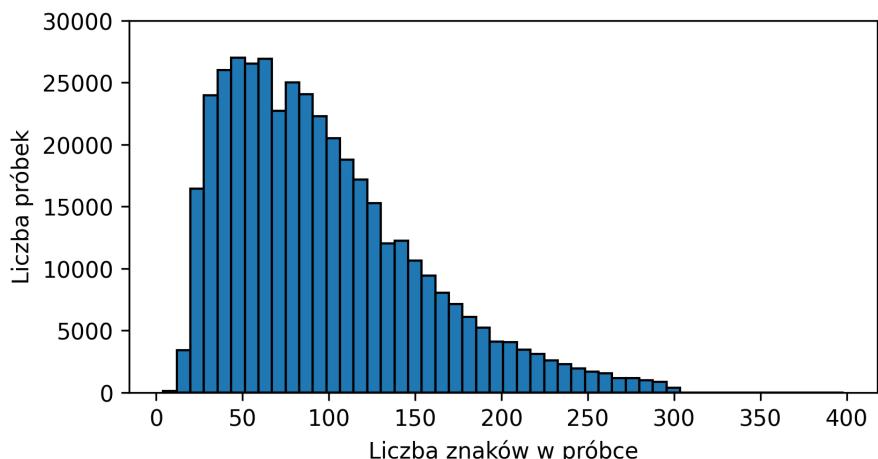
Na podstawie analizy rozkładu klas (Rysunek 3.3) widać wyraźnie, jak nierównomiernie rozkładają się przykłady przypisane do poszczególnych emocji. W celu uzyskania zbioru równomiernego pod względem klas i zgodnego z założoną, dla obu zbiorów, liczbą 100 tysięcy przykładów, z każdej klasy wylosowano maksymalnie 16 667 przykładów. Jedynym odstępstwem od tej zasady była klasa „Zaskoczenie”, której rozmiar nie pozwalał na osiągnięcie docelowej liczby próbek. W związku z tym zdecydowano się również na zastosowanie ważenia klas podczas trenowania modeli.

Analiza długości tekstów



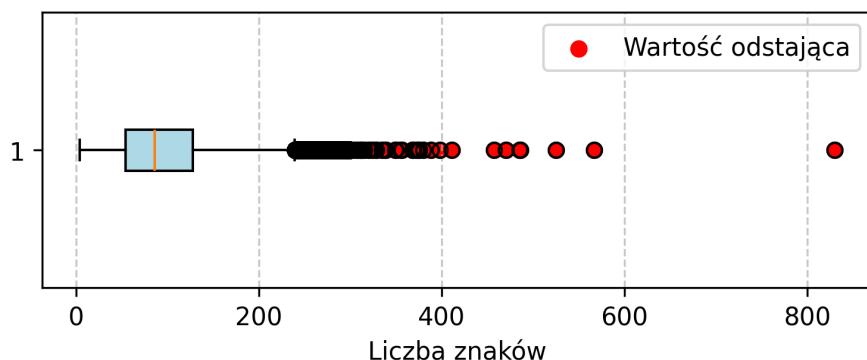
Rysunek 3.4: Rozkład długości tekstów w zbiorze danych do analizy emocji. Dane własne.

Powyższy histogram (Rysunek 3.4) ukazuje, że znaczna część danych koncentruje się w przedziale 50–150 znaków, natomiast najdłuższe wpisy przekraczały 800 znaków. Takie dane odstające mogłyby negatywnie wpływać na stabilność trenowania modeli.



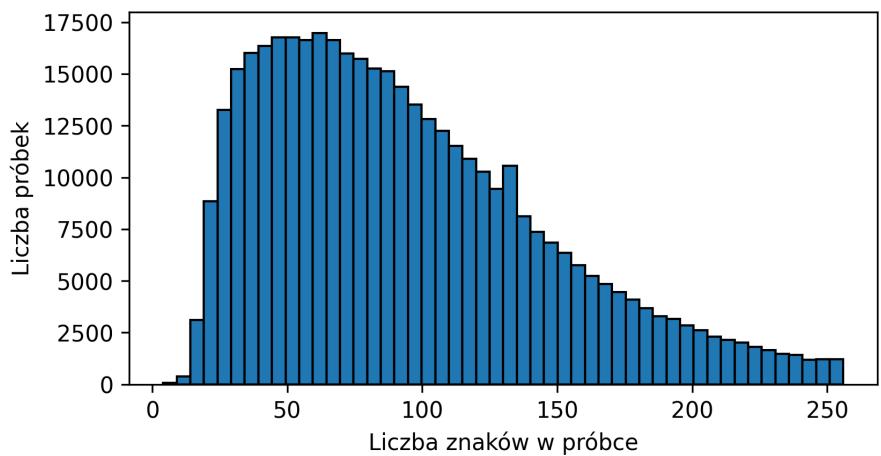
Rysunek 3.5: Rozkład długości tekstów w zbiorze danych do analizy emocji o długości poniżej 400 znaków. Dane własne.

Drugi histogram, odfiltrowujący dane o długości równej bądź większej niż 400 znaków (Rysunek 3.5), pozwolił na dokładniejsze uwidocznienie struktury rozkładu. Większość wartości mieści się w przedziale od 30 do 120 znaków, jednak pewna część danych rozkłada się również w zakresie od 250 do około 310 znaków.



Rysunek 3.6: Rozkład liczby znaków w próbkach w zbiorze do analizy emocji. Dane własne.

Na wykresie pudełkowym (Rysunek 3.6) widać, że wszystkie wartości powyżej 239 znaków można uznać za wartości odstające (outliers). Gdyby wszystkie te próbki były potraktowane jako outliers, ich liczba wynosiłaby 9 951. Zdecydowano się jednak odciąć jedynie próbki, których długość jest większa niż 256.



Rysunek 3.7: Rozkład długości tekstów w zbiorze danych do analizy emocji po usunięciu wartości odstających. Dane własne.

Po odcięciu 4 922 rekordów, pozostało 411 198 próbek. Jak wspomniano wcześniej, największe zagęszczenie danych jest wyraźnie widoczne w przedziale od około 30 do 120 znaków.

Analiza częstości występowania słów

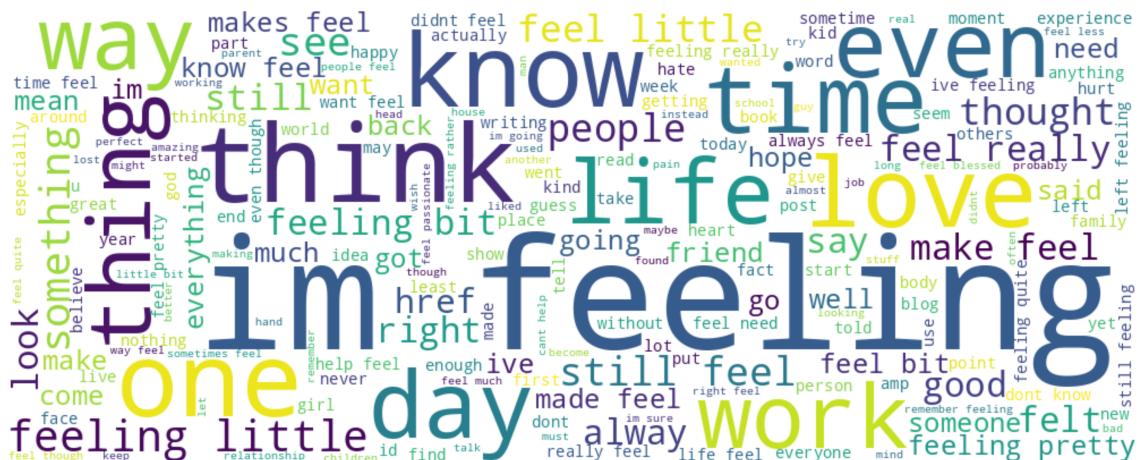
Przeprowadzono również analizę najczęściej występujących słów w korpusie, której wyniki zostały zwizualizowane za pomocą chmury słów (ang. *WordCloud*). W pierwszym etapie obliczono najczęściej pojawiające się słowa w całym zbiorze danych. Wśród nich znalazły się kolejno takie wyrazy jak: „*i*”, „*feel*”, „*and*”, „*to*”, „*the*”, „*a*”, „*feeling*”, „*that*”, „*of*” oraz „*my*”. Dominacja słów takich jak „*i*” (ja) oraz „*feel*” (czuje) sugeruje, że zbiór danych koncentruje się na odczuciach, co może wpływać na wyraźne rozróżnienie emocji przez modele klasyfikacyjne.



Rysunek 3.8: Chmura słów dla zbioru do analizy emocji. Dane własne.

Na chmurze słów dominują wyrazy związane z emocjami, takie jak „feel”, „feeling”, „love”, „life”, „happy”, co również potwierdza sugestję, że dane koncentrują się na odczuciach i emocjach.

W kolejnym kroku analizy z danych usunięto słowa funkcjonalne (ang. *stop words*). Po tym zabiegu, najczęściej występującymi słowami były kolejno: „feel”, „feeling”, „like”, „im”, „really”, „know”, „time”, „get”, „little” oraz „people”. Na pierwszy plan wysunęły się słowa takie jak, „feeling”, „like”, „little” czy „really”. „Feeling” występuje bezpośrednio po „feel”, a „like” oraz wzmacniające przekaz „really” i „little” także odgrywają istotną rolę, co znów sugeruje, że dane mogą koncentrować się na konkretnych emocjach.



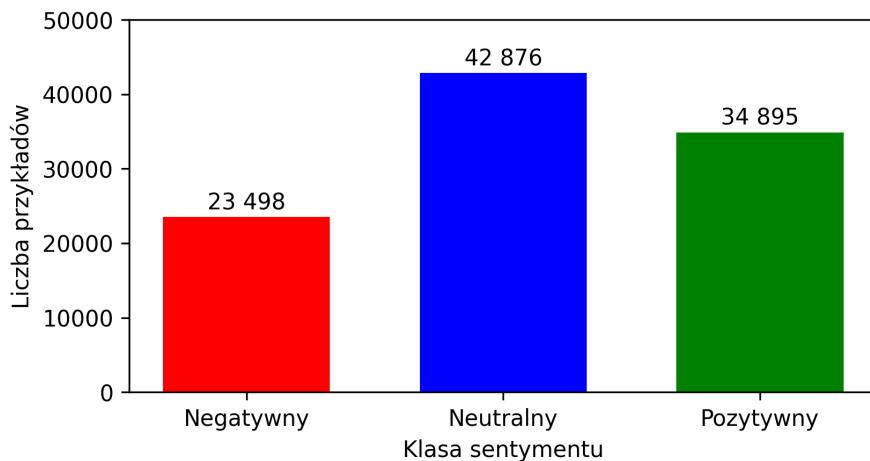
Rysunek 3.9: Chmura słów dla zbioru do analizy emocji po usunięciu słów funkcjonalnych.
Dane własne.

Mimo, że po usunięciu *stop words* na wizualizacji pozostały słowa związane z emocjami, takie jak „*love*”, ogólny obraz sugeruje, że zbiór może obejmować szeroką gamę słów związanych z emocjami, a nie koncentrować się na jednym dominującym zwrocie.

Pomimo przeprowadzonego czyszczenia, w dalszej analizie zachowano *stop words* w danych wykorzystywanych do treningu modeli. Jak wspomniano wcześniej, modele NLP, takie jak BERT czy LSTM, skutecznie wykorzystują kontekst zawarty w pisowni oraz elementach gramatycznych, dlatego nadmierne filtrowanie danych tekstowych nie jest konieczne.

3.4.2. Zbiór do analizy sentymenu

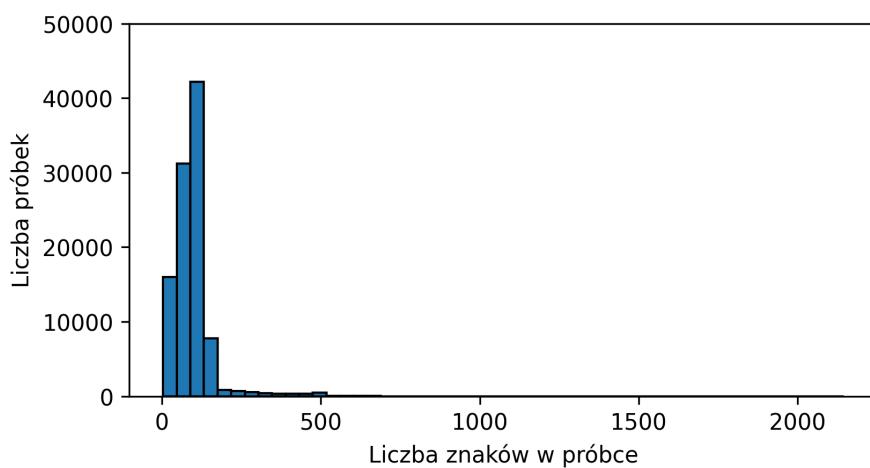
Rozkład klas



Rysunek 3.10: Rozkład liczebności poszczególnych klas w zbiorze danych do analizy sentymenu. Dane własne.

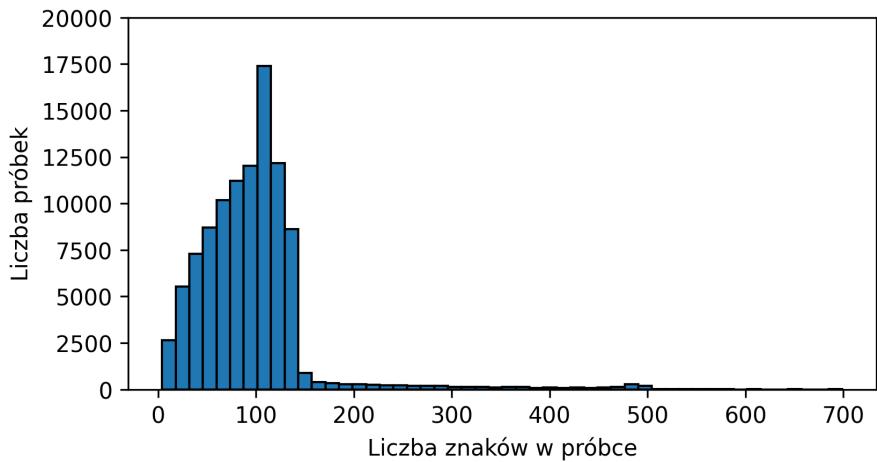
W zbiorze do analizy sentymenu również występuje znacząca dysproporcja klas, dlatego też analogicznie do zbioru związanego z emocjami, w celu zminimalizowania ryzyka przetrenowania względem klas dominujących, zastosowano ważenie klas dla modeli BERT, GRU, LSTM i SVM.

Analiza długości tekstów



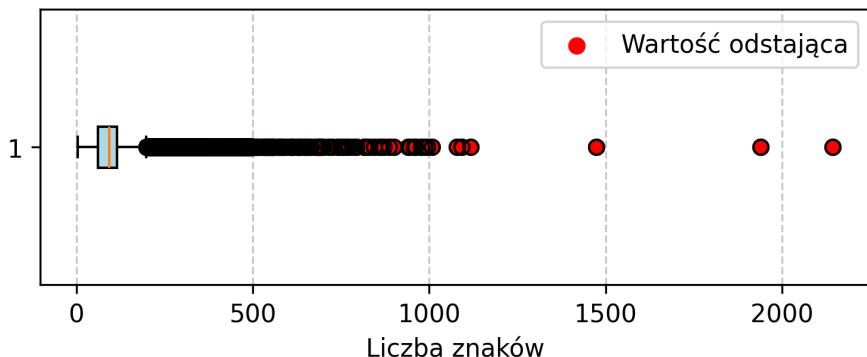
Rysunek 3.11: Rozkład długości tekstów w zbiorze danych do analizy sentymenu. Dane własne.

Histogram długości tekstów (Rysunek 3.11) pokazał znaczną rozpiętość – większość przykładów zawiera od 100 do 200 znaków, jednak najdłuższe wpisy przekraczały 2 000 znaków.



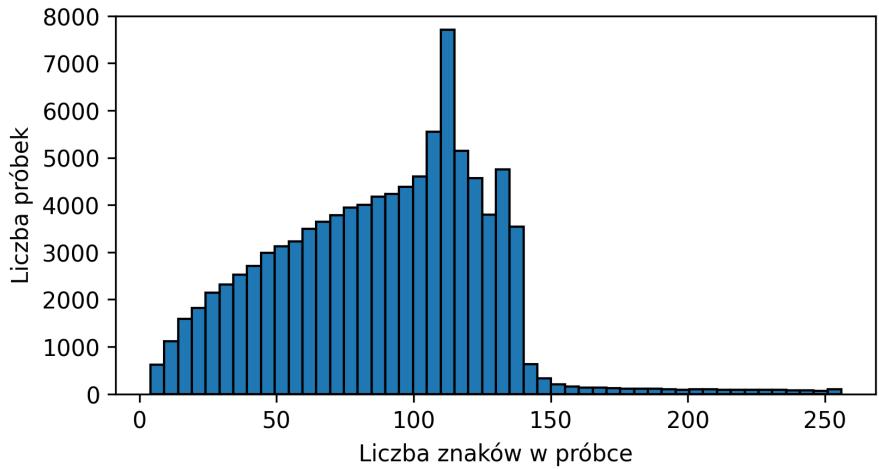
Rysunek 3.12: Rozkład długości tekstów w zbiorze danych do analizy sentymenu o długości poniżej 700 znaków. Dane własne.

Histogram ukazujący dane zawierające mniej niż 700 znaków (Rysunek 3.12) pozwolił na dokładniejsze przedstawienie gęstości rozkładu. Większość wartości znajduje się w okolicach 50–130 znaków, podczas gdy zauważalna część danych rozkłada się również w zakresie od 200 do 500 znaków.



Rysunek 3.13: Rozkład liczby znaków w próbkach w zbiorze do analizy sentymenu. Dane własne.

Na wykresie pudełkowym (Rysunek 3.13) widać, że próbki o długości powyżej 199 znaków można uznać za wartości odstające. Ich liczba wynosi 3 596. Jak wspomniano wcześniej, zdecydowano się odciąć jedynie dane o długości większej niż 256 znaków.

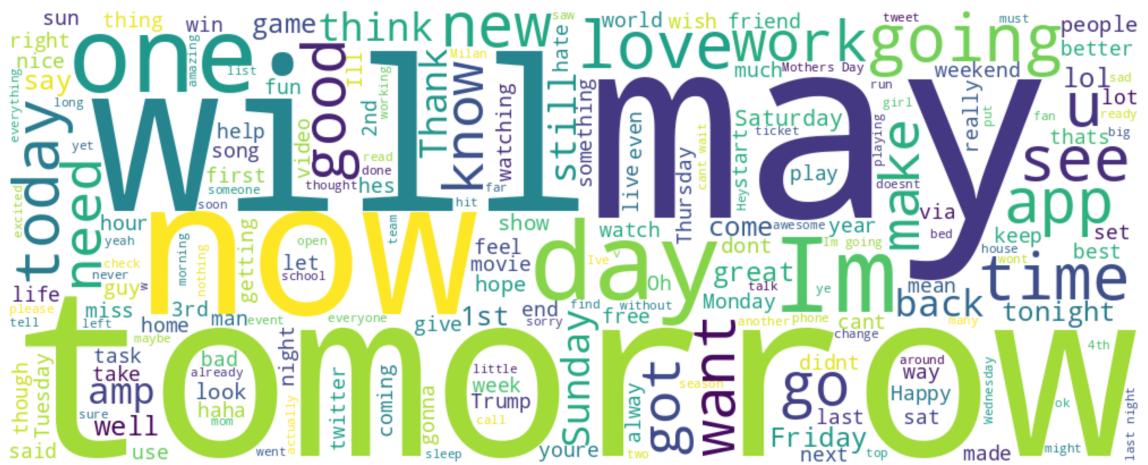


Rysunek 3.14: Rozkład długości tekstów w zbiorze danych do analizy sentymentu po usunięciu wartości odstających. Dane własne.

Po tym zabiegu rozkład długości tekstów na histogramie (Rysunek 3.14) ujawnia wyraźne zgaszczenie w przedziale 50–140 znaków, co jest typowe dla komentarzy internetowych. W wyniku przetwarzania usunięto 2 590 rekordów, pozostawiając 98 679 przykładów, co jest zbliżone do planowanej liczby stu tysięcy.

Analiza częstości występowania słów

Najczęściej pojawiające się słowa w zbiorze dla sentymentu to kolejno: „the”, „to”, „I”, „a”, „and”, „in”, „of”, „on”, „is” oraz „for”. Dominują tu słowa funkcjonalne.



Rysunek 3.15: Chmura słów dla zbioru do analizy sentymentu. Dane własne.

Na chmurze słów dominują wyrazy takie jak „tomorrow”, „day”, „may”, „will” i „now”, ale pojawiły się także słowa, które niosą pewną wartość wiłączaną z sentymentem, takie jak „best”, „good”, „awesome” czy „love”. Choć bez kontekstu trudno jednoznacznie określić, jak są nacechowane wypowiedzi zawierające te słowa, to jednak ich obecność wskazuje na możliwą emocjonalną treść.

Po usunięciu *stop words* najczęściej występującymi słowami były kolejno: „*may*”, „*I'm*”, „*like*”, „*tomorrow*”, „*get*”, „*going*”, „*see*”, „*day*”, „*time*” oraz „*go*”.



Rysunek 3.16: Chmura słów dla zbioru do analizy sentymentu po usunięciu słów funkcjonalnych. Dane własne.

Wyniki pozostały zbliżone do poprzednich. Podobnie jak w przypadku zbioru do analizy emocji, w zbiorze do analizy sentymentu zdecydowano się również na zachowanie *stop words* w danych wykorzystywanych do treningu.

3.5. Podział danych na zestawy treningowe, walidacyjne i testowe

W celu zapewnienia rzetelnej oceny skuteczności zastosowanych modeli klasyfikacyjnych, a także monitorowania ich procesu uczenia, oba zbiory danych zostały podzielone na trzy niezależne zestawy: treningowy, walidacyjny oraz testowy.

Zbiór danych do analizy sentymentu został podzielony w proporcji 70% / 15% / 15%, odpowiednio dla zestawu treningowego, walidacyjnego i testowego. Podział został wykonany w sposób losowy, przy zastosowaniu ustalonego ziarna losowości. Aby zagwarantować reprezentatywność każdego z podzbiorów, zastosowano stratyfikację względem etykiet klas – oznacza to, że proporcje klas (negatywna, neutralna, pozytywna) w każdym z podzbiorów odpowiadają proporcjom występującym w całym zbiorze. Wszystkie uzyskane podzbiory zostały zapisane i w dalszych etapach eksperymentów wykorzystano je niezmiennie dla każdego z modeli klasyfikacyjnych analizujących sentyment.

Analogiczne podejście zastosowano przy przygotowywaniu danych dla zadania klasyfikacji emocji, przy czym dodatkowo, przed dokonaniem podziału, zbiór został zrównoważony pod względem liczbowości klas. Zastosowano próbkowanie z zachowaniem równych udziałów, tzn. dla każdej z klas (z wyjątkiem najmniej licznej – „zaskoczenie”) losowo wybrano po 16 667 przykładów ($16\,667 \times 6 \approx 100\,000$).

W niniejszej pracy nie zastosowano walidacji krzyżowej, co było celowym wyborem metodologicznym wynikającym z potrzeby zapewnienia spójności porównania pomiędzy różnymi modelami, które znaczco różnią się pod względem złożoności oraz czasu trenowania, zwłaszcza w przypadku modeli transformatorowych, takich jak BERT, dla których proces treningu jest czasochłonny. Wykorzystany stały, stratyfikowany podział na zbiorę treningową, walidacyjną i testową umożliwił jednoznaczną i powtarzalną ocenę skuteczności modeli. Pozwoliło to na porównanie modeli w ich podstawowej konfiguracji, bez wprowadzania dodatkowej optymalizacji hiperparametrów, ponieważ celem badania nie było maksymalne dostosowanie każdego modelu indywidualnie.

Rozdział 4

Modele i metody treningu

W tym rozdziale przedstawione zostaną opisy modeli wykorzystanych w pracy, uwzględniając ich architekturę, cechy charakterystyczne oraz sposób treningu. Celem jest omówienie procesu uczenia tych modeli, a także przedstawienie parametrów, które zostały użyte podczas trenowania. W tym przypadku starano się zachować jak najmniejszą ingerencję w ustawienia modeli, by przeprowadzić porównanie samych algorytmów, a nie ich dostosowanych wersji, które mogłyby wymagać indywidualnego dostrajania hiperparametrów dla każdego modelu. Modele były trenowane w ten sam sposób zarówno dla klasyfikacji sentymentu, jak i emocji, dlatego opisy dotyczące procesów treningu nie będą powtarzane dla każdej z tych kategorii osobno.

4.1. Model SVM

Głównym celem maszyny wektorów nośnych jest znalezienie hiperpowierzchni (w przestrzeni cech), która maksymalizuje margines pomiędzy różnymi klasami. W klasyfikacji binarnej SVM stara się znaleźć prostą, która najlepiej separuje dwie klasy, zachowując jak największy odstęp (margines) od punktów należących do tych klas [28].

W klasyfikacji wieloklasowej, jak ma to miejsce w analizie sentymentu i emocji, gdzie w tym przypadku najmniejsza liczba klas wynosi trzy, algorytm SVM stosuje podejście rozszerzające model do obsługi większej liczby etykiet. Zamiast klasycznego podejścia binarnego, wykorzystuje się różne strategie. W pracy tej zastosowano metodę „*jeden kontra jeden*” (ang. *One-vs-One*, OvO). W tej metodzie dla każdej pary klas tworzony jest osobny klasyfikator, a każdy z tych klasyfikatorów wyznacza hiperpłaszczyznę, która oddziela te dwie klasy.

Klasyfikując sentyment, szczególnie ważnym elementem jest przekształcenie danych tekstowych do postaci, którą model może analizować w przestrzeni cech. Tekst, który jest z natury sekwencyjny i nieliczbowy, musi zostać odpowiednio przetworzony, aby mógł być użyty w algorytmie. W tym celu stosuje się odpowiednie techniki reprezentacji tekstu, jak chociażby TF-IDF, o którym mowa poniżej.

4.1.1. Zastosowana implementacja i parametry

W pracy wykorzystano implementację modelu SVM z biblioteki scikit-learn [29]. Implementacja ta została użyta z domyślnymi parametrami, z wyjątkiem kilku zmian, które zostały opisane poniżej.

Do reprezentacji tekstu w przestrzeni cech zastosowano TF-IDF (*Term Frequency-Inverse Document Frequency*), który przekształca słowa w wektory, uwzględniając częstotliwość występowania słów w danej próbce oraz ich istotność w kontekście całego zbioru danych. W procesie tokenizacji wybrano 20 000 najczęściej występujących słów w zbiorze, które zostały potraktowane jako cechy wejściowe dla modelu. Dzięki temu słowa w zbiorze tekstów są reprezentowane jako liczby, które model SVM wykorzystuje do klasyfikacji.

Model SVM został użyty z jądrem liniowym. Jest to najprostszy kernel, który jest także wydajny w klasyfikacji tekstu. W badaniu [30] stwierdzono, że jądro liniowe osiąga najwyższą dokładność klasyfikacji w najkrótszym czasie.

Zastosowano ważenie klas, które miało na celu zrównoważenie nierównomiernego rozkładu danych, omówionego w poprzednim rozdziale. Dzięki temu model miał szansę lepiej poradzić sobie z klasami występującymi rzadziej, minimalizując dominację bardziej reprezentowanych klas w zbiorze treningowym.

Parametr C, ustawiony został na wartość 1.0. Pełni on rolę regularyzacji, balansując pomiędzy dopasowaniem modelu do danych a jego zdolnością do generalizacji. Wyższe wartości C zmniejszają margines błędu, co może prowadzić do przeuczenia, natomiast niższe wartości zwiększą tolerancję na błąd, sprzyjając lepszemu uogólnianiu [28].

4.1.2. Proces treningu

W standardowej implementacji w bibliotece scikit-learn algorytm SVM jest zoptymalizowany do pracy na CPU, dlatego model nie był trenowany na GPU. Czas trenowania wyniósł około 5–7 minut zarówno dla klasyfikacji sentymentu, jak i emocji. Czasy treningu przedstawione w tej pracy służą głównie do porównań między modelami, a nie do wyciągania dalszych wniosków.

W analizie sentymentu model osiągnął średnią dokładność 66% na zbiorze testowym i 65% na walidacyjnym, co wskazuje na dobre dopasowanie, ponieważ wyniki są porównywalne, sugerując brak przetrenowania i niedotrenowania. Niska dokładność może wynikać z samej natury zadania oraz złożoności tekstów.

W klasyfikacji emocji model uzyskał dokładność 91% zarówno na zbiorze testowym, jak i walidacyjnym, co świadczy o dobrym dopasowaniu i wysokiej skuteczności, a także sugeruje, że model dobrze generalizuje lub teksty były wyraźnie zróżnicowane.

4.2. Model LSTM

Long Short-Term Memory (LSTM) to rodzaj rekurencyjnej sieci neuronowej (RNN), który został zaprojektowany do rozwiązania problemu zanikania gradientu, występującego w klasycznych RNN podczas trenowania na długich sekwencjach. Dzięki swojej strukturze, LSTM jest w stanie przechowywać istotne informacje przez dłuższy czas, co sprawia, że świetnie nadaje się do analizy sekwencji, takich jak teksty. W LSTM wykorzystywane są trzy bramki: wejściowa, zapomnienia oraz wyjściowa, które pozwalają modelowi kontrolować przepływ informacji oraz zapamiętywać lub zapominać ważne lub nieistotne dane w zależności od kontekstu. Dzięki tym mechanizmom LSTM jest w stanie uchwycić długoterminowe zależności w danych, co czyni go odpowiednim narzędziem w analizie tekstów [31].

4.2.1. Zastosowana implementacja i parametry

Implementacja modelu LSTM w tej pracy opiera się na bibliotece PyTorch, która oferuje dużą elastyczność w implementacji sieci neuronowych oraz zapewnia dostęp do narzędzi do ich treningu i ewaluacji [32]. W tej pracy użyto modelu LSTM z 256 jednostkami w warstwie ukrytej, z dodatkową warstwą osadzającą (ang. *embedding layer*), która przekształca tokeny na wektory o wymiarze 256.

Do tokenizacji tekstów użyto narzędzi z biblioteki Keras [33]. Proces tokenizacji polega na przekształceniu tekstów w sekwencje liczb (tokeny), które mogą być użyte w modelu. W przypadku tej implementacji podobnie jak w przypadku modelu SVM również wybierano 20 000 najczęściej występujących słów w zbiorze danych.

Początkowa implementacja modelu, zarówno dla klasyfikacji emocji, jak i sentymetu, wykazała problem z nadmiernym dopasowaniem do tylko i wyłącznie jednej klasy. Aby poprawić wyniki, zastosowano warstwę dropout o współczynniku 0.5. Dropout jest techniką regularyzacji, która polega na losowym wyłączaniu części neuronów podczas treningu, co pozwala na zwiększenie ogólnej zdolności modelu do generalizacji na nowych danych. Dodatkowo, w celu poprawy efektywności przetwarzania tekstu, użyto dwukierunkowego LSTM (ang. *bidirectional LSTM*). Implementacja modelu LSTM objęła także wykorzystanie funkcji aktywacji ReLU (ang. *Rectified Linear Unit*) w warstwie wyjściowej.

Cała sieć została zoptymalizowana przy użyciu algorytmu Adam (ang. *Adaptive Moment Estimation*), który jest jedną z popularniejszych metod optymalizacji w uczeniu głębokich sieci neuronowych. W celu zrównoważenia nierównomiernego rozkładu danych w zbiorze treningowym wprowadzono wagi klasowe.

W przypadku wersji wielozadaniowej model był trenowany równolegle do obu zadań. Dwie oddzielne warstwy wyjściowe zostały użyte do klasyfikacji sentymetu i emocji.

4.2.2. Proces treningu

Model LSTM był trenowany przez dwie epoki, co zostało uznane za standard dla wszystkich modeli po testach na modelu BERT. Taka długość treningu okazała się wystarczająca do uzyskania stabilnych wyników oraz zapobiegła przeuczeniu. Czas treningu wynosił średnio 10 minut dla wersji jednozadaniowych, a dla wersji multitask około 20 minut. Modele były trenowane na GPU.

Epoka	Sentyment		Emocje	
	Dokł. (train)	Dokł (val)	Dokł (train)	Dokł (val)
1	55.27%	63.04%	67.27%	92.06%
2	68.24%	66.68%	93.34%	94.76%

Tabela 4.1: Porównanie dokładności podczas treningu modeli jednozadaniowych LSTM.
Dane własne.

W przypadku wersji jednozadaniowej, dla klasyfikacji sentymetu model uzyskał dokładność 67% na zbiorze testowym, 68% na zbiorze treningowym oraz 66.7% na zbiorze walidacyjnym. Z danych przedstawionych w tabeli powyżej widać, że model poprawiał wyniki na zbiorze treningowym i walidacyjnym w procesie uczenia.

W analizie emocji model osiągnął dokładność 96.7% na zbiorze testowym, 93% na zbiorze treningowym oraz niemal 95% na zbiorze walidacyjnym, co sugeruje skuteczną klasyfikację. W trakcie procesu treningowego również nie zauważono oznak przeuczenia.

Epoka	Sentyment		Emocje	
	Dokł. (train)	Dokł. (val)	Dokł. (train)	Dokł. (val)
1	54.11%	64.53%	71.31%	93.57%
2	66.93%	66.13%	93.92%	94.54%

Tabela 4.2: Porównanie dokładności podczas treningu modelu wielozadaniowego LSTM. Dane własne.

W przypadku modelu wielozadaniowego proces treningu przebiegał bardzo podobnie, osiągając niemal identyczne wyniki jak modele jednozadaniowe dla obu zadań.

4.3. Model GRU

Gated Recurrent Units (GRU) to kolejny typ rekurencyjnej sieci neuronowej, który, podobnie jak LSTM, został zaprojektowany w celu rozwiązywania problemu zanikania gradientu. GRU różni się od LSTM tym, że posiada jedynie dwie bramki: bramkę aktualizacji oraz bramkę zapomnienia. Bramka aktualizacji decyduje, jakie informacje mają zostać zachowane, a bramka zapomnienia określa, które informacje należy zapomnieć. Dzięki tej uproszczonej strukturze, GRU jest szybsze w trenowaniu i potrzebuje mniej zasobów obliczeniowych, jednocześnie osiągając zbliżoną skuteczność w porównaniu do LSTM w wielu zadaniach, w tym w analizie tekstu [34].

4.3.1. Zastosowana implementacja i parametry

W przypadku GRU implementacja również opiera się o bibliotekę PyTorch [32]. Implementacja jest bliźniaczo podobna: model wykorzystuje warstwę osadzającą, która przekształca tokeny na wektory o wymiarze 256, a następnie przechodzi przez warstwę GRU z 256 jednostkami ukrytymi. Proces tokenizacji przebiegał dokładnie tak samo jak w przypadku modelu LSTM. Użyto tokenizera z biblioteki Keras [33] i wybrano 20 000 najczęściej występujących słów w zbiorze.

Różnice między modelami LSTM i GRU pojawiają się w tym miejscu. Dla standardowych, powyższych ustawień model GRU osiągnął wyniki porównywalne do tych, które uzyskano po wprowadzeniu dodatkowych parametrów. Dodatkowe parametry zostały wprowadzone w celu zapewnienia sprawiedliwego porównania między modelami. Zastosowano więc również warstwę dropout o współczynniku 0.5. Użyto również dwukierunkowego GRU (ang. *bidirectional GRU*), który tak jak dla modelu LSTM umożliwia modelowi analizowanie tekstu zarówno od lewej do prawej, jak i od prawej do lewej. W warstwie wyjściowej zastosowano funkcję aktywacji ReLU, a optymalizacja modelu przebiegała z wykorzystaniem algorytmu Adam. Tak jak w przypadku LSTM, również w tym modelu użyto wag klasowych.

Wersja wielozadaniowa modelu GRU była trenowana paralelnie do obu zadań – klasyfikacji sentymentu oraz emocji. Do każdego z tych zadań przypisano oddzielną warstwę wyjściową.

4.3.2. Proces treningu

Trening modelu GRU obejmował dwie epoki, a czas treningu był bardzo zbliżony do modelu LSTM i również wynosił około 10 minut dla wersji jednozadaniowych oraz około 20 minut dla modelu wielozadaniowego. Trening odbywał się na procesorze graficznym.

Epoka	Sentyment		Emocje	
	Dokł. (train)	Dokł. (val)	Dokł. (train)	Dokł. (val)
1	55.20%	64.28%	76.35%	94.22%
2	67.95%	64.38%	93.87%	94.33%

Tabela 4.3: Porównanie dokładności podczas treningu modeli jednozadaniowych GRU. Dane własne.

Klasyfikując sentyment model uzyskał dokładność na poziomie 64% na zbiorze testowym, 68% na zbiorze treningowym oraz 64% na zbiorze walidacyjnym. Analizując wyniki na zbiorze walidacyjnym w trakcie procesu uczenia, można zauważyc, że dokładność stabilizuje się, co wskazuje na pewną stagnację w dalszym uczeniu modelu. Niemniej jednak, model nie wykazuje oznak przeuczenia, ponieważ dokładność na zbiorze walidacyjnym nie spadła, mimo wzrostu wydajności na zbiorze treningowym.

W analizie emocji, model osiągnął dokładność na poziomie 94% na zbiorze testowym, niemal 94% na zbiorze treningowym oraz 94% na zbiorze walidacyjnym. Również w tym przypadku wyniki są stabilne, ale model nie wykazuje oznak przeuczenia.

Epoka	Sentyment		Emocje	
	Dokł. (train)	Dokł. (val)	Dokł. (train)	Dokł. (val)
1	54.38%	63.41%	77.53%	94.12%
2	66.92%	64.98%	93.77%	94.36%

Tabela 4.4: Porównanie dokładności podczas treningu modelu wielozadaniowego GRU. Dane własne.

Proces treningu modelu wielozadaniowego przebiegał w sposób zbliżony do modeli jednozadaniowych, osiągając prawie takie same wyniki dla obu zadań.

4.4. Model BERT

Bidirectional Encoder Representations from Transformers (BERT) to najbardziej zaawansowany model użyty w tej pracy, a także jeden z najbardziej zaawansowanych modeli w dziedzinie przetwarzania języka naturalnego. Model ten bazuje na architekturze Transformer, która wykorzystuje mechanizm *self-attention* do przetwarzania sekwencji. Główna innowacja BERT-a polega na jego dwukierunkowym podejściu do analizy tekstu, co oznacza, że model uwzględnia kontekst zarówno poprzedzających, jak i następujących słów w danym zdaniu. Jednak w przeciwieństwie do LSTM czy GRU, które realizują dwukierunkowość poprzez analizowanie tekstu sekwencyjnie, BERT wykorzystuje wspomniany wcześniej mechanizm samouwagi, który pozwala mu jednocześnie analizować całą sekwencję, uwzględniając kontekst wszystkich słów w zdaniu. To umożliwia modelowi lepsze uchwycenie semantycznych relacji między słowami [9].

BERT jest wstępnie przetrenowany na ogromnych zbiorach danych, co pozwala na uzyskanie uniwersalnych reprezentacji słów, które mogą być później dostosowane do różnych zadań za pomocą dodatkowego etapu *fine-tuningu*. W praktyce oznacza to, że model BERT może być łatwo zaadoptowany do szerokiego zakresu zadań NLP, w tym właśnie analizy sentymentu czy rozpoznawania emocji.

4.4.1. Zastosowana implementacja i parametry

W tej pracy użyto implementacji modelu BERT dostępnej w bibliotece Hugging Face transformers [35]. Model oparty jest na wersji *bert-base-uncased*, która została wstępnie przetrenowana na dużych zbiorach danych i dostosowana do klasyfikacji tekstów. Użyto tokenizera BERT z tej samej biblioteki, a w procesie tokenizacji wszystkie słowa zostały przekształcone na tokeny, a teksty zostały przycięte lub uzupełnione do maksymalnej długości 256 tokenów.

Wszystkie modele, zarówno jednozadaniowe, jak i wielozadaniowe, używały tych samych parametrów, aby zapewnić sprawiedliwe porównanie. Model BERT w użytej wersji implementacji oparty jest na 12 ukrytych warstwach transformera. Długość sekwencji wejściowej została ustalona na 256 tokenów. Liczba epok wynosiła początkowo pięć, jednak po analizie wyników stwierdzono, że model zaczyna się przetrenowywać po drugiej epoce, dlatego ostatecznie zdecydowano się na dwie epoki. Optymalizacja modelu odbywała się za pomocą algorytmu AdamW, który jest dostosowaną wersją algorytmu Adam, przystosowaną do pracy z parametrami BERT-a. Funkcja strat to *Cross-Entropy Loss*, z wagami klasowymi, które zostały obliczone na podstawie nierównomiernego rozkładu klas w zbiorze treningowym.

Wersja wielozadaniowa modelu BERT wykorzystywała dwie oddzielne warstwy wyjściowe: jedną do klasyfikacji sentymentu i drugą do klasyfikacji emocji.

4.4.2. Proces treningu

Epoka	Sentyment		Emocje	
	Dokł. (train)	Dokł. (val)	Dokł. (train)	Dokł. (val)
1	70.77%	72.92%	91.48%	95.18%
2	79.13%	73.41%	94.94%	95.24%
3	87.32%	72.93%	95.10%	95.10%
4	92.89%	72.07%	95.32%	95.24%
5	95.35%	72.61%	95.43%	95.04%

Tabela 4.5: Porównanie dokładności podczas treningu na pięciu epokach modeli jednozadaniowych BERT. Dane własne.

Epoka	Sentyment		Emocje	
	Dokł. (train)	Dokł. (val)	Dokł. (train)	Dokł. (val)
1	69.85%	72.86%	91.19%	94.99%
2	77.60%	72.24%	94.85%	94.89%
3	85.17%	72.13%	95.17%	95.11%
4	90.85%	71.64%	95.24%	95.05%
5	93.71%	71.26%	95.36%	95.23%

Tabela 4.6: Porównanie dokładności podczas treningu na pięciu epokach modelu wielozadaniowego BERT. Dane własne.

W pierwszym podejściu model BERT był trenowany przez pięć epok, zarówno dla wersji jednozadaniowych, jak i wielozadaniowych. Po przeanalizowaniu wyników na zbiorze walidacyjnym zauważono, że po pewnym czasie dokładność modelu na zbiorze walidacyjnym zaczynała się stabilizować lub wręcz maleć. Takie zachowanie wskazywało na problem z przeuczeniem, szczególnie w przypadku wersji jednozadaniowej dla klasyfikacji sentymentu. Podobny trend zauważono w wersji wielozadaniowej, gdzie dokładność na zbiorze walidacyjnym również malała.

W związku z tym, zdecydowano, że liczba epok będzie ograniczona do dwóch. Takie podejście miało na celu uniknięcie przeuczenia. Zmniejszenie liczby epok okazało się skuteczne, a czas treningu został skrócony. Trening modelu BERT na pięciu epokach trwał średnio dwie godziny dla wersji jednozadaniowej i około pięć godzin dla wersji wielozadaniowej. Wszystkie modele były trenowane na GPU.

Epoka	Sentyment		Emocje	
	Dokł. (train)	Dokł. (val)	Dokł. (train)	Dokł. (val)
1	70.64%	73.61%	91.25%	95.10%
2	78.91%	73.69%	94.92%	95.14%

Tabela 4.7: Porównanie dokładności podczas treningu modeli jednozadaniowych BERT. Dane własne.

Epoka	Sentyment		Emocje	
	Dokł. (train)	Dokł. (val)	Dokł. (train)	Dokł. (val)
1	69.58%	72.96%	91.06%	94.79%
2	77.71%	72.47%	94.82%	94.97%

Tabela 4.8: Porównanie dokładności podczas treningu modelu wielozadaniowego BERT. Dane własne.

W drugim podejściu, czas treningu modelu BERT na wersji jednozadaniowej wynosił średnio jedną godzinę, podczas gdy dla wersji wielozadaniowej trwał około dwóch godzin. W przypadku klasyfikacji sentymentu, model uzyskał dokładność 74% na zbiorze testowym, 72% na zbiorze walidacyjnym oraz 78% na zbiorze treningowym. Wyniki te wskazują, że model jest właściwie dopasowany, choć na zbiorze walidacyjnym i testowym wystąpił pewien spadek dokładności w porównaniu do zbioru treningowego, co może sugerować lekkie przeuczenie. Natomiast w analizie emocji model osiągnął bardzo wysoką dokładność 95% zarówno na zbiorze walidacyjnym,

jak i testowym, a na zbiorze treningowym wynik ten był niemal identyczny, co sugeruje, że model dobrze radzi sobie z klasyfikacją emocji i nie jest przetrenowany.

W przypadku wersji wielozadaniowej, wyniki były porównywalne.

4.5. Reprodukcja wyników

Kod źródłowy wykorzystany do przeprowadzenia eksperymentów oraz treningu wszystkich modeli jest dostępny pod adresem: <https://github.com/Primuu/MastersThesis>. Repozytorium zawiera dane, skrypty do przygotowania danych, implementacje modeli, procesy treningowe oraz skrypty do ewaluacji i wizualizacji wyników.

Wszystkie eksperymenty były realizowane z użyciem bibliotek:

- `pandas` – wersja 2.2.3,
- `scikit-learn` – wersja 1.5.2,
- `PyTorch` – wersja 2.4,
- `transformers` – wersja 4.46.2,
- `tensorflow` – wersja 2.10,
- `matplotlib` – wersja 3.9.2,
- `seaborn` – wersja 0.13.

4.6. Miary skuteczności

Wszystkie modele zostały ocenione przy użyciu czterech miar: *accuracy*, *precision*, *recall* oraz *F1-score*.

- *Accuracy* (dokładność) oblicza się jako stosunek liczby poprawnych klasyfikacji do całkowitej liczby próbek. W przypadku niezrównoważonych klas może nie oddać pełnej skuteczności modelu.
- *Precision* (precyzja) mierzy, ile z przewidywanych pozytywnych przypadków jest faktycznie pozytywnych. Wysoka precyzja oznacza, że model rzadko popełnia błędy.
- *Recall* (czułość) mierzy, ile z rzeczywistych pozytywnych przypadków zostało poprawnie zidentyfikowanych przez model.
- *F1-score* to średnia harmoniczna między precyzją a czułością, która pozwala na uzyskanie zrównoważonego wyniku. F1-score jest szczególnie użyteczne w przypadkach, gdy klasy są niezrównoważone, ponieważ traktuje precyzję i czułość w równym stopniu.

Rozdział 5

Eksperymenty i wyniki

Celem tego rozdziału jest zaprezentowanie wyników uzyskanych w trakcie przeprowadzonych eksperymentów. Jak zostało wpsomniane, analizowane były trzy zadania: klasyfikacja sentymentu, klasyfikacja emocji oraz jednoczesna klasyfikacja sentymentu i emocji. W pierwszej kolejności omówione zostaną wyniki uzyskane przez modele jednozadaniowe, a następnie zestawione zostaną ich osiągi z modelami wielozadaniowymi.

5.1. Modele jednozadaniowe dla analizy sentymentu

W zadaniu klasyfikacji sentymentu przetestowano cztery różne architektury: BERT, LSTM, GRU oraz klasyfikator SVM.

Model	Dokł.	Prec. (makro)	Czuł. (makro)	F1-score (makro)
BERT	0.74	0.74	0.74	0.74
LSTM	0.67	0.67	0.68	0.67
GRU	0.64	0.64	0.67	0.64
SVM	0.66	0.65	0.66	0.66

Tabela 5.1: Zestawienie miar skuteczności dla modeli jednozadaniowych w zadaniu klasyfikacji sentymentu. Dane własne.

Model BERT uzyskał najwyższe wyniki we wszystkich czterech miarach, osiągając 0.74 zarówno w dokładności, jak i w precyzji, czułości oraz *F1-score*. Oznacza to, że model ten najskuteczniej radził sobie z klasyfikacją wszystkich klas sentymentu w sposób zrównoważony.

Model LSTM uplasował się na drugim miejscu, osiągając dokładność 0.67 oraz zbliżone wartości pozostałych miar. Wskazuje to na stabilną, choć niższą niż BERT, skuteczność klasyfikacji.

Model SVM uzyskał dokładność 0.66, a pozostałe miary także utrzymywały się na podobnym poziomie, co świadczy o jego spójności w przewidywaniu klas, mimo że nie wykorzystuje mechanizmów głębokiego uczenia.

Model GRU osiągnął najniższe wartości, co sugeruje, że był najmniej skuteczny w klasyfikacji spośród porównywanych modeli jednozadaniowych.

5.2. Modele jednozadaniowe dla analizy emocji

W zadaniu identyfikacji emocji wykorzystano te same architektury, które były użyte przy klasyfikacji sentymentu, czyli: BERT, LSTM, GRU oraz SVM.

Model	Dokł.	Prec. (makro)	Czuł. (makro)	F1-score (makro)
BERT	0.95	0.95	0.95	0.95
LSTM	0.95	0.95	0.95	0.95
GRU	0.94	0.94	0.94	0.94
SVM	0.91	0.91	0.91	0.91

Tabela 5.2: Zestawienie miar skuteczności dla modeli jednozadaniowych w zadaniu klasyfikacji emocji. Dane własne.

Model BERT osiągnął najwyższe wyniki spośród wszystkich analizowanych modeli, uzyskując wartość 0.95 we wszystkich czterech miarach: dokładności, precyzji, czułości oraz *F1-score*. Choć model LSTM osiągnął identyczne wartości w tej zbiorczej tabeli, to BERT uzyskał wyższe wyniki dla większości poszczególnych klas emocji, co czyni go obiektywnie lepszym modelem w tym zadaniu. Różnice te zostaną przedstawione w sekcji niżej, poświęconej analizie błędów modeli.

Jak wspomniano, model LSTM uzyskał identyczne wyniki zbiorcze jak BERT, co wskazuje na porównywalną skuteczność tych dwóch architektur w klasyfikacji emocji, jednak wyniki LSTM dla poszczególnych klas emocji były nieco niższe niż w przypadku BERT-a.

Model GRU osiągnął nieco niższe rezultaty – wartości wszystkich miar wyniosły 0.94. Pomimo minimalnej różnicy, model ten prezentuje wysoki poziom skuteczności w zadaniu klasyfikacji emocji.

Model SVM uzyskał najniższe wyniki w zestawieniu – 0.91 dla każdej z miar. Pomimo tego, jego skuteczność pozostaje na zadowalająco wysokim poziomie, biorąc pod uwagę fakt, że jest to model nieneuronowy, oparty na klasycznych metodach przetwarzania tekstu.

5.3. Modele wielozadaniowe

W tej sekcji zestawiono wyniki modeli uczonych w podejściu jednozadaniowym oraz wielozadaniowym (multitasking). Celem porównania było sprawdzenie, czy równoczesne uczenie się klasyfikacji sentymentu i emocji może wpływać na jakość predykcji w porównaniu do klasycznych podejść, w których modele trenowane są oddzielnie dla każdego z zadań.

5.3.1. Zestawienie modeli wielozadaniowych

W tabeli poniżej zaprezentowano wyniki modeli wielozadaniowych opartych na architekturach BERT, LSTM i GRU. Model SVM nie został uwzględniony w analizie modeli wielozadaniowych, ponieważ tradycyjna implementacja SVM, stosowana w niniejszej pracy, nie umożliwia bezpośredniego uczenia wielozadaniowego.

Model	Zadanie	Dokładność	Precyzja	Czułość	F1-score
BERT	Sentyment	0.73	0.72	0.74	0.73
BERT	Emocje	0.95	0.95	0.95	0.95
LSTM	Sentyment	0.67	0.67	0.67	0.67
LSTM	Emocje	0.95	0.95	0.95	0.95
GRU	Sentyment	0.65	0.65	0.67	0.66
GRU	Emocje	0.94	0.94	0.94	0.94

Tabela 5.3: Zestawienie miar skuteczności (makro) dla modeli wielozadaniowych w zadaniu klasyfikacji sentymentu i emocji. Dane własne.

Wyniki pokazują, że modele multitaskingowe również bardzo dobrze radzą sobie w zadaniach klasyfikacyjnych, zwłaszcza w klasyfikacji emocji. Modele BERT i LSTM osiągnęły identyczne wyniki w tabeli zbiorczej dla tego zadania (dla poszczególnych klas wyniki również pozostają bardzo zbliżone), natomiast GRU był minimalnie słabszy. W zadaniu klasyfikacji sentymentu również najlepiej wypadł BERT, a wyniki LSTM i GRU były zbliżone.

5.3.2. Porównanie modeli jednozadaniowych i wielozadaniowych

Jak natomiast radzą sobie modele wielozadaniowe w porównaniu do ich jednozadaniowych odpowiedników? W poniższej tabeli przedstawiono porównanie skuteczności modeli BERT, LSTM oraz GRU, trenowanych osobno dla każdego zadania, z ich odpowiednikami uczonymi w sposób wielozadaniowy.

Model	Zadanie	Dokł.		Prec.		Czuł.		F1-score	
		1-zad.	W-zad.	1-zad.	W-zad.	1-zad.	W-zad.	1-zad.	W-zad.
BERT	Sent.	0.74	0.73	0.74	0.72	0.74	0.74	0.74	0.73
BERT	Emocje	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
LSTM	Sent.	0.67	0.67	0.67	0.67	0.68	0.67	0.67	0.67
LSTM	Emocje	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
GRU	Sent.	0.64	0.65	0.64	0.65	0.67	0.67	0.64	0.66
GRU	Emocje	0.94	0.94	0.94	0.94	0.94	0.94	0.94	0.94

Tabela 5.4: Zestawienie miar skuteczności (makro) dla modeli jednozadaniowych i wielozadaniowych dla każdej architektury. Dane własne.

Z przedstawionych danych wynika, że multitasking nie wpłynął istotnie na pogorszenie bądź polepszenie wyników klasyfikacji. W niektórych przypadkach (jak GRU) nawet lekko poprawił uzyskiwane wyniki dla sentymentu, w innych pogorszył – BERT uzyskuje odrobinę gorsze wyniki dla klasyfikacji sentymentu. Dla klasyfikacji emocji wyniki dla modeli jednozadaniowych i wielozadaniowych były identyczne lub niemal identyczne.

5.4. Analiza błędów modeli

Celem tej sekcji jest zrozumienie, w jakich przypadkach modele klasyfikacyjne zawodzą oraz czy istnieją wzorce w popełnianych błędach.

5.4.1. Klasyfikacja sentymentu

Modele BERT, LSTM, GRU i SVM zostały ocenione pod kątem ich zdolności do klasyfikacji poszczególnych klas sentymentu, a wyniki zostały przedstawione w tabeli poniżej.

Model	Klasa	Precyzja	Czułość	F1-score
BERT	Negatywny	0.72	0.72	0.72
	Neutralny	0.73	0.67	0.70
	Pozytywny	0.75	0.83	0.79
LSTM	Negatywny	0.60	0.70	0.65
	Neutralny	0.66	0.62	0.64
	Pozytywny	0.73	0.71	0.72
GRU	Negatywny	0.55	0.78	0.65
	Neutralny	0.69	0.48	0.57
	Pozytywny	0.69	0.76	0.72
SVM	Negatywny	0.58	0.68	0.62
	Neutralny	0.66	0.63	0.64
	Pozytywny	0.73	0.68	0.71
BERT (multi)	Negatywny	0.65	0.80	0.72
	Neutralny	0.74	0.63	0.68
	Pozytywny	0.78	0.80	0.79
LSTM (multi)	Negatywny	0.60	0.70	0.65
	Neutralny	0.65	0.65	0.65
	Pozytywny	0.76	0.67	0.71
GRU (multi)	Negatywny	0.54	0.77	0.63
	Neutralny	0.68	0.57	0.62
	Pozytywny	0.75	0.69	0.72

Tabela 5.5: Zbiorcze zestawienie miar skuteczności dla modeli sentymentu w podziale na klasy. Dane własne.

Model BERT uzyskał najlepsze wyniki spośród wszystkich modeli, zarówno w wersji jednozadaniowej, jak i wielozadaniowej. Szczególnie dobrze radził sobie w przypadku klasy *Pozytywny* – *F1-score* na poziomie 0.79. Wyniki dla klasy *Neutralny* są jednak zauważalnie słabsze, z *F1-score* odpowiednio równym 0.70 i 0.68. Klasa *Negatywny* również nie wypada najlepiej – *F1* równe 0.72. Może to sugerować, że klasyfikacja neutralnych i negatywnych opinii stanowi trudność.

LSTM, zarówno dla jednozadaniowego, jak i wielozadaniowego modelu, uzyskał wyniki nieco niższe niż BERT. Osiągnął on *F1-score* równy 0.72/0.71 dla klasy *Pozytywny*. Wyniki dla klasy *Neutralny* wynosiły odpowiednio 0.64 i 0.65, również ukazując znaczny spadek w porównaniu z powyższym modelem. Dla klasyfikacji negatywnego sentymentu osiągnięty wynik to 0.65. Wyniki sugerują, że model ten miał trudności nie tylko z rozróżnieniem klasy neutralnej, ale również z klasyfikacją negatywnych opinii.

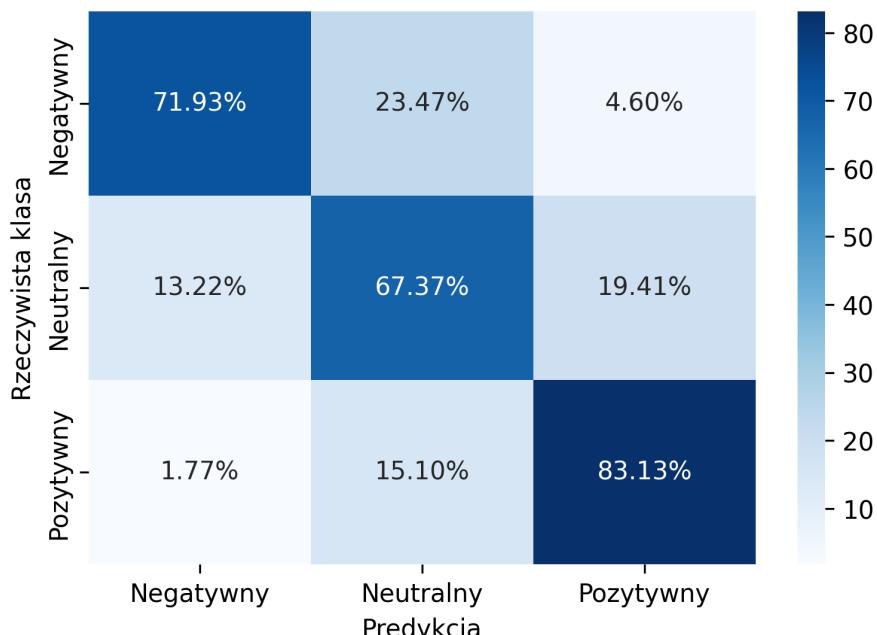
Model GRU radził sobie nieco gorzej w porównaniu do LSTM i BERT-a. Chociaż uzyskał dobrą czułość (0.78/0.77) dla klasy *Negatywny*, jego precyzja (0.55/0.54) była znacznie niższa, co prowadziło do niższego wyniku *F1* (0.65/0.63). Dla klasy *Neutralny* GRU uzyskał najniższe wartości spośród wszystkich modeli w zakresie czułości (0.48/0.57) i *F1* (0.57/0.62). Natomiast w klasyfikacji *Pozytywny* wyniki były solidne, z *F1-score* równym 0.72.

Model SVM z kolei uzyskał wyniki zbliżone do GRU. W klasyfikacji *Pozytywny* osiągnął *F1-score* na poziomie 0.71, a dla klasy *Negatywny* wynik wyniósł 0.62. Choć model ten osiągnął wyższy wynik dla klasy *Neutralny* (0.64), jego ogólna skuteczność w zadaniu była mniejsza w porównaniu do LSTM czy BERT-a.

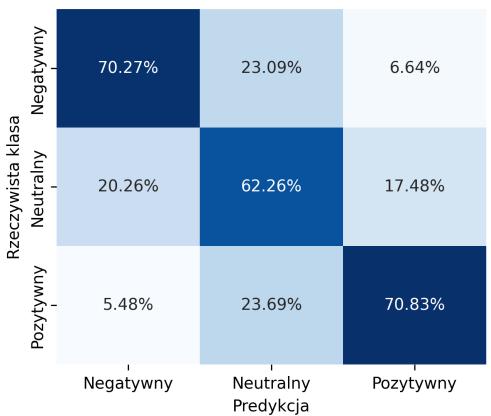
Dla modeli wielozadaniowych wyniki były zbliżone do wersji jednozadaniowych, ale z pewnymi różnicami. Model BERT (multi) uzyskał lepszy wynik w klasyfikacji *Negatywnego* sentymentu, w porównaniu do wersji jednozadaniowej. Model LSTM (multi) uzyskał gorsze wyniki w klasyfikacji *Pozytywnych* opinii, ale za to wykazał wyższą skutecznosć w klasyfikacji opinii *Neutralnych*. Natomiast GRU (multi) lepiej radził sobie z klasą *Neutralny*, jednak w przypadku klasy *Negatywnej* osiągnął gorsze rezultaty niż w wersji jednozadaniowej.

Z analizy wyników modeli jedno- i wielozadaniowych wynika, że multitasking nie poprawił znacznie wyników w porównaniu do podejścia jednozadaniowego, ale także nie pogorszył wyników w klasyfikacji sentymentu. W większości przypadków modele multitaskingowe wykazyły porównywalną skutecznosć, choć z minimalnymi różnicami.

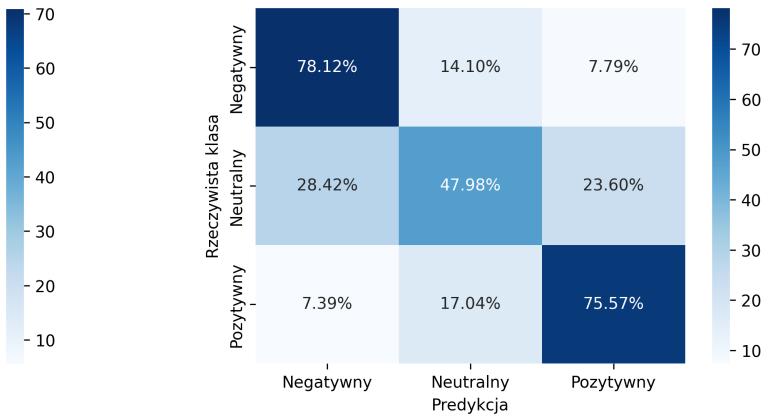
W celu wizualizacji najczęstszych pomyłek, poniżej przedstawiono zbiorcze macierze pomyłek dla każdego z modeli.



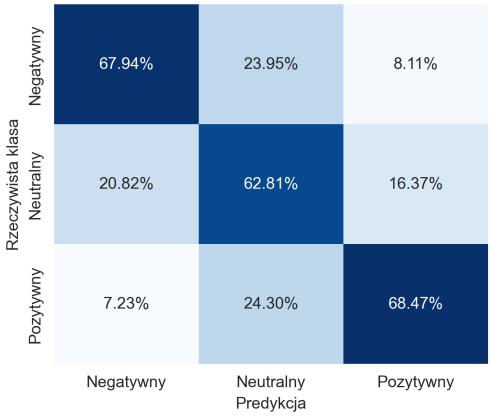
Rysunek 5.1: Macierz pomyłek – BERT. Dane własne.



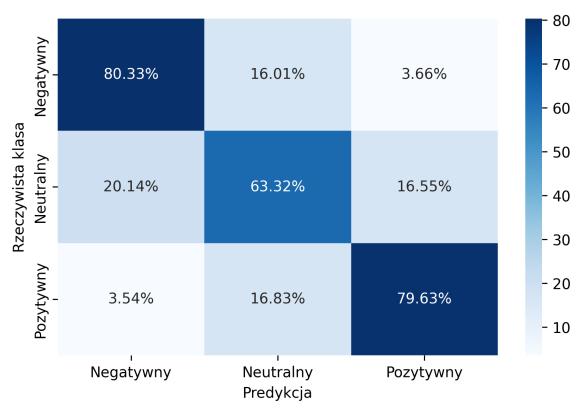
Rysunek 5.2: Macierz pomyłek – LSTM.



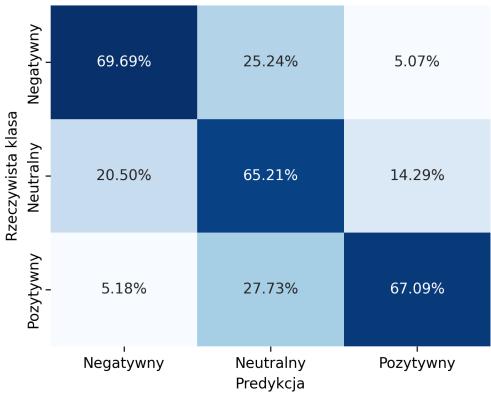
Rysunek 5.3: Macierz pomyłek – GRU.



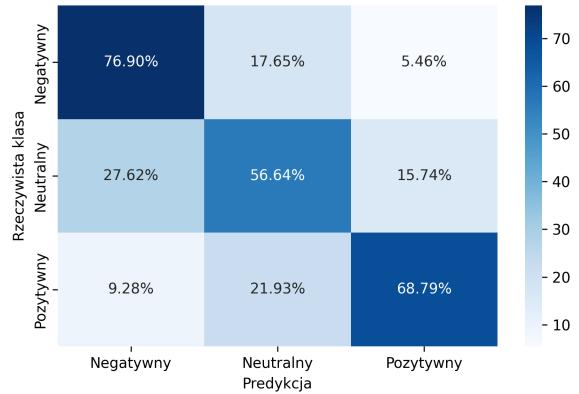
Rysunek 5.4: Macierz pomyłek – SVM.



Rysunek 5.5: Macierz pomyłek – BERT (multi).



Rysunek 5.6: Macierz pomyłek – LSTM (multi).



Rysunek 5.7: Macierz pomyłek – GRU (multi).

Rysunek 5.8: Kolaż macierzy pomyłek dla modeli klasyfikacji sentymentu. Dane własne.

Na podstawie analizy macierzy pomyłek można łatwo zauważyc, że najczęstsze pomyłki występują pomiędzy klasą *Neutralny* a pozostałymi klasami. Modele często błędnie klasyfikowały *Neutralne* wypowiedzi jako *Negatywne*. BERT (zarówno jedno-, jak i wielozadaniowy) znacznie lepiej radził sobie z rozróżnianiem tych klas, co potwierdzają wyższe wartości *F1-score* oraz mniejsze natężenie błędów w macierzy pomyłek.

5.4.2. Klasyfikacja emocji

Po ocenie modeli pod względem ich skuteczności w klasyfikacji emocji, uzyskane wyniki zaprezentowane zostały w poniższej tabeli.

Model	Miara	Emocje					
		Smutek	Radość	Miłość	Złość	Strach	Zaskoczenie
BERT	Precyza	0.99	0.99	0.93	0.96	0.93	0.90
	Czułość	0.95	0.92	1.00	0.95	0.90	1.00
	F1-score	0.97	0.95	0.96	0.95	0.92	0.95
LSTM	Precyza	0.98	1.00	0.93	0.95	0.93	0.90
	Czułość	0.95	0.90	1.00	0.95	0.90	1.00
	F1-score	0.96	0.94	0.96	0.95	0.91	0.94
GRU	Precyza	0.96	0.99	0.93	0.93	0.94	0.90
	Czułość	0.95	0.90	1.00	0.94	0.86	0.99
	F1-score	0.95	0.94	0.96	0.94	0.90	0.94
SVM	Precyza	0.94	0.94	0.89	0.92	0.90	0.86
	Czułość	0.88	0.86	0.97	0.91	0.86	0.97
	F1-score	0.91	0.90	0.93	0.92	0.88	0.92
BERT (m)	Precyza	0.98	0.99	0.93	0.92	0.97	0.90
	Czułość	0.95	0.91	1.00	0.98	0.85	1.00
	F1-score	0.97	0.95	0.96	0.95	0.90	0.95
LSTM (m)	Precyza	0.98	0.98	0.93	0.96	0.92	0.90
	Czułość	0.95	0.91	0.99	0.93	0.90	1.00
	F1-score	0.96	0.95	0.96	0.94	0.91	0.95
GRU (m)	Precyza	0.95	0.99	0.93	0.98	0.92	0.90
	Czułość	0.96	0.91	0.99	0.89	0.90	1.00
	F1-score	0.96	0.95	0.96	0.93	0.91	0.95

Tabela 5.6: Zbiorcze zestawienie miar skuteczności dla modeli emocji w podziale na klasy.
Dane własne.

Choć w klasyfikacji emocji wszystkie analizowane modele osiągnęły bardzo podobną, wysoką skuteczność, to jednak BERT osiągnął najlepsze wyniki, zarówno w wersji jedno- i wielozadaniowej. Szczególnie dobrze radził sobie w klasyfikacji klas *Smutek* oraz *Radość*, które charakteryzowały się największymi rozbieżnościami w wynikach – dla klasy *Smutek* wyniki wahaly się od 0.91 do 0.97, a dla *Radości* od 0.90 do 0.95. Dla pozostałych klas osiągnął najwyższe z uzyskanych przez wszystkie modele wartości. Model BERT w wersji wielozadaniowej wykazał minimalną różnicę – w klasyfikacji *Strachu* uzyskał o 0.02 gorszą skuteczność.

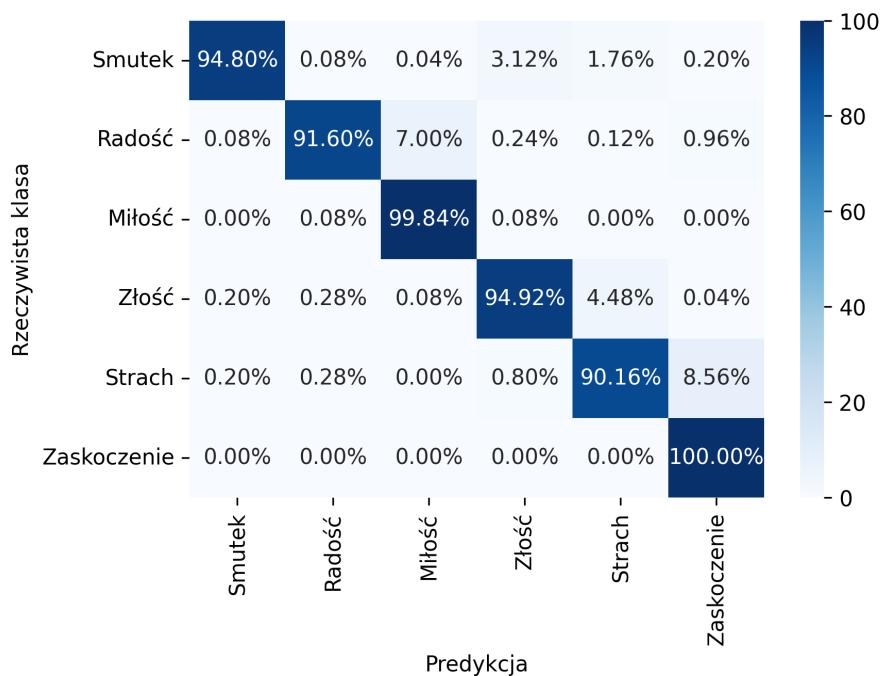
Model LSTM uzyskał dobre rezultaty w klasyfikacji emocji, z wysokimi wynikami dla klasy *Smutek* oraz *Radość*, delikatnie tylko różniąc się na swoją niekorzyść od modelu BERT. W pozostałych klasach uzyskał niemal identyczne wyniki. W wersji wielozadaniowej, model LSTM uzyskał poprawę w klasyfikacji klas *Radość* i *Zaskoczenie*, ale nieco niższe wyniki dla klasy *Złość*, wskazując na nieco lepszą skuteczność w tej wersji.

GRU w wersji jednozadaniowej uzyskał również dobre osiągi, zbliżone do LSTM, z wynikiem $F1$ -score dla klasy *Smutek* równym 0.95. W wersji wielozadaniowej zauważalna jest minimalna poprawa dla tej klasy, w której uzyskał $F1$ -score na poziomie 0.96. Najniższe wyniki w przypadku GRU pojawiły się w klasyfikacji *Strachu*, gdzie $F1$ -score wynosi 0.90/0.91.

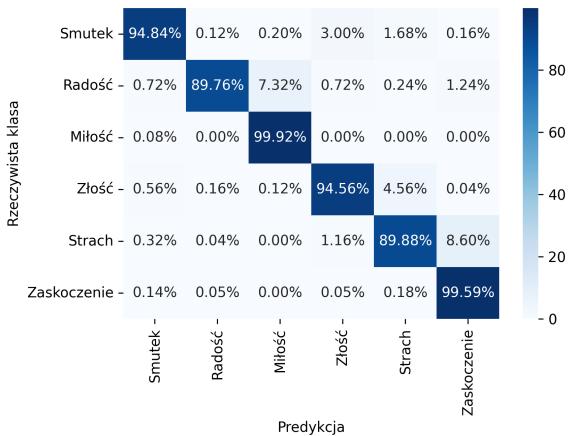
Model SVM uzyskał wyniki wyraźnie niższe niż sieci neuronowe, z niższym $F1$ -score dla wszystkich klas. Najniższy rezultat uzyskał dla klasy *Strach* równy 0.88, podczas gdy modele oparte o sieci neuronowe nie osiągnęły wyniku poniżej 0.90. Dla klas *Radość*, *Miłość* i *Zaskoczenie* model SVM uzyskał również stosunkowo niskie wyniki w porównaniu z innymi modelami.

W przypadku modeli wielozadaniowych, wyniki w każdym przypadku były zbliżone do wersji jednozadaniowych. Uzyskane różnice jednak nie były większe niż 0.01 w każdym przypadku, oprócz BERT-a. Wyniki sugerują, że multitasking ani nie poprawił wyników, ani ich nie pogorszył w porównaniu do podejścia jednozadaniowego.

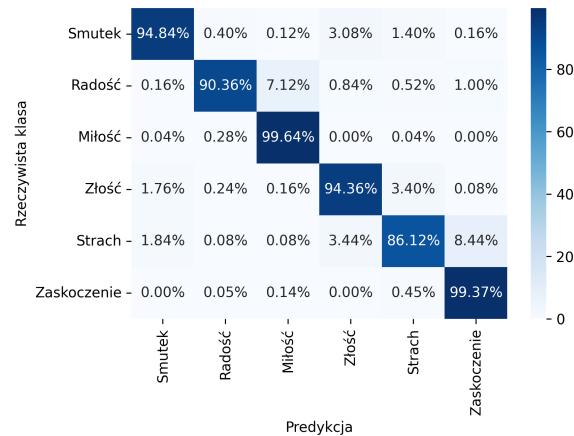
Aby zobrazować najczęstsze błędy, poniżej zaprezentowano zbiorcze macierze pomyłek dla poszczególnych modeli.



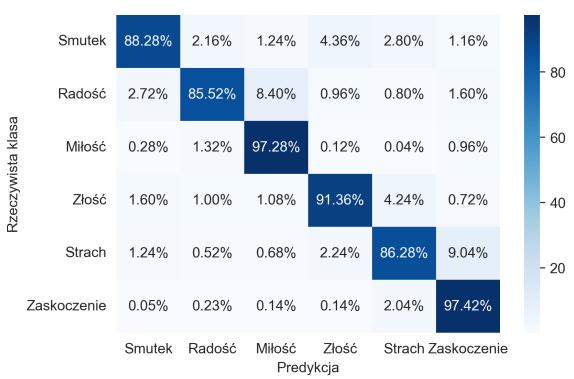
Rysunek 5.9: Macierz pomyłek – BERT. Dane własne.



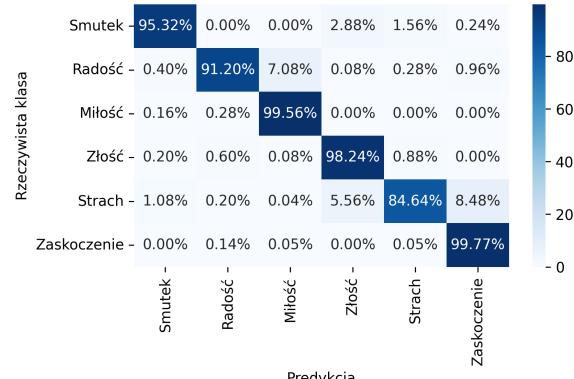
Rysunek 5.10: Macierz pomyłek – LSTM.



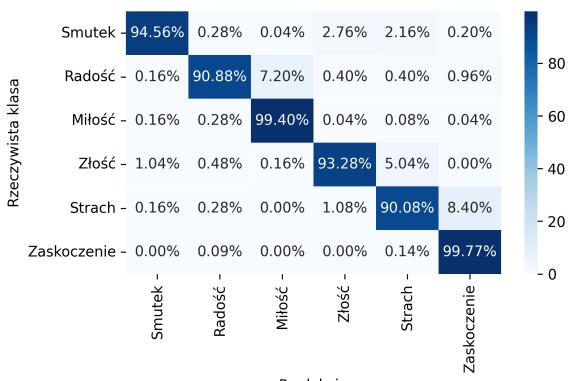
Rysunek 5.11: Macierz pomyłek – GRU.



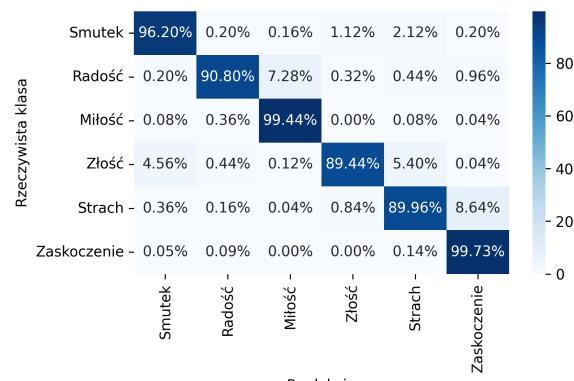
Rysunek 5.12: Macierz pomyłek – SVM.



Rysunek 5.13: Macierz pomyłek – BERT (multi).



Rysunek 5.14: Macierz pomyłek – LSTM (multi).



Rysunek 5.15: Macierz pomyłek – GRU (multi).

Rysunek 5.16: Kolaż macierzy pomyłek dla modeli klasyfikacji emocji. Dane własne.

Analizując macierze pomyłek można zaobserwować, że najczęstsze pomyłki występują między klasą *Strach* a klasą *Zaskoczenie*, a także między klasą *Radość* a klasą *Miłość*. Ponadto, modele często mylą klasę *Złość* z klasą *Strach*. Występują również inne mniejsze błędy pomiędzy poszczególnymi klasami. Ciekawym spostrzeżeniem jest fakt, że klasa *Zaskoczenie* rzadko jest mylona z innymi klasami przez wszystkie modele, z wyjątkiem modelu SVM.

Rozdział 6

Wnioski i podsumowanie

Analizując wyniki uzyskane przez poszczególne modele, można dojść do wniosku, który rzutuje na całą dalszą analizę podsumowującą: analiza sentymentu okazała się zadaniem trudniejszym niż analiza emocji. Z obserwacji błędów modeli oraz charakterystyki statystycznej danych można stwierdzić, że trudność w analizie sentymentu wynika z podobieństwa klasy neutralnej do klas pozytywnej i negatywnej. Brak cech charakterystycznych dla tej klasy sprawia, że modele mają trudności w jej jednoznacznym odróżnieniu od reszty, co prowadzi do większej liczby błędów klasyfikacyjnych. Z kolei analiza emocji była łatwiejsza, ponieważ teksty w tym zadaniu były wyraźnie nacechowane, co dało się zauważyc w charakterystyce statystycznej tego zbioru. Łatwość w poprawnym przypisywaniu właściwej emocji, potwierdza analiza błędów modeli. Część klas była rozpoznawana przez modele niemal bezbłędnie, co świadczy o intensywnym nacechowaniu próbek w zbiorze. Dlatego też analiza sentymentu traktowana będzie jako bardziej wymagający benchmark. Na tej podstawie przedstawione będą dalsze wnioski.

6.1. Najskuteczniejszy model w klasyfikacji sentymentu i emocji

Po przeprowadzeniu eksperymentów można stwierdzić, że najlepsze rezultaty uzyskano dla modelu BERT. W klasyfikacji sentymentu, BERT przewyższał inne modele, osiągając najwyższe wyniki. Model ten, wymagający pod względem zasobów, wykazał się wyższą skutecznością o kilka punktów procentowych w porównaniu do pozostałych podejść. W przypadku analizy emocji, różnice pomiędzy modelami były mniej zauważalne, a BERT osiągnął wyniki niemal identyczne do LSTM i GRU. Sugeruje to, że w prostszych zadaniach BERT jest nadmiernym rozwiązaniem. W takim przypadku LSTM i GRU czy nawet SVM, które są szybsze i mniej zasobożerne, mogą okazać się bardziej efektywne.

6.2. Wpływ uczenia wielozadaniowego

W przypadku uczenia wielozadaniowego wyniki nie wykazały istotnej poprawy ani pogorszenia w porównaniu do modeli jednozadaniowych. Dla BERT-a, wyniki wersji wielozadaniowej były nieznacznie gorsze niż w wersji jednozadaniowej. Natomiast w przypadku LSTM i GRU, uczenie wielozadaniowe miało nieznaczny, korzystny wpływ na wyniki. Z tego punktu widzenia, warto rozważyć użycie dwóch oddzielnych modeli, co zapewnia prostotę, łatwiejszą interpretację wyników i wyższą efektywność.

6.3. Czas treningu i zasoby obliczeniowe

W zakresie czasu treningu, BERT okazał się najwolniejszym modelem. Wysokie zapotrzebowanie na moc obliczeniową BERT-a jest istotnym czynnikiem, który należy wziąć pod uwagę przy jego wyborze. Z kolei LSTM i GRU wymagały znacznie mniej czasu. Są to modele o mniejszej liczbie parametrów, co sprawia, że są bardziej efektywne czasowo i wymagają mniejszych zasobów obliczeniowych, przy zachowaniu równie wysokiej skuteczności w prostych zadaniach. SVM charakteryzował się najszybszym czasem treningu i choć nie potrzebuje dużej mocy obliczeniowej, generalnie wypada gorzej niż bardziej złożone modele, szczególnie w zadaniach bardziej złożonych.

6.4. Wyzwania i ograniczenia modeli NLP w analizie sentymentu

Wyzwania związane z analizą sentymentu wynikają głównie z trudności w rozróżnieniu klasy neutralnej od klas pozytywnej i negatywnej. Analiza emocji okazała się zadaniem łatwiejszym, ponieważ teksty były prawdopodobnie silnie nacechowane emocjonalnie, co ułatwiało klasyfikację. Jednak wciąż niektóre modele miały trudności z klasyfikacją mniej wyraźnych emocji, co wskazuje na pewne ograniczenia. Modele, które potrafią uchwycić bardziej złożone zależności semantyczne, wykazują lepszą skuteczność, nawet w zadaniach, które na pierwszy rzut oka wydają się bardziej jednoznaczne.

W kontekście analizy sentymentu dużą uwagę poświęca się również problemowi biasu (uprzedzeń) w danych, który może negatywnie wpływać na jakość wyników [36]. Bias w danych objawia się jako systematyczne odchylenia lub nierównowagi, które mogą wynikać z różnorodnych źródeł, takich jak np. nieadekwatne reprezentowanie grup demograficznych, tendencyjne źródła danych czy stronnicze etykietowanie [37]. Obecność biasu prowadzi do ograniczonej zdolności generalizacji modeli, co może skutkować preferowaniem lub dyskryminowaniem określonych grup oraz utrwalaniem stereotypów językowych. W związku z tym współczesne badania w dziedzinie NLP podkreślają konieczność identyfikacji i minimalizacji biasu poprzez metody takie jak zrównoważony dobór danych, analiza wpływu cech demograficznych, czy zastosowanie metod regularizacji i uczenia przeciwdziałającego uprzedzeniom [38].

6.5. Podsumowanie

Z przeprowadzonych analiz wynika, że BERT jest najskuteczniejszym modelem w zadaniach trudniejszych, w tym wypadku – analizie sentymentu. W analizie emocji, różnice między modelami były mniejsze, a LSTM i GRU osiągnęły wyniki niemal identyczne do BERT-a. Uczenie wielozadaniowe miało marginalny wpływ na wyniki. Jeśli chodzi o czas treningu, BERT wymaga znacznie większych zasobów obliczeniowych i/lub dłuższego czasu treningu, podczas gdy LSTM i GRU oferują dobry kompromis między dokładnością a szybkością trenowania. SVM, choć szybki i efektywny w zakresie zasobów, nie zapewnia tak wysokiej jakości wyników jak bardziej zaawansowane modele.

Bibliografia

- [1] Bing Liu, *Sentiment Analysis and Opinion Mining*, Morgan & Claypool Publishers, 2012.
- [2] InternetLiveStats.com, *Twitter Usage Statistics*, <https://www.internetlivestats.com/twitter-statistics>, dostęp: maj 2025.
- [3] Yiyi Liu, Yequan Wang, Aixin Sun, Xuying Meng, Jing Li, Jiafeng Guo, *A Dual-Channel Framework for Sarcasm Recognition by Detecting Sentiment Conflict*, arXiv:2109.03587, 2021.
- [4] Nandwani, P., Verma, R, *A review on sentiment analysis and emotion detection from text*. *Soc. Netw. Anal. Min.* 11, 81, <https://doi.org/10.1007/s13278-021-00776-6>, 2021.
- [5] Dan Jurafsky and James H. Martin, *Speech and Language Processing (3rd ed. draft)*, <https://web.stanford.edu/~jurafsky/slp3>, 2025.
- [6] Johri, Prashant and Khatri, Sunil Kumar and Al-Taani, Ahmad and Sabharwal, Munish and Suvanov, Shakhzod and Chauhan, Avneesh, *Natural Language Processing: History, Evolution, Application, and Future Work*, https://doi.org/10.1007/978-981-15-9712-1_31, 2021.
- [7] Diksha Khurana, Aditya Koli, Kiran Khatter, Sukhdev Singh, *Natural Language Processing: State of The Art, Current Trends and Challenges*, arXiv:1708.05148, 2017.
- [8] Sharat Sachin, Abha Tripathi, Navya Mahajan, Shivani Aggarwal, Preeti Nagrath, *Sentiment Analysis Using Gated Recurrent Neural Networks*, <https://doi.org/10.1007/s42979-020-0076-y>, 2020.
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, arXiv:1810.04805, 2018.
- [10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, *RoBERTa: A Robustly Optimized BERT Pretraining Approach*, arXiv:1907.11692, 2019.
- [11] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut, *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*, arXiv:1909.11942, 2020.
- [12] Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf, *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*, arXiv:1910.01108, 2020.
- [13] International Business Machines Corporation, *Oracle Naive Bayes*, <https://www.ibm.com/docs/p1/spss-modeler/saas?topic=mining-oracle-naive-bayes>, dostęp: marzec 2025.
- [14] D.C. Asogwa, S.O. Anigbogu, I.E. Onyenwe, F.A. Sani, *Text Classification Using Hybrid Machine Learning Algorithms on Big Data*, arXiv:2103.16624, 2021.
- [15] D.C Asogwa, C.I Chukwuneke, C.C Ngene, G.N Anigbogu, *Hate Speech Classification Using SVM and Naive BAYES*, arXiv:2204.07057, 2022.
- [16] Yasmen Wahba, Nazim Madhvaji, John Steinbacher, *A Comparison of SVM against Pre-trained Language Models (PLMs) for Text Classification Tasks*, arXiv:2211.02563, 2022.
- [17] Keshav Kapur, Rajitha Harikrishnan, *Comparative Study of Sentiment Analysis for Multi-Sourced Social Media Platforms*, arXiv:2212.04688, 2022.
- [18] Mohamed Kayed, Rebeca P. Díaz-Redondo, Alhassan Mabrouk, *Deep Learning-based Sentiment Classification: A Comparative Survey*, arXiv:2312.17253, 2023.
- [19] Mahdi Rezapour, *Emotion Detection with Transformers: A Comparative Study*, arXiv:2403.15454, 2024.

- [20] Pankaj M Thakur and Tejas M, *Multi-class Classification of Twitter Sentiments using Frequency based, LSTM and BERT Methods*, <https://api.semanticscholar.org/CorpusID:248629612>, 2022.
- [21] Kaggle Datasets, *Twitter Tweets Sentiment Dataset*, <https://www.kaggle.com/datasets/yasserh/twitter-tweets-sentiment-dataset>, licencja: CC0 Public Domain, dostęp: listopad 2024.
- [22] Shahriar Parvez, *Multiclass Sentiment Analysis Dataset*, <https://huggingface.co/datasets/Sp1786/multiclass-sentiment-analysis-dataset>, licencja: Apache 2.0, dostęp: listopad 2024.
- [23] Sara Rosenthal, Noura Farra, Preslav Nakov, *SemEval-2017 Task 4: Sentiment Analysis in Twitter*, Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 502–518, 2017. https://huggingface.co/datasets/cardiffnlp/tweet_eval, licencja: CC BY 3.0, dostęp: listopad 2024.
- [24] Marcos Penelas, *Emotion Dataset*, <https://www.kaggle.com/datasets/marcospenelas/emotion>, licencja: MIT, dostęp: listopad 2024.
- [25] Aliyah Kurniasih and Lindung Parningotan Manik, *On the Role of Text Preprocessing in BERT Embedding-based DNNs for Classifying Informal Texts*, International Journal of Advanced Computer Science and Applications(IJACSA), 13(6) <http://dx.doi.org/10.14569/IJACSA.2022.01306109>, 2022.
- [26] Tan, Kian & Lee, Chin-Poo & Lim, Kian, *RoBERTa-GRU: A Hybrid Deep Learning Model for Enhanced Sentiment Analysis*, Applied Sciences. 13. 3915. 10.3390/app13063915, 2023.
- [27] Sophie Henning, William Beluch, Alexander Fraser, Annemarie Friedrich, *A Survey of Methods for Addressing Class Imbalance in Deep-Learning Based Natural Language Processing*, arXiv:2210.04675, 2022.
- [28] Aurélien Géron, *Uczenie maszynowe z użyciem Scikit-Learn, Keras i TensorFlow*, Helion, 2023.
- [29] Scikit-learn developers, *SVC - Support Vector Classification*, <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>, dostęp: luty 2025.
- [30] Kalcheva, Neli and Karova, Milena and Penev, Ivaylo, *Comparison of the accuracy of SVM kernel functions in text classification*, <https://doi.org/10.1109/BIA50171.2020.9244278>, 2020.
- [31] Hochreiter, Sepp and Schmidhuber, Jürgen, *Long Short-Term Memory*, <https://doi.org/10.1162/neco.1997.9.8.1735>, 1997.
- [32] PyTorch Contributors, *PyTorch documentation*, <https://pytorch.org/docs/stable/index.html>, dostęp: luty 2025.
- [33] Keras Contributors, *Keras Documentation*, <https://keras.io/api/>, dostęp: luty 2025.
- [34] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, Yoshua Bengio, *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*, arXiv:1412.3555, 2014.
- [35] Hugging Face, *Transformers Documentation*, <https://huggingface.co/docs/transformers/en/index>, dostęp: styczeń 2025.
- [36] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, Adam Kalai, *Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings*, arXiv:1607.06520, 2016.
- [37] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, Aram Galstyan, *A Survey on Bias and Fairness in Machine Learning*, arXiv:1908.09635, 2022.
- [38] Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, William Yang Wang, *Mitigating Gender Bias in Natural Language Processing: Literature Review*, arXiv:1906.08976, 2019.

Spis rysunków

3.1	Rozkład procentowy klas w zbiorze do klasyfikacji emocji. Dane własne.	12
3.2	Rozkład procentowy klas w zbiorze do klasyfikacji sentymentu. Dane własne.	13
3.3	Rozkład liczebności poszczególnych klas w zbiorze danych do analizy emocji. Dane własne.	14
3.4	Rozkład długości tekstów w zbiorze danych do analizy emocji. Dane własne.	14
3.5	Rozkład długości tekstów w zbiorze danych do analizy emocji o długości poniżej 400 znaków. Dane własne.	15
3.6	Rozkład liczby znaków w próbkach w zbiorze do analizy emocji. Dane własne.	15
3.7	Rozkład długości tekstów w zbiorze danych do analizy emocji po usunięciu wartości odstających. Dane własne.	16
3.8	Chmura słów dla zbioru do analizy emocji. Dane własne.	16
3.9	Chmura słów dla zbioru do analizy emocji po usunięciu słów funkcjonalnych. Dane własne.	17
3.10	Rozkład liczebności poszczególnych klas w zbiorze danych do analizy sentymentu. Dane własne.	18
3.11	Rozkład długości tekstów w zbiorze danych do analizy sentymentu. Dane własne.	18
3.12	Rozkład długości tekstów w zbiorze danych do analizy sentymentu o długości poniżej 700 znaków. Dane własne.	19
3.13	Rozkład liczby znaków w próbkach w zbiorze do analizy sentymentu. Dane własne.	19
3.14	Rozkład długości tekstów w zbiorze danych do analizy sentymentu po usunięciu wartości odstających. Dane własne.	20
3.15	Chmura słów dla zbioru do analizy sentymentu. Dane własne.	20
3.16	Chmura słów dla zbioru do analizy sentymentu po usunięciu słów funkcjonalnych. Dane własne.	21
5.1	Macierz pomyłek – BERT. Dane własne.	34
5.2	Macierz pomyłek – LSTM.	35
5.3	Macierz pomyłek – GRU.	35
5.4	Macierz pomyłek – SVM.	35
5.5	Macierz pomyłek – BERT (multi).	35
5.6	Macierz pomyłek – LSTM (multi).	35
5.7	Macierz pomyłek – GRU (multi).	35
5.8	Kolaż macierzy pomyłek dla modeli klasyfikacji sentymentu. Dane własne.	35
5.9	Macierz pomyłek – BERT. Dane własne.	37
5.10	Macierz pomyłek – LSTM.	38
5.11	Macierz pomyłek – GRU.	38
5.12	Macierz pomyłek – SVM.	38
5.13	Macierz pomyłek – BERT (multi).	38
5.14	Macierz pomyłek – LSTM (multi).	38
5.15	Macierz pomyłek – GRU (multi).	38
5.16	Kolaż macierzy pomyłek dla modeli klasyfikacji emocji. Dane własne.	38

Spis tabel

4.1	Porównanie dokładności podczas treningu modeli jednozadaniowych LSTM. Dane własne.	24
4.2	Porównanie dokładności podczas treningu modelu wielozadaniowego LSTM. Dane własne.	25
4.3	Porównanie dokładności podczas treningu modeli jednozadaniowych GRU. Dane własne.	26
4.4	Porównanie dokładności podczas treningu modelu wielozadaniowego GRU. Dane własne.	26
4.5	Porównanie dokładności podczas treningu na pięciu epokach modeli jednozadaniowych BERT. Dane własne.	27
4.6	Porównanie dokładności podczas treningu na pięciu epokach modelu wielozadaniowego BERT. Dane własne.	28
4.7	Porównanie dokładności podczas treningu modeli jednozadaniowych BERT. Dane własne.	28
4.8	Porównanie dokładności podczas treningu modelu wielozadaniowego BERT. Dane własne.	28
5.1	Zestawienie miar skuteczności dla modeli jednozadaniowych w zadaniu klasyfikacji sentymentu. Dane własne.	30
5.2	Zestawienie miar skuteczności dla modeli jednozadaniowych w zadaniu klasyfikacji emocji. Dane własne.	31
5.3	Zestawienie miar skuteczności (makro) dla modeli wielozadaniowych w zadaniu klasyfikacji sentymentu i emocji. Dane własne.	32
5.4	Zestawienie miar skuteczności (makro) dla modeli jednozadaniowych i wielozadaniowych dla każdej architektury. Dane własne.	32
5.5	Zbiorcze zestawienie miar skuteczności dla modeli sentymentu w podziale na klasy. Dane własne.	33
5.6	Zbiorcze zestawienie miar skuteczności dla modeli emocji w podziale na klasy. Dane własne.	36

Streszczenie

Praca przedstawia analizę porównawczą modeli uczenia maszynowego wykorzystywanych do klasyfikacji sentymentu oraz emocji w tekstach. Zastosowane modele obejmowały BERT, LSTM, GRU oraz SVM, które zostały ocenione pod kątem ich skuteczności w zadaniach klasyfikacji. Wyniki uzyskane podczas eksperymentów wykazały, że efektywność modeli zależy od stopnia skomplikowania zadania. Analiza samych zbiorów danych oraz późniejsza analiza błędów modeli ujawniły, że klasyfikacja sentymentu okazała się bardziej złożona niż klasyfikacja emocji. W kontekście skuteczności, model BERT okazał się najlepszy w zadaniu bardziej złożonym. Niemniej jednak, w przypadku analizy emocji, modele oparte na sieciach neuronowych, takie jak LSTM i GRU, nie ustępowały BERT-owi, wykazując porównywalną efektywność. W zakresie uczenia wielozadaniowego eksperymenty nie ujawniły wyraźnego wpływu tego podejścia na jakość predykcji. Jeśli chodzi o czas treningu, BERT jest modelem najwolniejszym. Jego zapotrzebowanie na moc obliczeniową może stanowić barierę, z kolei modele LSTM i GRU charakteryzowały się znacznie krótszym czasem treningu.

Abstract

Multiclass sentiment analysis using NLP models – method comparison

This paper presents a comparative analysis of machine learning models used for sentiment and emotion classification in text. The models applied include BERT, LSTM, GRU, and SVM, which were evaluated in terms of their effectiveness in classification tasks. The results obtained from the experiments showed that the effectiveness of the models depends on the complexity of the task. Analysis of the datasets and subsequent model error analysis revealed that sentiment classification was more complex than emotion classification. In terms of effectiveness, the BERT model proved to be the best for the more complex task. However, for emotion analysis, neural network-based models such as LSTM and GRU performed comparably to BERT. Regarding multitask learning, experiments did not reveal a significant impact of this approach on prediction quality. As for training time, BERT was the slowest model. Its computational requirements may be a barrier, while LSTM and GRU models exhibited significantly shorter training times.